This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102238, IEEE Internet of Things Journal

ADGCN: An Asynchronous Dilation Graph Convolutional Network for Traffic Flow Prediction

Tao Qi, Guanghui Li†, Lingqiang Chen and Yanming Xue

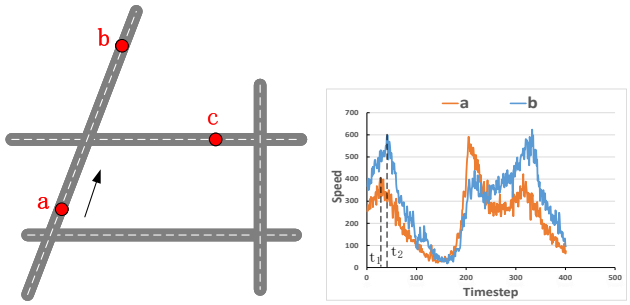
***Abstract*—Spatial-temporal graph modeling plays an important role in the fields of transportation, meteorology, and social networks. Traffic flow prediction is a classic spatial-temporal modeling task. Existing methods usually do not take into account the asynchronous spatial-temporal correlation in traffic data. In addition, due to the complexity and variability of traffic data, long-term traffic forecasting is highly challenging. In order to solve the above problems, this paper proposes a new deep learning-based Asynchronous Dilation Graph Convolution Network (ADGCN) to model the spatial-temporal graphs. We mine the asynchronous spatial-temporal correlation in the traffic network, and propose the Asynchronous Spatial-Temporal Graph Convolution (ASTGC) operation to extract this special relationship. Furthermore, we extend the dilated 1D causal convolution to a graph convolution. The receptive field of the model increases exponentially with the increase of the network depth. Experiments are conducted on three public traffic datasets, and the results show that the prediction performance of ADGCN is better than the existing counterpart methods, especially in long-term prediction tasks.**

***Index Terms*—Traffic flow prediction; graph convolutional network; asynchronous spatial-temporal correlation; dilated causal convolution.**

I. Introduction

THE modern city is gradually developing into a smart city [1]. With the acceleration of urbanization and the urban population's substantial growth, comprehensive urban governance faces tremendous pressure. Traffic management is a vital part of urban management, aiming to relieve the traffic congestion, exhaust pollution, and traffic accidents. In recent years, intelligent transportation systems (ITS) play an increasingly important role in urban traffic management and smart city construction. Traffic prediction is the basis of intelligent transportation systems, and accurate traffic prediction is essential for many applications. For example, traffic speed prediction can avoid accidents to the greatest extent; travel demand prediction can facilitate the online car-hailing operation platform to allocate sufficient online car-hailing resources to areas with high travel demand [2]; reliable congestion time prediction is of great benefit to ease congestion and reduce travel time [3].

Given the historical traffic flow information and the traffic network information, traffic prediction is to predict the traffic network's future traffic flow information. With the continuous growth of traffic-related available data sets and the development of machine learning technology, more and more researchers have begun to try to use deep learning methods to study traffic prediction problems. Most recent studies use Graph Neural Networks (GNN) to model the spatial connections of traffic data and use RNN or CNN to model the temporal correlation of traffic data. These models effectively model the spatial and temporal connections in the transportation network and have achieved good results in various spatial-temporal prediction tasks. However, none of these models considered the asynchronous spatial-temporal correlation in traffic data.



(a) (b)

Fig. 1: Asynchronous spatial-temporal correlation in traffic data. (a) The traffic network. (b) The traffic speed of node a and node b.

Consider the traffic network shown in Fig. 1(a). Node a, node b, and node c are sensor nodes placed on the traffic network, where node a is the upstream node of node b. Fig. 1(b) shows the changing trend of the traffic speed of node a and node b in a certain period. It can be seen from Fig. 1 that at time t_1, the traffic speed at node a reaches a local maximum, and the traffic speed at node b is not immediately affected but reaches the local maximum at time t_2. This means that the change in traffic flow between two nodes is asynchronous. In other words, the traffic speed has an obvious spatial correlation, this correlation is not isolated but highly correlated with time. We often observe this phenomenon in our daily life: when traffic congestion occurs on a road section,

† Corresponding author.

T. Qi is with the Department of Computer Science, Jiangnan University, Wuxi, Jiangsu, 214122, China (e-mail: 6191914042@stu.jiangnan.edu.cn).

G. Li is with the Department of Computer Science, Jiangnan University, Wuxi, Jiangsu, 214122, China, and is also with the Research Center for IoT Technology Application Engineering (MOE), Wuxi, Jiangsu, 214122 China (e-mail: ghli@jiangnan.edu.cn).

L. Chen is with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China (e-mail: lqchen@stu.jiangnan.edu.cn).

Y. Xue is with the Department of Computer Science, Jiangnan University, Wuxi, Jiangsu, 214122, China (e-mail: 6191910035@stu.jiangnan.edu.cn).

the adjacent road section may become congested after a while. We call this phenomenon the asynchronous spatial-temporal correlation of traffic flow. Modeling the asynchronous spatial-temporal correlation in the traffic network can effectively improve the prediction performance of the model.

Besides, the current research on spatial-temporal graph modeling is not effective in learning temporal dependence. RNN-based approaches often have problems such as time-consuming iterative propagation, gradient explosion/vanishing and do not perform well in long-term prediction tasks [4], [5]. The advantage of the CNN-based methods is that it can be calculated in parallel, the gradient is relatively stable during the calculation process, and the memory requirement is low. However, these works use standard 1D convolution. The receptive field size increases linearly with the increase of the number of hidden layers in the network, so these models need to use many layers to capture long-range sequences.

To solve the above-mentioned problems, we present a method based on a graph neural network: Asynchronous Spatial-temporal Dilation Graph Convolutional Network (ADGCN) to model the traffic network. We propose an Asynchronous Spatial-Temporal Correlation Matrix (ASTCM) and its corresponding Adaptive Asynchronous Spatial-Temporal Correlation Weight Matrix (adpASTCWM) to model the asynchronous spatial-temporal correlation in traffic data. Motivated by Graph WaveNet [6], we extend the 1D dilated causal convolution network to the graph convolution network, named Asynchronous Spatial-Temporal Dilated Causal Convolution (ASTDC). We stack multiple ASTDC layers to extract the long-term asynchronous spatial-temporal dependence of traffic data. The receptive field of the model increases exponentially as the number of ASTDC layers increases. The main contributions of this work are as follows:

1) We design two matrices (ASTCM and adpASTCWM) to model the asynchronous spatial-temporal correlation in traffic data. These two components flexibly model the complex and dynamic spatial-temporal dependencies in the transportation network.
2) We extend the 1D dilated causal convolution to the graph convolution network, and propose Asynchronous Spatial-Temporal Dilated Causal Convolution to capture the long-term temporal dependencies with a shallower network, effectively alleviating overfitting.
3) We conduct detailed experiments of the proposed model on three real-world traffic datasets. Compared with the existing model, the prediction performance of the ADGCN model has been significantly improved.

The remainder of this paper is organized as follows. Section II covers the literature on traffic prediction. Section III introduces the definition of the problem studied in this research. Section IV presents the details of our method. In section V, we evaluate the predictive performance of the ADGCN. We conclude the paper in Section VI.

II. RELATED WORK

A. Traffic Flow Prediction

In recent years, the research of traffic prediction has achieved great success. Recently proposed traffic prediction methods can be roughly divided into traditional parametric methods and machine learning methods [7]. The former mainly includes the time series model, the linear regression model [8], and the Kalman filter model. The representative method of the time series model is the Autoregressive Integrated Moving Average Model (ARIMA). Hamed *et al*. [9] proposed a prediction method, using ARIMA to predict the short-term traffic volume of urban arterial roads. In order to achieve better prediction performance, researchers have successively proposed Kohonen ARIMA [10], subset ARIMA [11], seasonal ARIMA [12], and other methods. The linear regression model uses a regression function based on historical traffic data to predict future traffic flow. Sun *et al*. [13] proposed a method based on a local linear predictor for interval prediction of traffic time series and achieved good results on real traffic data. The Kalman filter model uses the traffic state at the previous moment and the current moment to predict the future traffic state. Hinsbergen *et al*. [14] used the Kalman filter to complete the traffic prediction task. Traditional parametric methods are relatively simple and convenient to calculation. Nevertheless, these methods require data to satisfy certain assumptions, and time-varying traffic data is too complex to satisfy these assumptions [7].

Machine learning methods only need enough historical data to automatically learn statistical regularity from traffic data, which solves these problems well. Zhang *et al*. [15] applied the k-nearest neighbor model to predict short-term traffic flow. Yao *et al*. [16] proposed a method based on the support vector regression model to fit traffic flow data. Sun *et al*. [17] used Bayesian network models to model traffic data.

In recent years, with the rapid development of deep learning and the substantial increase in available traffic data, deep neural network models have received more and more attention. According to whether the spatial correlation is considered, deep neural network models can be divided into two categories. Some early methods only focus on temporal correlation, e.g., Rilett *et al*. [18] used a multilayer feedforward neural network (FNN) to predict freeway link travel times. Lint *et al*. [19] used the recurrent neural network to model traffic flow. Fu *et al*. [20] applied long short-term memory network (LSTM) and gated recurrent unit (GRU) to extract time dependence in traffic data. Fang *et al*. [21] proposed the Kalman-LSTM model, which combines Kalman filtering and LSTM for short-term traffic flow forecasting.

