IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

STGAT: Spatial-Temporal Graph Attention Networks for Traffic Flow Forecasting

XIANGYUAN KONG[1],WEIWEI XING[1](Member, IEEE),XIANG WEI[1](Member, IEEE)PENG BAO[1](Member, IEEE),JIAN ZHANG[1] and WEI LU[1]

[1]School of Software Engineering, Beijing Jiaotong University, 100044, China

Corresponding author: Wei Lu (e-mail: luwei@bjtu.edu.cn).

This work is supported by the National Natural Science Foundation of China (61876017, 61876018, 61906014, 61702031).

ABSTRACT Traffic flow forecasting is a critical task for urban traffic control and dispatch in the field of transportation, which is characterized by the high nonlinearity and complexity. In this paper, we propose an end-to-end deep learning based dual path framework, i.e., Spatial-Temporal Graph Attention Network (STGAT), for traffic flow forecasting. Specifically, different from previous structure-based approaches, STGAT can be directly generalized to the graph with arbitrary structure. Furthermore, STGAT is capable of handling long temporal sequence by stacking gated temporal convolution layer. The dual path architectures is proposed for taking both potential and existing spatial dependencies into account. By capturing potential spatial dependencies will naturally catch more useful information for forecasting. We design a gated fusion mechanism to combine the outputs from each path. The proposed model can be directly applicable to inductive learning tasks by introducing a graph attention mechanism into spatial-temporal framework, which means our model can be generalized to completely unseen graphs. Moreover, experimental results on two public real-world traffic network datasets, METR-LA and PEMS-BAY, show that our STGAT outperforms the state-of-the-art baselines. Additionally, we demonstrate the proposed model is competent for efficient migration between graphs with different structures.

INDEX TERMS Traffic Flow Forecasting; Spatial-Temporal Graph Neural Networks; Intelligent Transportation Systems

I. INTRODUCTION

WITH the rapid development of machine learning and the emergence of new data resources, examining and predicting traffic conditions in smart cities is becoming more and more accurate [1]. Through the above techniques, the design and management of transport services in smart cities can be flexibly optimized. Traffic flow forecasting is a typical task in the transportation domain, the job of which if to predict the future traffic flow speeds of a sensor network with given historical speeds data and underlying road networks. In practice, according to the length of prediction time, traffic flow forecasting is divided into short-term (5-30 minutes), medium-term (30-60 minutes) and long-term (over an hour) [2]. Most prevalent approaches can perform well on short-term forecast. However, due to the uncertainty and complexity of traffic flow, those methods are less effective for relatively long-term predictions [3]. In this paper, we study traffic flow forecasting problems on road networks for the full-time interval.

To reach more effective forecasting for traffic flow in long-term and complex spatial conditions, spatial-temporal graph modeling has received increasing attention with the advance of graph neural networks [4]. Traffic data are recorded by sensors in the city at fixed position. Those sensors construct a graph where the edge weights are determined by distance (Euclidean, road network distance, etc.) between two nodes. Apparently, traffic data at different time of the same node is not only affected by its historical data but also by neighborhood nodes. Therefore, the key to solve such problem is to effectively extract the spatial-temporal dependence of data. Meanwhile, spatial-temporal graph modeling is a powerful tool to deal with complex spatial-temporal dependencies.

Recent studies on spatial-temporal graph modeling mainly follow two aspects. They either introduce graph convolution

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/ACCESS.2020.3011186, IEEE Access

networks (GCN) into recurrent neural networks (RNN) [5], [6] or use convolution neural networks (CNN) directly [3], [4]. Despite showing the effectiveness by introducing the graph structure of data into a model, these approaches still face some shortcomings.

Specifically, most of these studies are structure-based [3], [4], [7]–[9], which have good performance on transductive learning tasks. Moreover, current studies for spatial-temporal graph modeling are effective to learn short-term or mid-term temporal dependencies, however, long-term temporal forecasting is still a problem. For instance, RNN-based approaches for sequence learning require iterative training, which introduce error accumulation by steps and spends extra training time. Additionally, RNN-based approaches suffer from gradient exploding or vanishing while capturing the long-term temporal dependency [5], [6], [8]. CNN-based approaches take the advantages of parallel computing, stable gradients, and low memory requirement. However, these methods need to use many layers to capture long-term temporal dependency, which will inevitably loss of some critical information. Moreover, GMAN [10] proposes an attention-based encoder-decoder framework, calculate spatial attention score from all vertices, which the time and memory consumption is heavy.

Inspired by the studies above, in this paper, we represent an end-to-end deep learning based dual path framework, i.e., Spatial-Temporal Graph Attention Network (STGAT), The main contributions of this paper are as follows:

- The proposed model is designed as a dual path networks with gating mechanisms and residual architecture, so that it can take both potential and existing spatial dependency into account. We design a Spatial-Temporal Block, which contains gated temporal convolution and graph attention layer to handle spatial and temporal dependency at the same time.
- The proposed model can be directly applicable to inductive learning tasks by introducing a graph attention mechanism into spatial-temporal framework, which means our model can be generalized to completely unseen graphs. When the training data in a certain area is lacking, we can directly use the model trained on other related datasets to make prediction.
- The proposed model outperforms all our competitors on two public real-world traffic datasets, to the best our knowledge. The source codes of STGAT are publicly available from github.