The traffic network has a natural topology. Hence the spatial correlation of traffic data is also a vital feature. Wu and Tan [22] proposed a novel short-term traffic prediction method that extracts the spatial-temporal correlation of traffic data by fusing CNN and LSTM. Cao *et al*. [23] designed an Interactive Time Recurrent Convolutional Network (ITRCN), which uses a CNN to extract spatial features and a GRU to extract temporal features. Zhang *et al*. [24] proposed the DeepST model, which employs the framework of CNN to simultaneously model spatial near and distant dependencies and temporal closeness, period, and trend. Wang *et al*. [25] present a geo-convolution operation by integrating the geographic information into the classical convolution, capable of capturing spatial correlations. Zonoozi *et al*. [26] constructed a novel Periodic-CRN (PCRN) model, which adapts convolutional

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102238, IEEE Internet of Things Journal

recurrent networks (CRN) to capture spatial and temporal correlations, combined with explicit periodic representations. Lu *et al*. [27] present a temporal-aware LSTM enhanced by loss-switch mechanism for short-term traffic flow forecasting.

The performance of CNN-based model on the data in non-Euclidean space is not satisfactory. In recent years, the emergence and rapid development of Graph Convolutional Networks (GCN) [28] [29] have provided a solution to this problem. Yu *et al*. [30] proposed a Spatio-temporal Graph Convolution Network (STGCN) model, which uses Chebyshev approximation graph convolution operation to extract spatial dependencies, and uses Gated Convolutional Neural Network (gate-CNN) modeling temporal dependent, achieving faster training speed with fewer parameters. Geng *et al*. [31] encoded different kinds of non-Euclidean pairwise correlations between regions into multiple spatial graphs, then they used multi-graph convolution to extract these correlations. Chen *et al*. [32] proposed a Multiple Residual Recurrent Graph Neural Network (MRes-RGNN), which simultaneously captures the spatial dependencies and temporal dynamics based on the graph. Zhao *et al*. [33] proposed a Time Graph Convolutional Network (T-GCN) model, representing the traffic network as a graph structure, and combined GCN and GRU to model the spatial-temporal dependence of traffic data. Li *et al*. [34] first modeled traffic flow as a diffusion process on a directed graph and proposed a Diffusion Convolutional Recurrent Neural Network (DCRNN) model. Wu *et al*. [6] improved the diffusion process in DCRNN by applying an adaptive adjacency matrix to consider the dynamics of spatial associations between nodes and their neighbors. Xie *et al*. [35] believed that the process of information diffusion in spatial and temporal dimensions is homogeneous and proposed an ISTD-GCN model to simultaneously extract spatial and temporal features.

The above-mentioned methods have achieved great success in traffic prediction tasks. However, there is few research on modeling the asynchronous spatial-temporal correlation in traffic data. In this study, we propose a new neural network method, which models the asynchronous spatial-temporal correlation in traffic data and combines the dilated causal convolutional network to achieve high prediction accuracy with fewer parameters.

B. Graph Convolution Network

Convolutional neural networks have achieved remarkable performance on various tasks in Euclidean space. However, many cases in the real world are non-Euclidean, such as social networks, protein molecular structures, traffic networks. A graph is a classic non-Euclidean data structure, which can well represent the rich attribute information and their connections. Due to the powerful expressive ability of graphs, research of analyzing graphs with machine learning has been attracted more and more attention [36]. The concept of graph neural network was first proposed in [37]. After decades of development, researchers have proposed various variants of graph neural networks. Existing graph neural networks are mainly divided into spectral-based methods and spatial-based methods. Spectral-based approaches define graph convolution from the perspective of graph signal processing [38], where the result of the graph convolution operation is the graph signal with noise removed [39], [40], [28]. Spatial-based approaches define graph convolution from the perspective of information dissemination. Monti *et al*. [41] use a Gaussian kernel to learn the connection weights between nodes and neighbors. Velickovic *et al*. [42] proposed a graph attention network, which leverages a masked self-attention mechanism to update node weights.

III. PRELIMINARIES

In this section, we introduce the problem definition studied in this article. Table I lists the description of key symbols. For brevity, we describe some abbreviations in Table II.

The traffic network is regarded as a spatial graph, which can be expressed as $G = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathcal{V}$ is a set of road nodes, $\mathcal{E}$ is a set of edges, $\mathbf{A} \in \mathbb{R}^{N \times N}$ is adjacency matrix of the graph, which represent the connection between nodes. If $v_i, v_j \in \mathcal{V}$ and $(v_i, v_j) \in \mathcal{E}$, then the matrix element $\mathbf{a}_{ij}$ is 1 otherwise it is 0. At each time step t, the spatial graph G has a feature matrix $\mathbf{X}^{(t)} \in \mathbb{R}^{N \times D}$. Given the feature matrix of the graph G and its S historical time steps, the task is to predict the feature matrix of the graph G in the next T time steps. The mapping relationship can be expressed as:

$$\left[\mathbf{X}^{(t-S):t}, G\right] \xrightarrow{f} \mathbf{X}^{(t+1):(t+T)} \tag{1}$$

Where $\mathbf{X}^{(t-S):t} \in \mathbb{R}^{N \times D \times S}$ is the input of the model, $\mathbf{X}^{(t+1):(t+T)} \in \mathbb{R}^{N \times D \times T}$ is the prediction output of the model, and f is the model we learned.

IV. METHODOLOGY

In this section, we first describe the various modules in the ADGCN model in detail: Asynchronous Spatial-Temporal Correlation Matrix, Adaptive Asynchronous Spatial-Temporal Correlation Weight Matrix, Asynchronous Spatial-Temporal Graph Convolution operation, and Asynchronous Spatial-Temporal Dilated Convolution layer. Then we outline the framework of ADGCN.

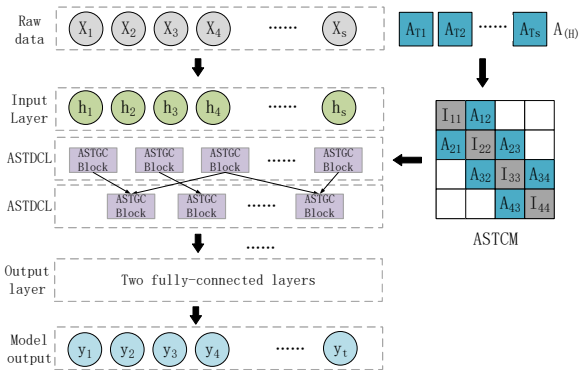


Fig. 2: Framework of ADGCN.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102238, IEEE Internet of Things Journal

TABLE I: Summary of key symbols

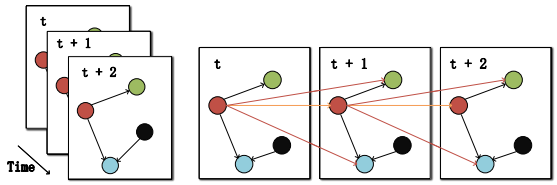
Symbol	Description	Symbol	Description
G	The spatial graph of traffic network	N	The number of sensor nodes
$\mathbb{G}$	Collection of spatial graph	D, C, C'	The number of feature dimensions
$G_{(H)}$	Asynchronous spatial-temporal correlation graph	m	The number of spatial graphs in asynchronous spatial-temporal graph
$\mathcal{V}, \mathcal{V}_{(H)}$	The set of vertices of graph	d, d^s	Dilation factor
$\mathcal{E}, \mathcal{E}_{(H)}$	The set of edges of graph	I	Identity matrix
$\mathbf{A}, A_{(H)}$	Adjacency matrix of graph	$A_{(adp)}$	Adaptive Asynchronous Spatial-Temporal Correlation Weight Matrix
$\mathbf{X}^{(t)}$	Feature matrix of spatial graph at time t	h	Intermediate calculation results of the model
S, S^s, T, T_o	The number of time steps	$\mathbf{W}, \mathbf{b}$	Trainable parameters

TABLE II: Description of the abbreviated symbols

Abbreviations	Description
ADGCN	Asynchronous Spatial-Temporal Dilation Graph Convolutional Networks
ASTCM	Asynchronous Spatial-Temporal Correlation Matrix
ASTGC	Asynchronous Spatial-Temporal Graph Convolution
adpASTCWM	Adaptive Asynchronous Spatial-Temporal Correlation Weight Matrix
ASTDC	Asynchronous Spatial-Temporal Dilated Causal Convolution

A. Asynchronous Spatial-Temporal Correlation Matrix

Classical graph convolutional network-based traffic flow prediction methods often include two components: one is a spatial information extraction component, which is mainly composed of GCN; the other is the temporal information extraction component, which is mainly based on RNN. By alternately using these two components, these models can pass the information of a single node to other nodes related to it in spatial and temporal dimensions. However, there are two issues with this extraction method. First, the spatial-temporal correlation between traffic data is often asynchronous. The classic models does not take this issue into account. The second is that some information may be lost in the process of alternately extracting spatial-temporal correlations.



(a) (b)

Fig. 3: (a) The historical observation data at each time step constitutes a spatial graph separately. (b) Asynchronous spatial-temporal correlation graph.

To this end, we represent the traffic network as an asynchronous spatial-temporal graph to model the asynchronous spatial-temporal correlation between traffic data directly. As shown in Fig. 3(a), the traffic network has a natural topological structure. Hence the traffic data collected at each time step can form a spatial graph. In Fig. 3(b), the red node, the green node, and the blue node have an adjacency relationship in space, so we construct an asynchronous spatial-temporal correlation in the time dimension, which is represented by a red arrow. In addition, the red nodes are related to themselves in time, and we use yellow arrows to connect them in the time dimension.

Formally, the above process can be expressed as follows. We arbitrarily select $m(m \leq S)$ continuous spatial graph $\mathbb{G}_{(t_0:t_m)} = \left(G_{(t_0)}, G_{(t_1)}, \cdots, G_{(t_m)}\right)$ in historical observation data, then the asynchronous spatial-temporal correlation graph formed by them can be expressed as $G_{(H)} = \left(\mathcal{V}_{(H)}, \mathcal{E}_{(H)}, A_{(H)}\right)$, where $\mathcal{V}_{(H)}$ and $\mathcal{E}_{(H)}$ respectively represent the vertex set and edge set in the asynchronous spatial-temporal correlation graph. $A_{(H)}$ is the Asynchronous Spatial-Temporal Correlation Matrix, defined as follows:

$$A_{(H)} = \begin{pmatrix} I & A & 0 & \cdots & 0 & 0 & 0 \\ A & I & A & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & I & A & 0 \\ 0 & 0 & 0 & \cdots & A & I & A \\ 0 & 0 & 0 & \cdots & 0 & A & I \end{pmatrix} \in \mathbb{R}^{mN \times mN} \quad (2)$$

among them, I is the identity matrix, A is the adjacency matrix of the spatial graph, and A is calculated as follows:

$$A_{i,j} = \begin{cases} 1, & \text{if } v_i \text{ connects to } v_j \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where v_i represents the i node in the traffic network. Algorithm 1 describes the construction of ASTCM in detail.

Algorithm 1 Construction of ASTCM

Input : N: the number of Nodes
m: the number of spatial graphs, $m \in [2, S]$
D: the distance between nodes
W, A: $N \times N$ matrix,initial to zero matrix
ϵ: the thresholds
Output: $ASTCM$: a sparse matrix for graph convolution
for each row in D **do**
$i, j, distance \leftarrow (row[0]\ row[1]\ row[2])$
Calculate $W_{i,j}$ as Eq. (15)
end for
$A \leftarrow$ Construct A as Eq. (3)
$ASTCM \leftarrow mN \times mN$ zero matrix
$ASTCM \leftarrow$ Generate ASTCM as Eq. (2)

B. Adaptive Asynchronous Spatial-Temporal Correlation Weight Matrix

In the transportation network, the strength of asynchronous spatial-temporal correlation between nodes is different. Specifically, affected by the factors such as the distance between

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102238, IEEE Internet of Things Journal

nodes and road conditions, the impact of each node on other nodes is non-fixed and difficult to quantify. The adjacency matrix A in the ASTCM only describes the connectivity of the nodes. It cannot accurately describe the strength of the asynchronous spatial-temporal correlation between nodes. To solve this problem, we constructed an Adaptive Asynchronous Spatial-Temporal Correlation Weight Matrix to add adaptive weights to the asynchronous spatial-temporal correlation matrix. The adpASTCWM does not require any prior knowledge and will be used as a model parameter in the training process of the model. The strength of the asynchronous spatial-temporal relationship between nodes is learned end-to-end through stochastic gradient descent.

To reduce the redundant parameters in the model and improve the calculation efficiency of the model, we can use the ASTCM $A_{(H)}$ to initialize the adpASTCWM. The initialization rule can be described as:

$$A_{(adp)i,j} = \begin{cases} \text{Any non-zero value}, & \text{if } A_{(H)i,j} \text{ equal to 1} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

In other words, we force the model to only focus on the strength of asynchronous spatial-temporal relationships between nodes that have adjacency relationships.

C. Asynchronous Spatial-Temporal Graph Convolution

Based on the ASTCM and the adpASTCWM, we define the convolution operation of ADGCN in the spatial domain. For each node, the convolution operation aggregates three types of features: the features of the node itself at current time step, the features of this node at other time steps, and the features of other nodes that are spatially adjacent at other time steps. We call the latter two features asynchronous spatial-temporal features. We perform a linear weighted combination of these three features in the convolution operation, and the weight value is the parameter value learned in the adpASTCWM. Then we apply a linear transformation to the aggregated feature matrix. The graph convolution operation can be expressed as:

$$h_{(l)} = \mathbf{A}X\mathbf{W} + \mathbf{b} \tag{5}$$

$$\mathbf{A} = A_{(adp)} \odot A_{(H)} \tag{6}$$

where $\odot$ means element-wise multiplication, $X \in \mathbb{R}^{mN \times C}$ is the input of the convolutional layer, $\mathbf{W} \in \mathbb{R}^{C \times C'}$ and $\mathbf{b} \in \mathbb{R}^{C'}$ are learnable parameters. In order to expand the receptive field of convolution operation, we stack multiple layers of ASTGC layers together to form an ASTGC block, as shown in the Fig. 4. In addition, to solve the over-smoothing problem caused by the deep graph convolution network, we designed a gated fusion layer to fuse the results of each layer of spatial-temporal convolution operation, as shown in Fig. 5. We denote the output of the spatial-temporal graph convolutional layer of the $1, 2, ..., l$ layer as $h_{(1)}, h_{(2)}, ..., h_{(l)}$ in the gated fusion layer, the output $h_{(l)}$ of each convolutional layer is first cropped. Only the node features at the last time step are retained, and then the cropped features are connected. We use a linear transformation with a gating mechanism to fuse them:

$$h_f = g\left([h_{c(1)}, h_{c(2)}, \ldots, h_{c(l)}]\mathbf{W_f} + \mathbf{b_f}\right) \tag{7}$$

$$h_g = \sigma\left([h_{c(1)}, h_{c(2)}, \ldots, h_{c(l)}]\mathbf{W_g} + \mathbf{b_g}\right) \tag{8}$$

$$h_M = h_f \odot h_g \tag{9}$$

where $\mathbf{W_f} \in \mathbb{R}^{lC' \times C'}$,$\mathbf{W_g} \in \mathbb{R}^{lC' \times C'}$,$\mathbf{b_f} \in \mathbb{R}^{C'}$ and $\mathbf{b_g} \in \mathbb{R}^{C'}$ are learnable parameters, $h_{c(l)} \in \mathbb{R}^{N \times C'}$ is the cropped output feature, $[\cdot]$ represents concatenation operation, $g(\cdot)$ is the activation function, such as relu or tanh, $\sigma(\cdot)$ is the sigmoid function, which determines the ratio of information input to the next layer, and $\odot$ is element-wise multiplication.

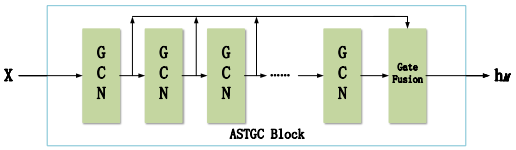


Fig. 4: Asynchronous Spatial-Temporal Graph Convolution Block.

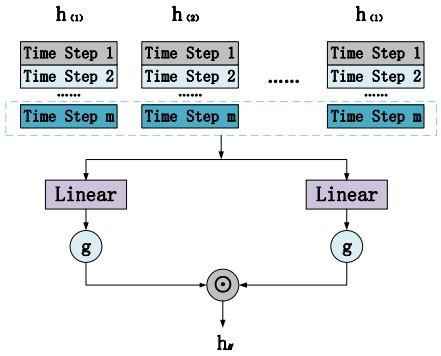


Fig. 5: Gated Fusion Layer.

D. Asynchronous Spatial-Temporal Dilated Causal Convolution Layer

During the process of constructing ASTCM, it is not the optimal choice to connect all spatial graphs in time steps together $(m = S)$, especially if the scale of spatial graphs is huge. The performance of the model will decrease with the increase of the number of connected spatial graphs because the large ASTCM needs a huge parameter matrix as the convolution kernel, which will lead to the model spending much time on the convolution operation. Simultaneously, too many parameters will make the complexity of the model far greater than the complexity of the problem we want to model, thus leading to the occurrence of the overfitting phenomenon. However, the ASTCM constructed with fewer spatial graphs can only extract the short-term asynchronous spatial-temporal correlation of traffic data. It cannot extract the long-term asynchronous spatial-temporal correlation of traffic data.

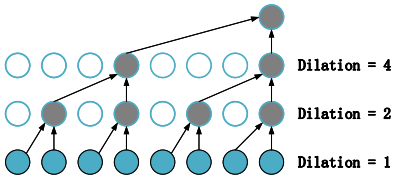


Fig. 6: 1D dilated causal convolution with kernel size 2.

Inspired by applying 1D dilated causal convolution to extract time dependence in Graph WaveNet, we generalized

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102238, IEEE Internet of Things Journal

the 1D dilated causal convolution operation, designed and implemented the ASTDC layer. As shown in Fig. 6, the receptive field of the model can increase exponentially by increasing the dilated factor of each layer in increasing order. We generalize the dilated causal convolution to the asynchronous spatial-temporal graph convolution operation, as shown in Fig. 7.

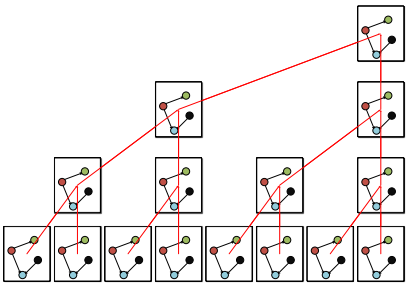


Fig. 7: Asynchronous Spatial-Temporal Dilated Causal Convolution.