II. RELATED WORK

Graph data contains rich relation information among elements. Graph neural networks (GNNs) are connectionist models that capture the dependence on graphs via message passing between the nodes of graphs. In recent years, GNNs have a wide variety of applications. In this section, we first present the overview of traffic flow forecasting, then we present practical applications of Graph Spatial-temporal Networks which combine time series model and GNN model.

A. TRAFFIC FLOW FORECASTING

The literature often refers to traffic as a flow, because it has similar properties to fluids [1]. Traffic flow forecasting has been long regarded as a key functional component in ITSs. As early as 1970s, the autoregressive integrated moving average (ARIMA) model was used to predict short-term freeway traffic flow [11]. Most early traffic (i.e., [12]–[14]) forecasting approaches use shallow traffic models and are still somewhat unsatisfying. Compared with traditional approachs, recent advances in deep learning show more superior performance, such as [15]–[18].

B. GRAPH NEURAL NETWORKS

There have been several attempts in the literature to extend neural networks to deal with arbitrarily structured graphs. Graph Neural Networks is first introduced in [19] and [20], which can directly deal with a more general class of graphs, e.g., cyclic, directed and undirected graphs [21].

Nevertheless, there is an increasing interest in generalizing convolutions to the graph domain. Some researchers categorize graph neural networks as graph convolution networks, graph attention networks and graph autoencoders. **Graph Convolution Networks** play a core role in capturing structural dependencies. Spectral CNN [22] proposes the first spectral convolution neural network. Chebyshev Spectral CNN [23] is proposed to approximate the filters by means of a Chebyshev expansion of the graph Laplacian, removing the need to compute the eigenvectors of the Laplacian and yielding spatially localized filters. ChebNet [24] simplifies the previous method by restricting the filters to operate in a 1-step neighborhood around each node. **Graph Attention Networks** (GAT) [25] also plays an important role in capturing structural dependencies. The idea of GAT is to compute the hidden representations of each node in the graph by attending over its neighbors following a self-attention strategy. **Graph Autoencoder** is an unsupervised learning framework on graph-structured data, which is a kind of popular approaches to learning the graph embedding for plain graphs without attributed information [26], [27]. These approaches focus on spatial dependencies and ignore information on temporal dependencies, thus they are unstable for spatial-temporal graph modeling.

C. SPATIAL-TEMPORAL GRAPH NETWORKS

Graph Spatial-Temporal Networks aims at learning unseen patterns from spatial-temporal graphs, which is increasingly important in many applications such as traffic forecasting and human activity prediction. Graph Spatial-Temporal Networks combines time series model and GNN model. The key idea of graph spatial-temporal networks is to consider spatial dependency and temporal dependency at the same time.

Recently, Graph Spatial-temporal Networks is being applied to various fields by researchers. Surgical Tool Graph Convolutional Networks [28] utilizes both spatial and temporal information from surgical videos for surgical tool presence detection. Spatial-Temporal Graph Convolutional Networks

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/ACCESS.2020.3011186, IEEE Access

[29] takes advantage of the dynamics of human body skeletons convey significant information for human action recognition. In addition, most researchers are interested in traffic forecasting. By developing effective graph spatial-temporal network models, researchers can accurately predict the traffic status of the whole traffic system. Many current methods in traffic forecasting apply GCNs to capture the dependency together with some RNN [5], [6], [8], [30] or CNN [3], [4], [31] to model the temporal dependency.

OGCRNN (optimized graph convolution recurrent neural network) [32] is a RNN based approach for traffic prediction. Additionally, OGCRNN learns an optimized graph in the training phase and reveals the latent relationship among the road segments from the traffic data.

ST-MGCN [31] is a RNN based approach, which represents their data in the form of graphs from three perspectives (spatial geographic relationship, regional functional similarity and regional traffic connectivity), and defines the problem as a multi-modal machine learning problem on multi-graph convolution networks. This requires additional traffic information to help make predictions. ST-MetaNet [8] is RNN based approach, which employs a sequence-to-sequence architecture consisting of an encoder to learn historical information and a decoder to make predictions step by step.

ASTGCN [7] is a CNN based approach, by using three independent components to respectively model three temporal properties of traffic flow (recent, daily-periodic, and weekly-periodic) dependencies, this approach requires more data for training and prediction. Such methods have achieved great performance in traffic prediction. However, in all of the aforementioned approaches with applying GCNs to capture the spatial dependence, the learned filters depend on the Laplacian eigenbasis. Thus, a model trained as a specific structure can not be directly applied to a graph with a different structure.

Graph WaveNet [4] develops a novel adaptive dependency matrix to capture the hidden spatial dependency in the data, however, it still can not be directly applied to a graph with a different structure. GMAN [10] adapts an encoder-decoder architecture, where both the encoder and the decoder consist of multiple spatio-temporal attention blocks to model the impact of the spatio-temporal factors on traffic conditions. As an attention based approach, GMAN simply compute the sum of spatial attention scores from all vertices. STGRAT [9] also proposes an attention-based encoder-decoder architecture, use node attention to capture the spatial correlation among roads and temporal attention to capture the temporal dynamics of the road network by directly attending to features in long sequences.

III. PRELIMINARIES

In this section, we start by offering a definition of Traffic Networks, then give the formal definition of Traffic Flow Forecasting problem explored in this paper.

A. TRAFFIC NETWORKS

In our model, we define a traffic network as a directed graph $G = (V, E)$, where V is the set of nodes with the scale N and E is the set of edges. The adjacency matrix of G is denoted by $\boldsymbol{W} \in \mathbb{R}^{N \times N}$. We extract the whole time period (e.g., one hour) into m continuous time steps accroding to the same sampling frequency. At each time step t, the graph G has a dynamic feature matrix $X \in \mathbb{R}^{N \times F}$, which is used interchangeably with graph signals. In other words, each node on the traffic network G detects F measurements at time step t. The Graph-structured traffic data is shown as Figure 1.

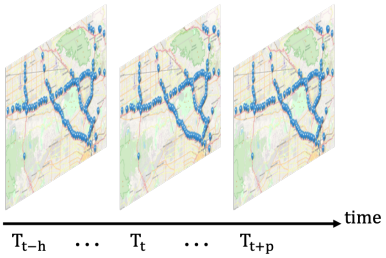


FIGURE 1: The Graph-structured traffic data. Each slices indicates a frame of current traffic status at time step t, which is recorded in a graph-structured data matrix.

B. PROBLEM DEFINITION

Traffic flow forecasting is a typical time-series prediction problem, i.e., predicting the most likely traffic flow in the next p time steps given the previous h steps traffic observations. Suppose we have h historical time step, each time step t has a dynamic feature matrix X_t. The historical data H is denoted as $H = \{X_{t-p+1}, X_{t-p+2}, ..., X_t\}$.

a: Traffic Flow Forecasting

Given the historical data H, the traffic flow Forecasting problem aims to predict $P = \{X_{t+1}, X_{t+2}, ..., X_{t+p}\}$ the next p time step graph signals. In practice, according to the size of p, we follow the most studied dividing principle: where $p \leq 30$ as short-term, where $30 < p \leq 60$ as long-term, where $60 < p$ as long-term.

IV. METHODOLOGY

In this section, we proposed the framework of Spatial-Temporal Graph Attention Networks. We design a dual path architecture by stacked Spatial-Temporal Blocks.

A. THE DUAL PATH ARCHITECTURE

We present the proposed framework of STGAT in Figure 2. We consider that the existing and potential spatial dependency which are both important for learning good representations. To enjoy the benefits from both path topologies, each path in our proposed STGAT shares common structure while maintaining the spatial dependencies to explore new features through dual path architecture. One path

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/ACCESS.2020.3011186, IEEE Access

takes adjacency matrix as input and the other takes the self-adaptive adjacency matrix. In addition, self-adaptive adjacency matrix is initially as a directed complete graph and the final values are obtained by training. Each path consists of stacked spatial-temporal convolutional blocks(ST-Block). The spatial-temporal convolutional blocks are composed of three GTCN with different dilation factors (the factors are 1, 2, 1) and one graph attention layer. By stacking multiple spatial-temporal layers, STGAT is capable of handling spatial and temporal dependencies at the same time. In the end, we design a gated fusion mechanism to combine the nodes features get from each path. The inputs of STGAT is the traffic history data $H = \{X_{t-p+1}, X_{t-p+2}, ..., X_t\}$ as we defined. Instead generating X_p recursively through p steps, STGAT predict $P = \{X_{t+1}, X_{t+2}, ..., X_{t+p}\}$ as a whole.

B. SELF-ADAPTIVE ADJACENCY MATRIX

In order to take potential spatial dependencies into account and catch more useful information from the historical traffic data for better forecasting performance, we propose a self-adaptive adjacency matrix A_{adj}. A_{adj} is a parameterized module consists of $N \times N$ (N is the number of nodes) learnable parameters, where $a_{adj(i,j)}$ determines the weight between node i and node j. We initialize $A_{adj} = 1$, which means we first compute attention scores from all vertices and adjust A_{adj} during the training.

C. SPATIAL-TEMPORAL BLOCK

In order to capture features of both spatial and temporal domains, the spatio-temporal block (ST-Block) is constructed to jointly process graph-structured time series. The block itself can be stacked or extended based on the scale and complexity of particular cases. Each block is composed of three Gated Temporal Convolution Layer (GTCN) with different dilation factors (the factors are 1, 2, 1) and one graph attention layer.