At each layer, the ADGCN model determines the number of spatial graphs participating in the construction of ASTCM according to the value of m. In addition, the model chooses to skip the spatial graph at which time step according to the size of the dilated factor d. Since the spatial-temporal pattern presented by traffic data in each period is not the same, in each layer of ASTDC layer, we use multiple ASTGC blocks to extract the information in different asynchronous spatial-temporal correlation graphs:

$$h_M^i = ASTGCBlock_i(\mathbf{X}[t, t-d\times 1, \ldots, t-d\times m]) \quad (10)$$

where h_M^irepresents the output of the i-th ASTGC block of this layer, and the output of the ASTDC layer is obtained by stacking all the outputs of the ASTGC block of this layer together:

$$H^s = Stack\left[h_M^1, h_M^2, \ldots, h_M^{T-d\times m}\right] \quad (11)$$

where H^s is the output of the s ASTDC layer. We stack multiple ASTDC layers to extract a large range of asynchronous spatial-temporal correlations without a sharp increase in network depth. At the same time, the calculation time of the model is also greatly reduced. Subsequently, we built an output layer to generate predicted values for T time steps.

E. Output Layer

We utilize a two-layer fully connected network as the output layer of the ADGCN model. The model can output the result of T time steps at one time instead of iteratively calculating and generating T steps. Therefore, the prediction output of each time step does not depend on the prediction output of the previous time step, avoiding the accumulation of prediction errors, as shown below:

$$\hat{y} = \mathrm{ReLU}\left(H^{last}\mathbf{W_1} + \mathbf{b_1}\right)\cdot \mathbf{W_2} + \mathbf{b_2} \quad (12)$$

where $\hat{y} \in \mathbb{R}^{N\times T}$ is the predicted output of the model. $H^{last} \in \mathbb{R}^{N\times T_o\times C'}$ is generated from the last layer of ASTDC layer, the value of T_o is related to the depth of the asynchronous spatial-temporal dilated convolutional network, the dilated factor d of each layer, and the number of connected spatial graphs m. $\mathbf{W_1} \in \mathbb{R}^{T_o\times C'\times D}$, $\mathbf{W_2} \in \mathbb{R}^{D\times T}$, $\mathbf{b_1} \in \mathbb{R}^D$, $\mathbf{b_2} \in \mathbb{R}^T$ are trainable parameters.

F. Framework of ADGCN

As shown in Fig. 2, we use a layer of full connection layer as the input layer of the ADGCN model. The input data is first transformed into a high-dimensional space. The asynchronous spatial-temporal correlation in the data is then extracted through the multi-layer ASTDC layer. Each of ASTDC layers is composed of $S^s - d^s$ ASTGC blocks, which are used to extract the characteristic information of each asynchronous spatial-temporal graph respectively, where S^s and d^s are the input sequence length and dilated factor of this layer, respectively. Finally, the output of the multi-layer ASTDC layer is sent to the output layer composed of two fully connected layers to produce the predicted output of T time steps.

We select Mean Absolute Error (MAE) as the loss function of ADGCN, which is defined by

$$L = \frac{1}{TND}\sum_{s=t+1}^{t+T}\sum_{n=1}^{N}\sum_{d=1}^{D}\left|\hat{X}_{nd}^{(s)} - X_{nd}^{(s)}\right| + \lambda\|\Theta\|_2 \quad (13)$$

where *T*, *N*, *D* represent the number of predicted time steps, the number of nodes in a spatial graph G, and the dimension of features, respectively. To avoid overfitting, we apply the L2 regularization term $\lambda\|\Theta\|_2$ and λ is a hyperparameter. Detailed process of ADGCN is shown as Algorithm 2.

Algorithm 2 ADGCN

Input : N: the number of Nodes
m: the number of spatial graphs, $m \in [2, S]$
$ASTCM$: generated by Algorithm 1
L: the number of ASTGC layers
S, T: input / output sequence length
X: traffic data sequence, $X \in R^{N\times S\times D}$
$dilations$: the list of dilation factors
Output: Y: the predicted output of the model, $Y \in R^{N\times T}$

```
adpASTCWM ← initial by ASTCM
X ← Perform a linear transformation on X
for l = 1, 2, ..., L do
    d_l ← dilations_l
    for i = 1, 2, ..., S − d_l × (m − 1) do
        Perform the ASTGC Block as Eqs. (5) − (10)
    end for
    H^s ← Stack the results of ASTGC Blocks as Eq. (11)
    X ← H^s
    S ← S − d_l × (m − 1)
end for
H^last ← X
Y ← Perform the output layer as Eq. (12)
```

V. EXPERIMENTS

We conducted experiments on three real-world datasets and verified the effectiveness of the ADGCN model by comparing it with several traditional traffic flow prediction models, and the latest traffic flow prediction model using deep learning.

A. Dataset Description

We validated our model on three public traffic datasets, PeMS04, PeMS08 and METR-LA. PeMS04 and PeMS08 are collected in real-time every 30 seconds by Caltrans Performance Measurement System [43]. The system deploys more

TABLE III: Summary statistics of datasets

Data	Nodes	Time range
PEMS04	307	1/1/2018 - 2/28/2018
PEMS08	170	7/1/2016 - 8/31/2016
METR-LA	207	3/1/2012 - 6/30/2012

than 39000 traffic detectors on highways in major metropolitan areas of California. METR-LA records four months of statistics on traffic speed on 207 sensors on the highways of Los Angeles County [34]. All the above datasets are public and can be obtained through the cited literature.

- PeMS04: This traffic dataset contains traffic information collected from 3848 detectors on 29 roads in the San Francisco Bay Area, and the time is from January to February 2018.
- PeMS08: Traffic data in the San Bernardino area includes 1979 detectors on 8 roads. The data were collected from July to August 2016.
- METR-LA: This traffic dataset collected from loop detectors in the highway of Los Angeles County. The data was collected from Mar 1st 2012 to Jun 30th 2012.

Table III shows the details of the processed datasets. In these three datasets, we aggregate the average vehicle speed readings into 5 minutes window. In the experiment, Z-score normalization is applied to the aggregated datasets; that is, the average value of the data is set to 0, and the standard deviation is 1. The formula is given as follows:

$$Z = \frac{X - \bar{X}}{S} \tag{14}$$

where X is the sample to be processed, $\bar{X}$ represents the average value of the sample, and S is the standard deviation of the sample. In addition, we divide 70% data into training set, 20% data into test set, and the remaining 10% data is used for validation set. We calculated the distance of the paired road network between the sensors and removed the detectors that were too close, and then used the threshold Gaussian kernel [38] to construct the adjacency matrix. The calculation formula is given as follows:

$$W_{ij} = \begin{cases} \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) & d_{ij} \leq \epsilon \\ 0 & d_{ij} > \epsilon \end{cases} \tag{15}$$

where d_{ij} represents the distance between detector i and detector j, σ is the standard deviation of the distance, and ϵ is the threshold. The larger the W_{ij}, the more relevant the vertex i and the vertex j. The values of σ and ϵ are the same as the settings in Graph WaveNet [6].

B. Baselines

We compare ADGCN with the following models.

- HA [44]: The Historical Average method, which models the traffic flow as a periodic process, and uses the weighted average of the previous periods as the predicted value.
- SVR [45]: Support Vector Regression. It is a variant method of support vector machine model. In this article, we use linear kernel function for traffic prediction.
- FNN: Feedforward Neural Network, using two hidden layers and L2 regularization.
- FC-LSTM [46]: Multi-layer long short-term memory network for sequence-to-sequence learning
- T-GCN [33]: Temporal Graph Convolutional Network, a deep learning model that combines graph convolutional network and gated recurrent unit for traffic prediction.
- DCRNN [34]: Diffusion Convolutional Recurrent Neural Network, this method models the traffic flow as a diffusion process on a directed graph.
- Graph WaveNet [6]: A powerful model which combines diffusion convolution and adaptive adjacency matrix to extract spatial dependence, and uses dilated causal convolution to deal with temporal dependence.
- ISTD-GCN [35]: Iterative Spatial-Temporal Diffusion Graph Convolutional Network, this method synchronously models the temporal dependence and spatial dependence in traffic data.
- GMAN [47]: Graph Multi-Attention Network, this model adapts an encoder-decoder architecture, and uses an attention mechanism to model the impact of spatial-temporal factors on traffic conditions.

C. Evaluation Metrics

We use Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) to evaluate the prediction performance of the ADGCN model:
(1)Mean Absolute Error

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{16}$$

(2)Root Mean Squared Error

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{17}$$

(3)Mean Absolute Percentage Error

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{18}$$

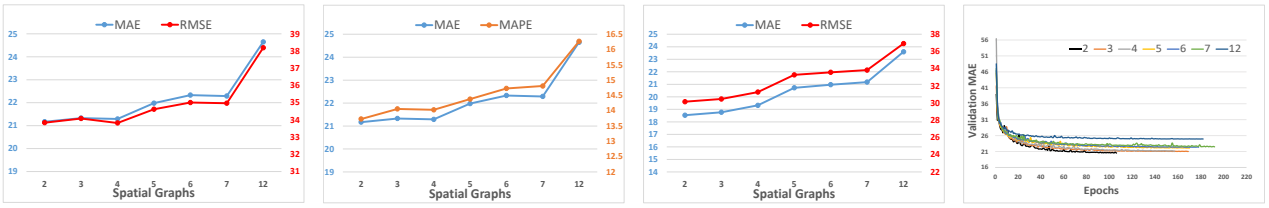
where y_i and $\hat{y}_i$ are the real traffic information and the predicted value of the model at i time step, respectively. n is the number of time steps. Specifically, these three metrics are used to measure the prediction error.