1) Gated Temporal Convolutional Layer

In the temporal dimension, to capture the complex temporal dependencies, we adopt a Gated Temporal Convolutional Layer (GTCN). GTCN is designed as residual architecture to let more information flow through, we stacked three GTCN layer and design a residual connect from input $\mathcal{X}$ to the output of last GTCN layer, as shown in Figure 2. The structure of each GTCN layer is shown in Figure 3. In our GTCN, each layer contains two convolutional layers with the same parameters number. For ensuring a suitable temporal context window to take advantage of the long-term dependence of the traffic signal, we set different dilation factors of different layers, respectively. Given the input $\mathcal{X} \in \mathbb{R}^{N \times F \times K}$, where K is the number of kernels, the temporal gated convolution can be defined as equation 1, 2.

$$gated = \sigma(\mathcal{X} \times M + b) \tag{1}$$

$$\begin{aligned} GTCN(\mathcal{X}) = gated \otimes (\mathcal{X} \times V + c) \\ +(1 - gated) \otimes (\mathcal{X} \times U + d) \end{aligned} \tag{2}$$

where M, V, U, b, c, d are learnable parameters in convolution, $\otimes$ is the element-wise product, and $\boldsymbol{\sigma}$ is the sigmoid function which determines the ratio of information passed to the next layer.

2) Graph Attention Layer

The traffic network is usually described by a graph structure. It is natural to treat the road networks as graphs mathematically. Graph Attention Layer [25] is a novel neural network approach that operates on graph-structured data. Graph Attention Layer is not structure-dependent like Graph Convolutional Network (GCN) and the model with Graph Attention Layer can be directly applicable to inductive learning problems. The implementation of GAT is as same as [25]. The input to the layer is a set of node features, $\mathbf{h} = \{\vec{h}_1, \vec{h}_2, ..., \vec{h}_N\}$, $\vec{h}_i \in \mathbb{R}^F$, where N is the number of nodes in the graph and F is the number of nodes features. Here we formulate the GAT from equation 3 to 5 as follows.

$$e_{ij} = LeakyReLU(\vec{a_i^T}(W\vec{h_i} \parallel W\vec{h_j})) \tag{3}$$

In equation 3, $\parallel$ denotes concate operation, $\vec{a}_i \in \mathbb{R}^{2F'}$ is a learnable weight vector.

$$a_{ij} = softmax_j(e_{ij}) = \frac{exp(e_{ij})}{\sum_{k \in \mathcal{N}_{(i)}} exp(e_{ik})} \tag{4}$$

$$h'_i = \sigma\left(\sum_{j \in \mathcal{N}_{(i)}} a_{ij} W\vec{h_i}\right) \tag{5}$$

In equation 4 and 5, $\mathcal{N}_{(i)}$ denotes a neighbor nodes set of node i, σ a nonlinearity function, $\boldsymbol{\sigma}$, and h'_i is the new features of node i.

To further stabilize the learning process, a multi-head attention is designed. The K independent attention heads have their parameters respectively, and their outputs can be merged into two ways, concatenation (equation 6) and average (equation 7). In this work, we use concatenation for hidden layers and average for the final layer.

$$h'_i = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_{(i)}} a_{ij}^k A^k \vec{h}_j\right) \tag{6}$$

$$h'_i = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j \in \mathcal{N}_{(i)}} a_{ij}^k A^k \vec{h}_j\right) \tag{7}$$

D. FUSION

Instead of simple linear combination, we design a gated fusion mechanism. The input is the output of each path $\vec{t} \in R^{N*P}$ and $\vec{s} \in R^{N*P}$, where N is the number of nodes, P are the features size of output. The formulation of Gated Fusion as follows,

$$gate = \sigma(W_1 \times (\vec{t} + \vec{s})) \tag{8}$$

$$Fusion(\vec{s}, \vec{t}) = tanh\big(W_2 \times \vec{t} \times gate + (I - gate) \times W_3 \times \vec{s}\big) \tag{9}$$

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/ACCESS.2020.3011186, IEEE Access

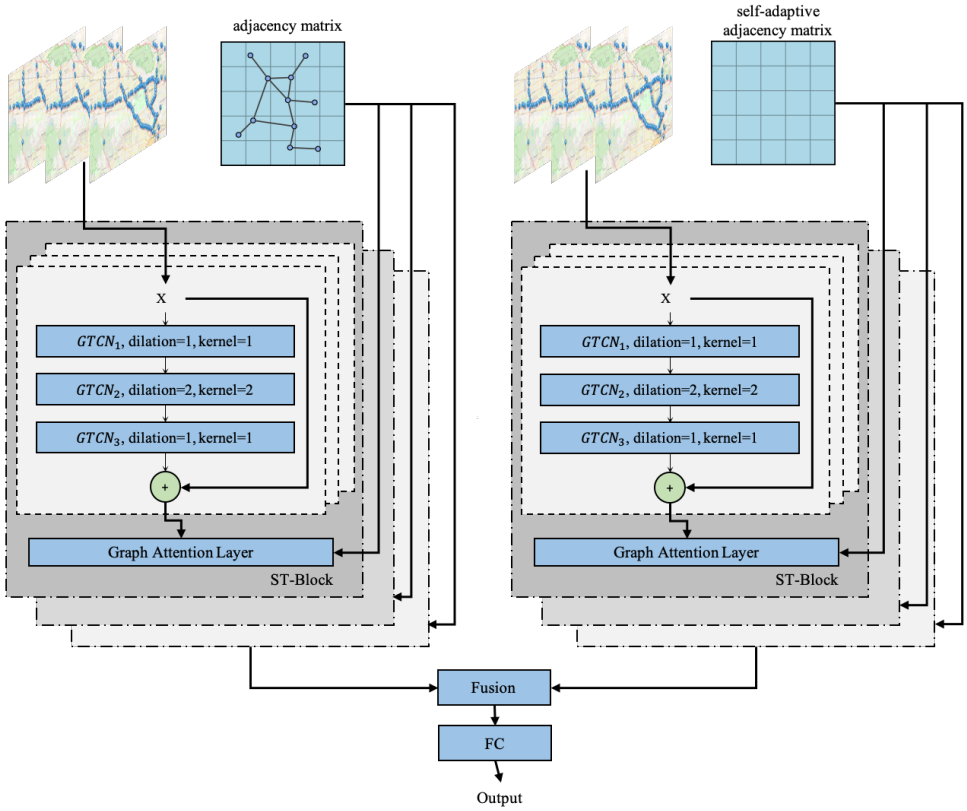


FIGURE 2: The framework of Spatial-Temporal Graph Attention Networks.

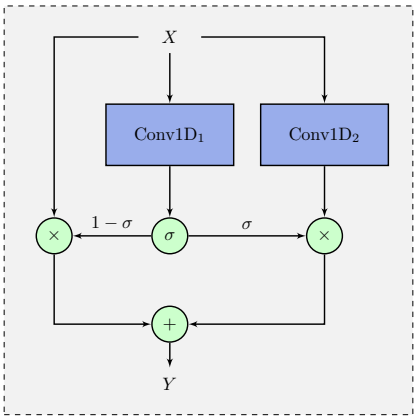


FIGURE 3: The structure of Gated Temporal Convolutional Layer.

Dataset	#Sensor(Nodes)	#Edges	#Time Steps
METR-LA	207	1515	34272
PEMS-BAY	325	2369	52116

TABLE 1: Detailed dataset information of METR-LA and PEMS-BAY.

where matrix I has the same shape with $gate$ and each element in I is 1, W_1, W_2 and W_3 are learnable parameters in network.

V. EXPERIMENTS

In this section, we compare the proposed STGAT with other state-of-the-art baselines on two public traffic (network) datasets for traffic flow forecasting.

A. DATASETS

We design the same procedures as DCRNN [6], verifying STGAT on two public real-world large-scale traffic network datasets: (1) **METR-LA** including traffic speed data with 207 sensors collected from Mar 1st, 2012 to Jun 30th, 2012 for 4 months on the highways of Los Angeles County [33]. (2) **PEMS-BAY** is collected by California Transportation Agencies (CalTrans) Performance Measurement System (PeMS). We select 325 sensors data in the Bay Area and collect 6 months of data ranging from Jan 1st, 2017 to May 31th, 2017 as DCRNN [6] for the experiment. Detailed dataset information are provided in Table 1. For both datasets, the readings of the sensors are aggregated into 5-minutes windows and apply Z-Score normalization.

B. PREPROCESSING

To have a fair comparison, we design exactly the same preprosessing procedures as discribed in [6]. The adjacency matrix of the sensor is constructed by road network distance with a threshold Gaussian kernel [34]. As shown in equation 10, W_{ij} represents the weight between sensor v_i and sensor v_j, $\mathrm{dist}(v_i, v_j)$ denotes the road network distance (not spatial

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/ACCESS.2020.3011186, IEEE Access

distance) from sensor v_i to sensor v_j, the Floyd algorithm is applied to find the shortest distance for each sensor (As data is insurficient, we only adopt Floyd in Los Angeles). σ is the standard deviation of distances and k is predefined threshold.

$$W_{ij} = \begin{cases} \exp\left(- \frac{\mathrm{dist}(v_i, v_j)^2}{\sigma^2} \right) & , \mathrm{dist}(v_i, v_j) > k \\ 0 & , \mathrm{otherwise} \end{cases} \quad (10)$$

The datasets are splited into three parts, where 70% is for training, 20% is for testing while the remaining 10% for validation.