D. Model Parameters Selection

There are six important hyperparameters of the ADGCN model: learning rate, batch size, training epoch, the number of connected spatial graphs in the asynchronous spatial-temporal graph, the dilated factor, and the number of ASTGC layers in the ASTGC block. In the experiment, we apply grid search to adjust and set the learning rate to 0.001, the batch size to 32, and the training epoch to 200.

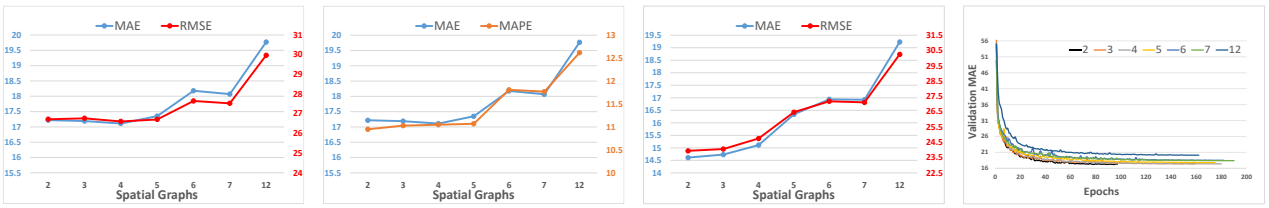
The remaining three parameters are the core parameters of ADGCN. Their different values may significantly affect the prediction performance of the model. We conducted detailed experiments to select the optimal values of these three parameters.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102238, IEEE Internet of Things Journal



(a) (b) (c) (d)

Fig. 8: Comparison of predicted performance over different number of connected spatial graphs in the training and validation set based on PeMS04 dataset. (a) MAE and RMSE on validation set. (b) MAE and MAPE on validation set. (c) MAE and RMSE on training set. (d) MAE downward trend on the validation set during training.



(a) (b) (c) (d)

Fig. 9: Comparison of predicted performance over different number of connected spatial graphs in the training and validation set based on PeMS08 dataset. (a) MAE and RMSE on validation set. (b) MAE and MAPE on validation set. (c) MAE and RMSE on training set. (d) MAE downward trend on the validation set during training.

1) The Number of Connected Spatial Graphs: In our experiment, for the PeMS04 dataset, we connect 2, 3, 4, 5, 6, 7, and all 12 spatial graphs, respectively, and analyze their influence on the performance of the model. Fig. 8 compares the performance of the ADGCN model on the PeMS04 dataset when connecting different numbers of spatial graphs. In Fig. 8(a), Fig. 8(b) and Fig. 8(c), the horizontal axis represents the number of connected spatial graphs, and the vertical axis represents the change of different metrics. The horizontal axis in Fig. 8(d) represents the training epochs, and the vertical axis represents the MAE change trend on the validation set. It can be seen from Fig. 8(a) and Fig. 8(b) that when the number of connected spatial graphs is smaller, the prediction performance of the model is better. This trend is evident in Fig. 8(c). This is mainly because a too large asynchronous spatial-temporal graph will cause the model to require more parameters, which will increase the complexity of the model, and as a result, overfitting occurs. Fig. 8(d) shows that when the number of connected spatial graphs is 2, the model only needs to be trained for 98 epochs to converge. When the number of connected space graphs increases, the model becomes more difficult to train. In summary, we connect 2 spatial graphs to form an asynchronous spatial-temporal graph.

Similarly, the results on the PeMS08 dataset are shown in Fig. 9. We still choose to connect two spatial graphs to form an asynchronous spatial-temporal graph. For the METR-LA dataset, we set the number of connected spatial graphs to 4.

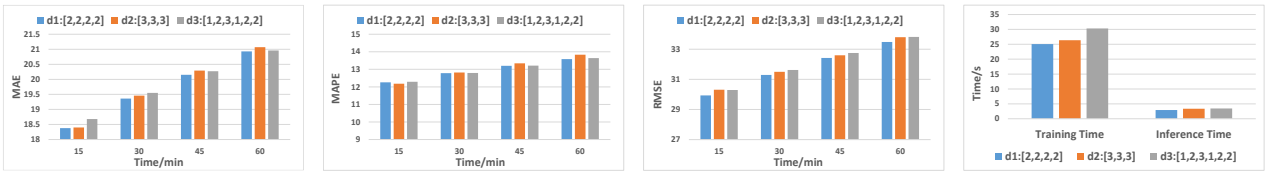
2) Dilated Factor: Using dilated causal convolution, the receptive field of the model can increase exponentially as the number of convolutional layers increases. However, since the dilated causal convolution will skip a part of the spatial graph for feature extraction, some information may be lost. Therefore, it is very important to choose an appropriate dilated factor. We try to keep the model in balance between training time and prediction performance. Specifically, in our experiment, for PeMS04 dataset, we take 3 sets of dilated factors for comparison, which are $[2, 2, 2, 2]$, $[3, 3, 3]$ and $[1, 2, 3, 1, 2, 2]$. The experimental results are shown in Fig. 10.

Fig. 10(a), Fig. 10(b) and Fig. 10(c) respectively show the influence of different dilated factors on the prediction performance of the ADGCN model. Fig. 10(d) shows the influence of different dilated factors on the prediction time and reasoning time of the model. From the Fig. 10, it can be obviously found that when the dilated factor is $[2, 2, 2, 2]$, the prediction performance and time consumption of the model have achieved the best results.

Fig. 11 shows the experimental results of using three groups of dilated factors $[1, 2, 1, 2, 2]$,$[1, 2, 3, 4]$ and $[1, 2, 3, 1, 2, 2]$ on the PeMS08 dataset. Considering the calculation time and prediction performance, we set the dilated factor as $[1, 2, 3, 4]$ for the PeMS08 dataset. For the METR-LA dataset, we set the dilated factor to $[1, 2]$.

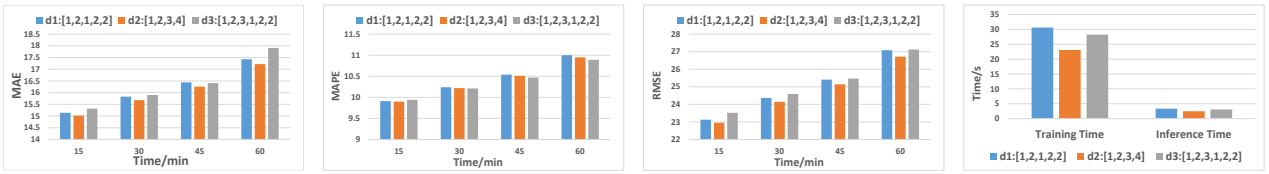
3) The Number of ASTGC Layers: For a node on the graph, there are many types of neighbors. As shown in Fig. 12, take the red node as an example, the green nodes are its first-order neighbor, and the black nodes are the second-order neighbor of the red node. And so on, the yellow nodes are the third-order neighbor of the red node. In our model ADGCN, each additional layer of the ASTGC layer in the ASTGC block can aggregate one more order of neighbor information. For the PeMS04 dataset, We select the number of ASTGC layers from 2,3,4,5,8,10,12 and analyze the change of prediction performance. The experimental results are plotted in Fig. 13(a), the horizontal axis represents the number

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102238, IEEE Internet of Things Journal



(a) (b) (c) (d)

Fig. 10: Comparison of predicted performance over different dilated factors on PeMS04 dataset. (a) MAE of different dilated factors over multiple prediction time steps. (b) MAPE of different dilated factors over multiple prediction time steps. (c) RMSE of different dilated factors over multiple prediction time steps. (d) The influence of different dilated factors on model training time(s/epoch) and inference time.



(a) (b) (c) (d)

Fig. 11: Comparison of predicted performance over different dilated factors on PeMS08 dataset. (a) MAE of different dilated factors over multiple prediction time steps. (b) MAPE of different dilated factors over multiple prediction time steps. (c) RMSE of different dilated factors over multiple prediction time steps. (d) The influence of different dilated factors on model training time(s/epoch) and inference time.

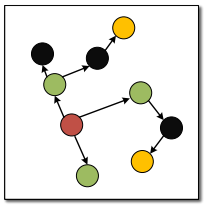
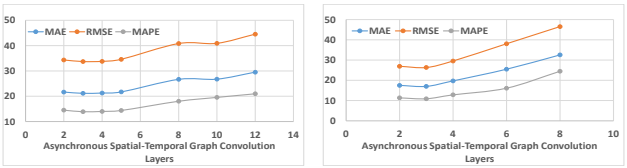


Fig. 12: K-order neighbors of nodes on the graph.



(a) (b)

Fig. 13: The number of ASTGC layers. (a) 60-minute prediction performance on PeMS04 dataset. (b) 60-minute prediction performance on PeMS08 dataset.

of ASTGC layers, and the vertical axis represents different evaluation metrics. We can observe that when the number of layers is 3, the MAE, RMSE and MAPE errors of the model are the smallest. When increasing the ASTGC layer, the prediction performance of the model firstly increases and then decreases sharply. This is because, in the beginning, the model's expressive ability is low, and it is unable to model the spatial-temporal dependence of the data fully. As the number of ASTGC layers increases, the representations of the nodes in ADGCN are inclined to converge to a certain value and thus become indistinguishable [48]. Such a phenomenon is called over-smoothing [49]. Therefore, we set the number of ASTGC layers to 3 in our experiments on the PeMS08 dataset.

In addition, the experimental results on the PEMS08 dataset are shown in Fig. 13(b). It can be seen that the prediction performance is the best when the number of ASTGC layers is 3. For the METR-LA dataset, we set this parameter to 4.