C. BASELINES

In order to evaluate the performance of our proposed STGAT, we conduct the comparison experiments with the state-of-the-art baselines listed as follows: **ARIMA.** Auto-Regressive Integrated Moving Average model with Kalman filter [6]. **FC-LSTM.** Recurrent neural network with fully connected LSTM hidden units [6]. **DCRNN.** Diffusion convolution recurrent neural network [6], which captures the spatial dependency using bidirectional random walks on the graph, and the temporal dependency using the encoder-decoder architecture with scheduled sampling. **GGRU.** Graph gated recurrent unit network [35]. A convolutional sub-network is used to control each attention head's importance. Design the Graph Gated Recurrent Unit (GGRU) to address the traffic speed forecasting problem. **STGCN.** Spatial-temporal graph convolution network [3], which formulate the problem on graphs and build the model with complete convolutional structures, instead of applying regular convolutional and recurrent units, to get much faster training speed with fewer parameters. **Graph WaveNet.** A framework for Deep Spatial-Temporal Graph Modeling [4]. Use a learnable adaptive dependency matrix to capture the hidden spatial dependency in the data. With a stacked dilated 1D convolution component with different receptive fields to handle very long sequences. **ST-MetaNet** [8] employs a sequence-to-sequence architecture, consisting of an encoder to learn historical information and a decoder to make predictions step by step.**STGRAT** [9] propose an attention-based encoder-decoder architecture, use node attention captures the spatial correlation among roads and temporal attention captures the temporal dynamics of the road network by directly attending to features in long sequences. **GMAN.** [10] adapts an encoder-decoder architecture (similar with transformer), where both the encoder and the decoder consist of multiple spatio-temporal attention blocks to model the impact of the spatio-temporal factors on traffic conditions.

Since ST-MGCN [31] requires additional traffic information to construct graphs and ASTGCN [7] needs longer time spans data for training and forecasting, we not include these two methods ST-MGCN and ASTGCN in our comparison experiments.

Data	Model	15 min			30 min			60 min		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
PEMS-BAY	ARIMA	1.62	3.30	3.50%	2.33	4.76	5.40%	3.38	6.50	8.30%
	FC-LSTM	2.05	4.19	4.80%	2.20	4.55	5.20%	2.37	4.96	5.70%
	WaveNet	1.39	3.01	2.91%	1.83	4.21	4.16%	2.35	5.43	5.87%
	STGCN	1.36	2.96	2.90%	1.81	4.27	4.17%	2.49	5.69	5.79%
	DCRNN	1.38	2.95	2.90%	1.74	3.97	3.90%	2.07	4.74	4.90%
	GGRU	-	-	-	-	-	-	-	-	-
	Graph WaveNet	1.30	2.74	2.73%	1.63	3.70	3.67%	1.95	4.52	4.63%
	GMAN	1.34	2.82	2.81%	1.62	3.72	3.62%	**1.86**	**4.32**	**4.31%**
	STGRAT	**1.29**	**2.71**	**2.67%**	1.61	3.69	3.63%	1.95	4.54	4.64%
	STGAT-TCN	1.33	2.82	2.79%	1.65	3.75	3.71%	1.95	4.54	4.73%
	STGAT-Adj	1.34	2.90	2.82%	1.66	3.93	3.85%	1.93	4.52	4.61%
	STGAT-Adp	1.33	2.87	2.76%	1.65	3.86	3.77%	1.92	4.51	4.58%
	STGAT	1.32	2.76	2.80%	**1.61**	**3.68**	**3.66%**	**1.91**	**4.43**	**4.57%**
METR-LA	ARIMA	3.99	8.21	9.60%	5.15	10.45	12.70%	6.90	13.23	17.40%
	FC-LSTM	3.44	6.30	9.60%	3.77	7.23	10.90%	4.37	8.69	13.20%
	WaveNet	2.99	5.89	8.04%	3.59	7.28	10.25%	4.45	8.93	13.62%
	STGCN	2.88	5.74	7.62%	3.47	7.24	9.57%	4.59	9.40	12.70%
	DCRNN	2.77	5.38	7.30%	3.15	6.45	8.80%	3.60	7.60	10.50%
	GGRU	2.71	5.24	6.99%	3.12	6.36	8.56%	3.64	7.65	10.62%
	ST-MetaNet	2.68	5.15	6.96%	3.09	6.28	8.43%	3.60	7.52	10.54%
	Graph WaveNet	2.69	5.15	6.90%	3.07	6.22	8.37%	3.53	7.37	10.01%
	GMAN	-	-	-	-	-	-	-	-	-
	STGRAT	**2.60**	**5.07**	6.61%	3.01	6.21	8.15%	3.49	7.42	10.01%
	STGAT-TCN	2.73	5.28	7.31%	3.09	6.37	8.68%	3.51	7.37	10.38%
	STGAT-Adj	2.72	5.26	7.15%	3.09	6.36	8.53%	3.49	7.30	10.08%
	STGAT-Adp	2.73	5.27	7.32%	3.08	6.37	8.70%	3.48	7.32	10.28%
	STGAT	2.66	5.12	6.89%	**3.01**	**6.12**	**8.14%**	**3.46**	**7.19**	**9.79%**

TABLE 2: Performance comparison of STGAT and baseline models.