E. Experimental Results

First, we evaluate the overall prediction performance of ADGCN and other benchmark models. Table IV shows the average values of MAE, RMSE, and MAPE of various models for multi-step traffic flow predictions on PeMS04, PeMS08 and METR-LA datasets. The bold part indicates that the model has the best performance. It can be seen that the ADGCN model is superior to almost all other prediction models, including traditional methods and the latest deep learning methods.

In addition, we compare the short-term, medium-term, and long-term prediction performance of various models. Table V shows the performance of the ADGCN model and other benchmark models for 15 minutes, 30 minutes, and 60 minutes prediction tasks on PeMS04, PeMS08 and METR-LA datasets. As shown from the table, the ADGCN model has achieved the best prediction performance for almost all prediction horizons, especially in terms of 60-minute prediction performance, which is greatly improved compared with other benchmark models. On the METR-LA dataset, the performance of ADGCN and ISTD-GCN are very similar, but the time efficiency of ADGCN is better than that of ISTD-GCN, which we will analyze later.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102238, IEEE Internet of Things Journal

TABLE IV: Comparison of average prediction errors of different models

Models	PEMS04			PEMS08			METR-LA		
	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
HA	28.22	41.85	19.99	23.08	34.18	14.52	4.16	7.80	13.0
SVR	33.79	52.51	20.31	28.20	43.62	15.54	5.25	11.02	12.7
FNN	24.41	37.18	17.47	19.36	29.55	13.66	4.23	8.26	12.2
FC-LSTM	23.76	36.35	17.46	18.52	28.44	11.99	3.86	7.40	11.23
T-GCN	21.42	32.91	15.71	17.98	26.98	14.37	3.89	6.57	10.56
DCRNN	21.76	33.99	14.70	16.28	25.40	10.52	3.17	6.47	8.86
Graph Wavenet	21.66	34.26	14.85	17.14	27.59	11.06	3.09	6.24	8.42
ISTD-GCN	21.68	35.32	15.63	18.51	28.09	12.36	2.80	**5.0**	**7.59**
GMAN	21.54	33.28	16.30	16.26	25.10	12.17	3.06	6.29	8.51
ADGCN	**19.55**	**31.39**	**12.81**	**15.90**	**24.46**	**10.28**	**2.79**	5.49	7.77

TABLE V: Comparison of prediction errors of different models at multiple time steps

Data	Models	MAE			RMSE			MAPE(%)		
		15min	30min	60min	15min	30min	60min	15min	30min	60min
PEMS04	HA	28.22	28.22	28.22	41.85	41.85	41.85	19.99	19.99	19.99
	SVR	29.60	31.54	40.23	47.08	48.45	62.02	17.40	19.11	24.43
	FNN	22.01	22.90	30.69	32.88	35.32	46.33	16.67	16.06	21.51
	FC-LSTM	22.25	22.04	29.18	33.22	34.33	44.62	16.68	15.77	20.35
	T-GCN	19.56	20.96	23.75	30.58	32.38	35.78	13.76	15.04	18.33
	DCRNN	19.20	21.22	24.87	30.44	33.27	38.28	12.79	14.21	17.11
	Graph Wavenet	18.67	21.37	26.13	29.88	33.87	40.73	12.61	14.54	18.48
	ISTD-GCN	20.15	21.33	22.94	32.88	34.54	38.08	14.78	15.34	16.39
	GMAN	20.06	21.37	24.13	31.12	33.11	36.90	15.12	16.07	18.36
	ADGCN	**18.38**	**19.36**	**20.90**	**29.43**	**30.94**	**33.32**	**12.24**	**12.78**	**13.58**
PEMS08	HA	23.08	23.08	23.08	34.18	34.18	34.18	14.52	14.52	14.52
	SVR	23.15	25.90	35.56	35.71	40.21	54.94	13.23	14.29	19.11
	FNN	17.35	18.03	24.72	25.55	28.04	37.83	12.87	12.79	16.54
	FC-LSTM	17.54	16.90	23.17	26.16	26.51	35.43	11.43	10.94	14.90
	T-GCN	16.38	18.01	19.55	24.72	27.11	29.11	12.69	14.58	15.86
	DCRNN	**14.50**	15.93	18.42	**22.67**	25.01	28.53	**9.31**	10.26	11.98
	Graph Wavenet	14.84	16.61	19.98	23.73	26.87	32.18	9.60	10.53	13.06
	ISTD-GCN	16.58	18.14	20.23	25.21	27.61	30.56	11.98	13.26	15.02
	GMAN	15.21	15.98	18.40	23.44	24.86	28.17	11.29	11.87	13.81
	ADGCN	15.02	**15.68**	**16.81**	22.95	**24.15**	**26.03**	9.90	**10.21**	**10.73**
METR-LA	HA	4.16	4.16	4.16	7.80	7.80	7.80	13.0	13.0	13.0
	SVR	3.99	5.05	6.72	8.45	10.87	13.76	9.3	12.1	16.7
	FNN	3.99	4.23	4.49	7.94	8.17	8.69	9.9	12.9	14.0
	FC-LSTM	3.44	3.77	4.37	6.30	7.23	8.69	9.6	10.9	13.2
	T-GCN	3.46	3.80	4.43	5.68	6.29	7.20	8.99	10.40	12.29
	DCRNN	2.77	3.15	3.60	5.38	6.45	7.59	7.3	8.8	10.5
	Graph Wavenet	2.69	3.07	3.53	5.15	6.22	7.37	6.90	8.37	10.01
	ISTD-GCN	**2.50**	2.81	**3.10**	**4.40**	**5.03**	**5.68**	**6.50**	**7.37**	**8.91**
	GMAN	2.77	3.09	3.44	5.49	6.42	7.35	7.35	8.62	10.04
	ADGCN	**2.50**	**2.79**	3.19	4.70	5.50	6.61	6.63	7.71	9.44

1) Prediction Accuracy Analysis: We can find that the methods based on deep neural networks, such as FNN, FC-LSTM, T-GCN, DCRNN, Graph WaveNet, ISTD-GCN, and our model ADGCN, are superior to the traditional linear methods including HA and SVR in multi-step traffic flow prediction tasks. This is mainly due to the complex and dynamic temporal dependence in traffic data. Traditional models such as HA and SVR are difficult to deal with this temporal dependence. FC-LSTM only considers the temporal correlation of traffic information and does not utilize the spatial dependence in the spatial-temporal network. Hence, its prediction performance has a large gap compared with the spatial-temporal models such as T-GCN, Graph WaveNet, and ADGCN. This shows that the modeling of spatial dependence is crucial for the traffic prediction task.

We plot predicted values v.s real values of ADGCN model on the PeMS08 dataset in Fig. 15 and Fig. 16. Specifically, we randomly select 4 sensor nodes and arbitrary 864 continuous-time steps (3 days) to compare the real value with the predicted value. The comparison result is shown in Fig. 15. In addition, in order to evaluate the overall prediction performance of the model on the traffic network, we compare the real average velocity of all detector nodes randomly selected in 864 continuous-time steps with the average speed predicted by our model, as shown in Fig. 16.

It can be seen that our model ADGCN can extract local features well and adapt to the complex speed change patterns of different nodes. Specifically, in the traffic network, the speed of different road sections has different variation patterns. For example, the change of node speed in Fig. 15(a) is relatively stable, while the node speed in Fig. 15(b), Fig. 15(c) and Fig. 15(d) fluctuates greatly. ADGCN can extract complex velocity change patterns from different nodes, thus achieving better prediction performance. The traffic flow often shows obvious periodicity (early peak, evening peak, and other). ADGCN can identify the beginning and end time of the peak period and make a prediction result similar to the actual traffic speed. From a macro perspective, as shown in Fig. 16, ADGCN can

accurately predict the traffic speed of the entire traffic network.

2) Long-term Prediction Ability Analysis: We found that the time step of prediction has a more significant impact on the prediction performance of the model. The prediction accuracy of some models will drop sharply as the prediction step increases. As shown in Table V, the performance of our model ADGCN and other benchmark models on long-term prediction tasks is lower than short-term prediction tasks. Note that since the HA method does not rely on short-term data, its prediction performance does not change with a small increase in the prediction time step. Fig. 14 shows the MAE and MAPE trend of the T-GCN, ISTD-GCN, Graph WaveNet, and our ADGCN model over different time steps on PeMS04 and PeMS08 datasets. It can be seen from the Fig. 14 that no matter how much the prediction time step increases, the ADGCN model almost always obtains the best prediction performance and the prediction errors have less tendency to change. On the contrary, the performance of other models on long-term prediction tasks is unsatisfactory. This shows that our ADGCN model is not sensitive to the prediction horizons. Accurate long-term traffic prediction can provide a reliable basis for traffic planning tasks, and our ADGCN model can accomplish this well.

3) Computation Time Analysis: We compared the computation time of ADGCN with DCRNN, GMAN, and ISTD-GCN on the METR-LA dataset in Table VI. ADGCN runs faster than DCRNN, GMAN and ISTD-GCN. Especially ISTD-GCN, its runs ten times slower than ADGCN in training and inference. The time cost of DCRNN is mainly used for iterative calculation based on RNN components. For GMAN, it takes a lot of time to calculate spatial attention and temporal attention. In addition, the demand for GPU resources is relatively large. The iterative strategy in ISTD-GCN requires T-1 (T is the length of the input time series) iterative calculation, which will increase the computation time of the model. In ADGCN, our ASTCM and adpASTWCM are sparse, hence sparse-dense matrix multiplication can be used to reduce the computation time of graph convolution. Eq. (19) calculates the sparsity of these two matrices. K_0 is the number of zero elements in the matrix, $M \times N$ is the total elements of the matrix. The sparsity of ASTCM used in METR-LA is 0.97. In addition, ASTDC also avoids unnecessary redundant calculations, which greatly reduces the computation time of the model. We discussed the effect of this component in the ablation experiment.

$$S_M = \frac{K_0}{M \times N} \tag{19}$$

TABLE VI: The computation cost on the METR-LA dataset

Model	Training(s/epoch)	Inference(s)
DCRNN	249.31	18.73
GMAN	611.20	28.20
ISTD-GCN	890.40	52.31
ADGCN	61.20	7.27

F. Ablation Experiments

We designed and conducted ablation experiments to verify the effectiveness of several key components, including the ASTCM, the adpASTCWM, and the ASTDC. Table VII shows the average prediction performance of ADGCN and its multiple variants on the PeMS04, PeMS08 and METR-LA datasets. Fig. 19 shows the MAE trend of these models over different time steps on METR-LA.

1) Effect Analysis of Asynchronous Spatial-Temporal Correlation Matrix: To verify the effect of the ASTCM in ADGCN, we replace it with a standard spatial-temporal correlation matrix, called ADGCN-noASTM, which only focuses on the spatial information of different nodes in the same time step and the temporal information of the same node in different time steps. It can be seen from Table VII that compared with the ADGCN-noASTM model, the ADGCN model achieves a lower prediction error. This result means that the ASTCM can effectively extract the asynchronous spatial-temporal correlation from traffic data, proving the effectiveness of the ASTCM.

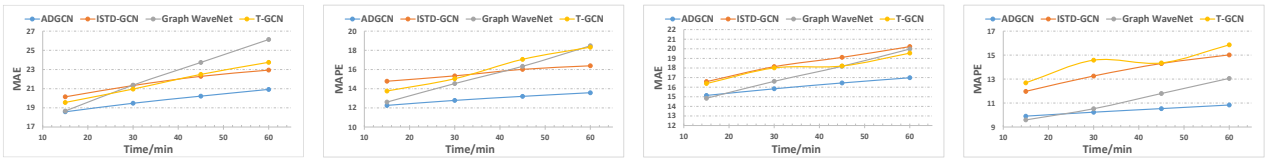
2) Effect Analysis of Adaptive Asynchronous Spatial-Temporal Correlation Weight Matrix: To verify the effect of the adpASTCWM, we replace it with a fixed unit matrix, which we call ADGCN-noAdpASTWM. It can be seen from table VII that the adpASTWM greatly improves the prediction performance of the model. Due to the high complexity of the spatial-temporal correlation of traffic data, fixed weight can not reflect the complex asynchronous spatial-temporal correlation between different nodes.

In Fig. 17 and Fig. 18 we further study the asynchronous spatial-temporal correlation in ADGCN. Fig. 17 is the ASTCM used by ADGCN on the METR-LA dataset. In the parameter selection experiment above, we connected four spatial graphs to form an asynchronous spatial-temporal graph. Fig 18 is a part of adpASTCWM on METR-LA. Here, Fig. 18(a) is in the initial state, Fig. 18(b) is the state after training. According to our definition, in ASTCM, areas I only pay attention to the information of the node itself in their respective time steps. Areas A represents the asynchronous spatial-temporal correlation between nodes.

We can draw the following conclusions from Fig. 18 : (1) ADGCN can effectively capture the asynchronous spatial-temporal correlations in traffic data. (2) the strength of asynchronous spatial-temporal correlation among nodes is different. Some nodes influence multiple nodes, while some nodes hardly interact with other nodes. (3) The weights of nodes in spatial graphs of different time steps are also different. For example, in Fig. 18(b), the weight value of the heatmap in different area is quite different. Therefore, the ASTCM and adpASTCWM components can reflect the complex asynchronous spatial-temporal correlation in traffic data. In addition, it can be seen from Fig. 19 that these two components play an essential role in multi-time step prediction. The results of ablation experiments confirmed the effectiveness of these two components.

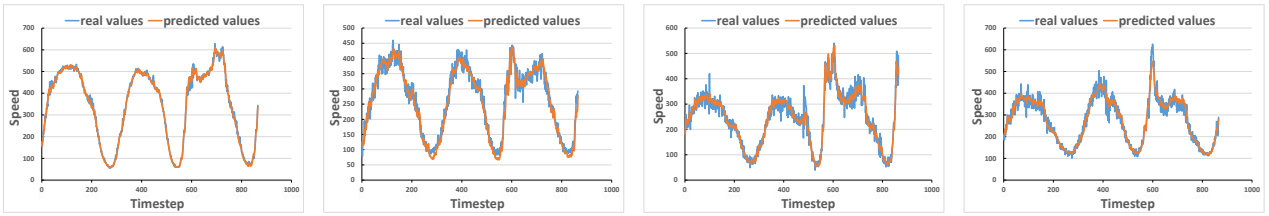
3) Effect of the Asynchronous Spatial-Temporal Dilated Causal Convolution: We removed the ASTDC in ADGCN to verify the effectiveness of this component. Specifically, we use an iterative calculation method similar to RNN to extract asynchronous spatial-temporal correlations in the data called ADGCN-iter. The experimental results are shown in Table VIII. We compared the 60-minute prediction error and computation time of ASTDC and iterative calculation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102238, IEEE Internet of Things Journal



(a) (b) (c) (d)

Fig. 14: Long-term prediction ability. (a) MAE trend of ADGCN model and other models on PeMS04 dataset. (b) MAPE trend of ADGCN model and other models on PeMS04 dataset. (c) MAE trend of ADGCN model and other models on PeMS08 dataset. (d) MAPE trend of ADGCN model and other models on PeMS08 dataset.



(a) (b) (c) (d)

Fig. 15: Comparison of the predicted values and real values of 4 randomly selected nodes($a-d$) in 864 continuous time steps.

TABLE VII: Ablation Experiments

Models	PEMS04			PEMS08			METR-LA		
	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
ADGCN-noASTM	20.15	33.40	13.20	17.28	26.73	11.21	2.90	5.66	7.89
ADGCN-noAdpASTWM	20.78	34.90	13.99	22.46	32.66	14.75	3.06	6.04	8.80
ADGCN	**19.55**	**31.39**	**12.81**	**15.90**	**24.46**	**10.28**	**2.79**	**5.49**	**7.77**

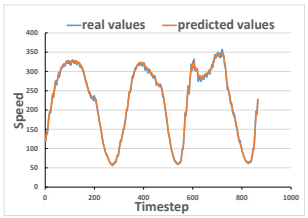


Fig. 16: Comparison of the average predicted value and real value of all nodes in 864 time steps.

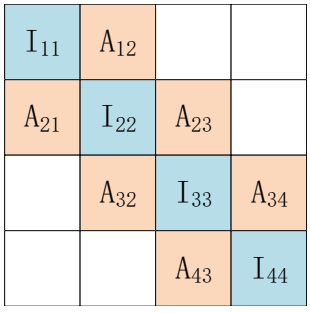
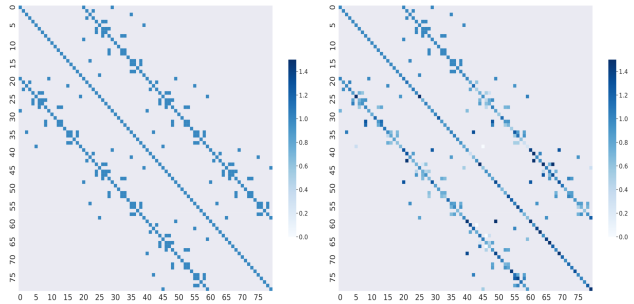


Fig. 17: ASTCM for METR-LA.



(a) (b)

Fig. 18: Heatmap of adpASTCWM with 20 nodes in METR-LA. (a) Initial state. (b) Learned weights.

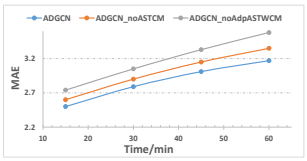


Fig. 19: MAE trend in ablation experiments on METR-LA.

on the PEMS04 dataset. It can be seen that the prediction performance of ADGCN is better than that of ADGCN-iter. Moreover, no matter the training time or the inference time, ADGCN is far less than ADGCN-iter. This proves that the ASTDC can effectively model the spatial-temporal dependence of the data and greatly reduce the calculation time of the model.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102238, IEEE Internet of Things Journal

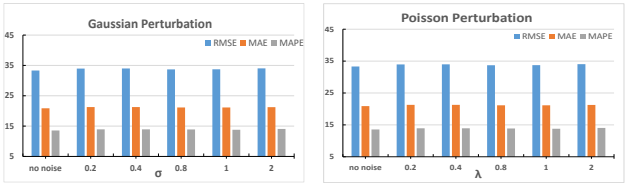
TABLE VIII: Asynchronous Spatial-Temporal Dilated Causal Convolution v.s iterative calculation

Model	MAE	RMSE	MAPE	Training(s/epoch)	Inference(s)
ADGCN	20.9	33.32	13.58	25.05	2.93
ADGCN-iter	21.67	34.41	14.52	79.95	9.52

G. Robustness Experiments

In the real world, the data collection process will inevitably produce noise. We test the robustness of the ADGCN model through perturbation analysis experiments.