D. EXPERIMENTAL DETAILS

The experiments are compiled and tested on Linux system, with the environment of Intel(R) Core(TM) i9-9900K CPU@3.60GHz and NVIDIA GeForce GTX 2080 Ti. We adopt a grid search strategy to find the best hyper-parameters on validations. All the tests take 60 minutes (with 12 observed data points) with the historical time window. The historical time window is used to forecast traffic conditions in the following 15, 30, and 60 minutes (with 3, 6, 12 observed data points respectively), where 15, 30, 60 minutes correspond to short-term, medium-term and long-term respectively.

Each path of daul path architecture has 4 ST-Blocks and one output layer to construct STGAT. Equtaion 6 is used as the first three ST-Block's graph attention layers. Both of the first three layers consist of $K = 4$ attention heads computing $F' = 64$ features. Equation 7 is used as the last floor graph attention layers. The last layer consists of $K = 6$ attention heads computing $F' = 64$ features. Each block is stacked by three Gated Temporal Convolution Layer (GTCN) with dilation factors $1, 2, 1$. We apply dropout, batch normalization and residual connection inside each block.

STGAT is trained using Adam optimizer with an initial learning rate of 0.0003, dropout with $p = 0.6$ is applied to the outputs of the graph attention layer. All models are initialized using Glorot initialization [36]. The evaluation metrics follows Graph WaveNet's work [4], including mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE). Missing values are excluded in both training and testing.

E. RESULTS

The overall performance of all competing methods across datasets is illustrated in Table 2. Our framework achieves superior performance in all three evaluation metrics (MAE, RMSE, MAPE) on both dataset. The results demonstrate our proposed framework STGAT can be applied to a variety of prediction scenarios.

Our STGAT performs much better than temporal meth-

ods (including ARIMA, FC-LSTM, and WaveNet) and spatial-temporal methods (including convolution-based approach STGCN and recurrent-based approaches DCRNN and GGRU). The explanation is that STGAT considers not only temporal but also spatial dependencies, especially in the spatial dimension, while considering both potential and existing dependencies.

A phenomenon worth considering is that STGAT achieves small improvement over the latest method Graph WaveNet and ST-MetaNet in 15 minutes ahead of prediction and outperforms by a large margin in 30 minutes and 60 minutes ahead of prediction. Graph WaveNet also uses gating mechanisms in temporal dimension, but without residual architecture. STGRAT uses temporal attention attends to important time steps of each node, but without gating mechanisms may contain redundant information for prediction. Moreover, to further understand the effect of GTCN, we design a variant STGAT-TCN of our STGAT, which only uses TCN as describe in [37] without gated mechanism and residual architecture. For experimental result is shown in Table 2, STGAT-TCN has worse performance in long-term forecast than other variants. The possible explanation is that in our STGAT the residual architecture allows more information pass to the next layer, while the dilation factor is used to ensure a suitable temporal context window to take advantage of the long-term dependence of the traffic data, which results a good performance in long-term prediction.

Another interesting phenomenon is that all the methods perform better in dataset PEMS-BAY than METR-LA. Moreover, each method has similar performance in dataset PEMS-BAY even if for the traditional method ARIMA. The reason is that Los Angeles is known for its complicated traffic conditions, so traffic forecasting in METR-LA dataset is more challenging than that in the PEMS-BAY (Bay Area) dataset. What's more, our STGAT outperforms all the baseline methods with a significant reduction of prediction error on dataset METR-LA under three evaluation metrics, which means that compared with baseline methods, our STGAT can better handle more complex situations.

1) Ablation Study

To understand the impact of different paths in our dual path network architecture, we split our dual path network STGAT into two subnetworks STGAT-Adj and STGAT-Adp, where STGAT-Adj with adjacency matrix as input and STGAT-Adp with self-adaptive adjacency matrix as input. As shown in Table 2, comparing to our full version STGAT, the two simplified versions of STGAT all lead to certain degradation of performance. This empirically proves the effectiveness of the two important components of STGAT.

Both subnetworks can use stacked ST-Block to capture the spatial-temporal dependencies effectively. The subnetwork with adjacency matrix is more capable of detecting spatial dependencies at each temporal stage. Since traffic flows can be affected by many complex external factors, the subnetwork with self-adaptive adjacency can bring extra useful information and capture the hidden spatial dependence in the traffic data. We believe that the traffic data contains many implicit spatial dependencies that are not represented in the adjacency matrix, future traffic flow is not only influenced by neighboring areas but also by other further nodes. Thus, through the joint efforts of both component, STGAT can better explore that implicit information in the data. It demonstrates that adopting a component with learnable self-adaptive adjacency matrix to capture the hidden spatial dependence in the data can improve the performance of traffic flow forecasting.