We added two common random noises to the data. The first one is random noise that obeys the Gaussian distribution $N \in \left(0, \sigma^2\right)$, where $\sigma \in (0.2, 0.4, 0.8, 1, 2)$; the second type is random noise that obeys the Poisson distribution $P(\lambda)$, where $\lambda \in (1, 2, 4, 8, 16)$. The experimental results are shown in Fig. 20. Fig. 20(a) shows the results before and after adding Gaussian noise to the PeMS04 data set, with the horizontal axis indicating that the σ, the vertical axis is the value of each evaluation metric. Similarly, Fig. 20(b) shows the results before and after adding Poisson noise to the PeMS04 data set. It can be seen that: (1) the decline of the prediction performance of the ADGCN model after adding noise is negligible; (2) regardless of the noise distribution, the change of each evaluation metric is not obvious. This proves that the ADGCN model is robust and can overcome the disturbance caused by noise.



(a) (b)

Fig. 20: Robustness analysis. (a) Performance changes before and after adding Gaussian noise to the PeMS04 data set. (b) Performance changes before and after adding Poisson noise to the PeMS04 data set.

VI. Conclusion

In this paper, we designed an ASTGC operation to mine the asynchronous spatial-temporal relationship in the transportation network. In addition, we extended the 1D dilated causal convolution to the graph convolution network (called ASTDC), which can learn long-term dependencies on data. Finally, the Asynchronous Dialted Graph Convolutional Network (ADGCN) model was proposed to handle traffic flow prediction issues with the above two tricks. Experiments on three real traffic datasets show that the prediction performance of our proposed model is better than other existing traffic data prediction methods, especially in the performance of long-term predictions. Moreover, this model is also suitable for other scenarios with asynchronous spatial-temporal correlations, such as social networks and recommendation systems. In the future, we will further improve the model to achieve real-time prediction of urban traffic flow.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (No. 62072216), and the Jiangsu Agriculture Science and Technology Innovation Fund (No. CX (19)3087).

References

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things journal*, vol. 1, no. 1, pp. 22–32, 2014.

[2] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[3] R. Jia, P. Jiang, L. Liu, L. Cui, and Y. Shi, "Data driven congestion trends prediction of urban transportation," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 581–591, 2017.

[4] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D.-Y. Yeung, "Gaan: Gated attention networks for learning on large and spatiotemporal graphs," *arXiv preprint arXiv:1803.07294*, 2018.

[5] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in *International Conference on Neural Information Processing*. Springer, 2018, pp. 362–373.

[6] Wu, Zonghan and Pan, Shirui and Long, Guodong and Jiang, Jing and Zhang, Chengqi, "graph wavenet for deep spatial-temporal graph modeling," *arXiv preprint arXiv:1906.00121*, 2019.

[7] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, "A comprehensive survey on traffic prediction," *arXiv preprint arXiv:2004.08555*, 2020.

[8] G. Dudek, "Pattern-based local linear regression models for short-term load forecasting," *Electric Power Systems Research*, vol. 130, pp. 139–147, 2016.

[9] M. M. Hamed, H. R. Al-Masaeid, and Z. M. B. Said, "Short-term prediction of traffic volume in urban arterials," *Journal of Transportation Engineering*, vol. 121, no. 3, pp. 249–254, 1995.

[10] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining kohonen maps with arima time series models to forecast traffic flow," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307–318, 1996.

[11] S. Lee and D. B. Fambro, "Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting," *Transportation Research Record*, vol. 1678, no. 1, pp. 179–188, 1999.

[12] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of transportation engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[13] H. Sun, C. Zhang, and B. Ran, "Interval prediction for traffic time series using local linear predictor," in *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No. 04TH8749)*. IEEE, 2004, pp. 410–415.

[14] C. P. Van Hinsbergen, T. Schreiter, F. S. Zuurbier, J. Van Lint, and H. J. Van Zuylen, "Localized extended kalman filter for scalable real-time traffic state estimation," *IEEE transactions on intelligent transportation systems*, vol. 13, no. 1, pp. 385–394, 2011.

[15] X.-l. Zhang, G.-g. HE, and H.-p. LU, "Short-term traffic flow forecasting based on k-nearest neighbors non-parametric regression," *Journal of Systems Engineering*, vol. 24, no. 2, pp. 178–183, 2009.

[16] Z.-s. YAO, C.-f. SHAO, and Y.-l. GAO, "Research on methods of short-term traffic forecasting based on support vector regression [j]," *Journal of Beijing Jiaotong University*, vol. 30, no. 3, pp. 19–22, 2006.

[17] S. Sun, C. Zhang, and G. Yu, "A bayesian network approach to traffic flow forecasting," *IEEE Transactions on intelligent transportation systems*, vol. 7, no. 1, pp. 124–132, 2006.

[18] D. Park and L. R. Rilett, "Forecasting freeway link travel times with a multilayer feedforward neural network," *Computer-Aided Civil and Infrastructure Engineering*, vol. 14, no. 5, pp. 357–367, 1999.

[19] J. Van Lint, S. Hoogendoorn, and H. J. van Zuylen, "Freeway travel time prediction with state-space neural networks: modeling state-space dynamics with recurrent neural networks," *Transportation Research Record*, vol. 1811, no. 1, pp. 30–39, 2002.

[20] R. Fu, Z. Zhang, and L. Li, "Using lstm and gru neural network methods for traffic flow prediction," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*. IEEE, 2016, pp. 324–328.

[21] Fang, Weiwei and Cai, Weihong and Fan, Bo and Yan, Jingwen and Zhou, Teng, "kalman-lstm model for short-term traffic flow forecasting," in *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 5. IEEE, 2021, pp. 1604–1608.

[22] Y. Wu and H. Tan, "Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework," *arXiv preprint arXiv:1612.01022*, 2016.

[23] X. Cao, Y. Zhong, Y. Zhou, J. Wang, C. Zhu, and W. Zhang, "Interactive temporal recurrent convolution network for traffic prediction in data centers," *IEEE Access*, vol. 6, pp. 5276–5289, 2017.

[24] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "Dnn-based prediction model for spatio-temporal data," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2016, pp. 1–4.

[25] D. Wang, J. Zhang, W. Cao, J. Li, and Y. Zheng, "When will you arrive? estimating travel time based on deep neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[26] A. Zonoozi, J.-j. Kim, X.-L. Li, and G. Cong, "Periodic-crn: A convolutional recurrent model for crowd density prediction with recurring periodic patterns." in *IJCAI*, 2018, pp. 3732–3738.

[27] Lu, Huakang and Ge, Zuhao and Song, Youyi and Jiang, Dazhi and Zhou, Teng and Qin, Jing, "a temporal-aware lstm enhanced by loss-switch mechanism for traffic flow forecasting," *Neurocomputing*, vol. 427, pp. 169–178, 2021.

[28] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[29] Jiang, Weiwei and Luo, Jiayun, "graph neural network for traffic forecasting: A survey," *arXiv preprint arXiv:2101.11174*, 2021.

[30] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv preprint arXiv:1709.04875*, 2017.

[31] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3656–3663.

[32] C. Chen, K. Li, S. G. Teo, X. Zou, K. Wang, J. Wang, and Z. Zeng, "Gated residual recurrent graph neural networks for traffic prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 485–492.

[33] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[34] Li, Yaguang and Yu, Rose and Shahabi, Cyrus and Liu, Yan, "diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[35] Y. Xie, Y. Xiong, and Y. Zhu, "Istd-gcn: Iterative spatial-temporal diffusion graph convolutional network for traffic speed forecasting," *arXiv preprint arXiv:2008.03970*, 2020.

[36] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *arXiv preprint arXiv:1812.08434*, 2018.

[37] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2008.

[38] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[39] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[40] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, pp. 3844–3852, 2016.

[41] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5115–5124.

[42] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[43] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia, "Freeway performance measurement system: mining loop detector data," *Transportation Research Record*, vol. 1748, no. 1, pp. 96–102, 2001.

[44] B. L. Smith and M. J. Demetsky, "Traffic flow forecasting: comparison of modeling approaches," *Journal of transportation engineering*, vol. 123, no. 4, pp. 261–266, 1997.

[45] W. W. Wei, "Time series analysis," in *The Oxford Handbook of Quantitative Methods in Psychology: Vol. 2*, 2006.

[46] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, pp. 3104–3112, 2014.

[47] Zheng, Chuanpan and Fan, Xiaoliang and Wang, Cheng and Qi, Jianzhong, "gman: A graph multi-attention network for traffic prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 1234–1241.

[48] Chen, Ming and Wei, Zhewei and Huang, Zengfeng and Ding, Bolin and Li, Yaliang, "simple and deep graph convolutional networks," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1725–1735 .

[49] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

Tao Qi received the bachelor's degree in computer science and technology from Shandong University of Science and Technology in 2018. He is a graduate student in the School of Artificial Intelligence and Computer at Jiangnan University. His research interests include smart transportation and graph neural networks.

Guanghui Li received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005. He is currently a Professor with the Department of Computer Science, Jiangnan University, Wuxi, China. He has published over 70 papers in journal or conferences. His research interests include edge computing, wireless sensor networks, fault tolerant computing, and nondestructive testing and evaluation. His research was supported by the National Natural Science Foundation of China, Jiangsu Provincial Science and Technology Foundation, and other governmental and industrial agencies.

Liangqiang Chen received the bachelor's degree in the internet of things engineering from Zhejiang University of Technology, Hangzhou, China, in 2017. He is currently a Ph.D. candidate in the School of Artificial Intelligence and Computer Science, Jiangnan University. His current research interest is anomaly detection and streaming data mining in the wireless sensor network.

Yanming Xue received the bachelor's degree from Internet of Things Engineering, Guilin University of Electronic Technology in 2019. He is a graduate student in the School of Artificial Intelligence and Computer at Jiangnan University. His research interests include smart transportation and graph neural network.