2) Generalization Performance of STGAT

In order to verify the generalization ability of the framework, we trained STGAT-BAY with dataset PEMS-BAY while test it on the other dataset METR-LA (not similar with work [38], we do not use transfer learning.). Since the two road networks may have different hidden dependencies, the self-adaptive adjacency matrix learned on dataset PEMS-BAY can not be directly applied. To tackle this problem, we use the adjacency matrix of METR-LA to replace the self-adaptive adjacency matrix. For both path in our STGAT, we use adjacency matrix as input in generalization experiments. The hyper-parameter settings and experimental setup is identical as described previously. Table 3 demonstrates the performance of STGAT-BAY and other GCN-based model for 15 minutes, 30 minutes and 60 minutes ahead prediction on the datasets PEMS-BAY. Since the learned filters depend on the Laplacian eigenbasis which requires the graph structure, all GCN-based models trained on a specific structure can not be directly applied to another graph. This suggests that except for our STGAT, all GCN-based models cannot be trained with PEMS-BAY and test on METR-LA. STGAT trained on a specific structure can be directly applied to another graph without additional training step. When the training data in a certain area is lacking, we can directly use the model trained in other places to make predictions. These experimental results demonstrate that our framework is fairly flexible to be directly applied to another graph without any prior knowledge.

Data	Time	MAE	RMSE	MAPE
METR-LA	15 min	3.96	8.32	9.98%
	30 min	5.14	10.40	12.51%
	60 min	6.68	12.39	18.71%

TABLE 3: Generalization performance of the STGAT on different graph structure. We training STGAT with dataset PEMS-BAY, and testing it on dataset METR-LA.

However, the generalization performance of our model are not ideal. We also design another experiment to improve the generalization performance of our STGAT which uses both datasets for training. We set self-adaptive adjacency matrix as $N \times N$, where $N = 325$ for dataset PEMS-BAY, and use the first 207×207 elements for dataset METR-LA. The hyper-parameter settings and experimental setup is also identical described previously. The results are shown in

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/ACCESS.2020.3011186, IEEE Access

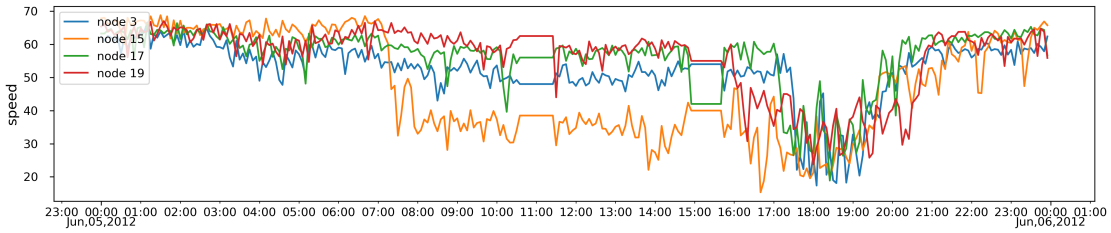
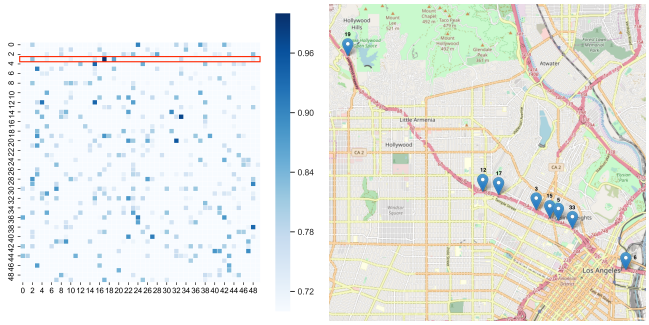


FIGURE 4: The speed curves of nodes on dataset METR-LA.



(a) The heatmap of self-adaptive adjacency matrix in subnetwork STGAT-Adp for the first 50 nodes.

(b) Location of a part of nodes marked on OpenStreetMap.

FIGURE 5: The subnetwork STGAT-Adp with self-adaptive adjacency matrix.

Table 4. Our STGAT has better performance over baseline (STGCN, WaveNet, FC-LSTM and ARIMA) on long-term prediction. Moreover, our STGAT has better performance than baselines (WaveNet, FC-LSTM and ARIMA) on both short-term and mid-term prediction. This demonstrates that our model has the potential for further improvement on generalization performance.

Data	Time	MAE	RMSE	MAPE
METR-LA	15 min	3.15	6.03	8.42%
	30 min	3.64	7.08	10.49%
	60 min	4.37	8.48	13.19%

TABLE 4: Generalization performance of the STGAT on different graph structure. We training STGAT with both dataset PEMS-BAY and METR-LA, testing it on dataset METR-LA.

F. EFFECT OF SELF-ADAPTIVE ADJACENCY MATRIX

In order to better understand our model, we further discuss about the hidden dependencies captured by subnetwork STGAT-Adp as shown in Figure 5. According to Figure 5a, some columns in row 3 have higher value than other rows, such as columns 15, 17 and 19. It suggests that each node has different influence on the other nodes in graph. Figure 5b confirms our observation, that is the strong hidden influence between nodes pretends to be able to match up with a real geographic routine that links each other. To further study the correlation between node 3, 15, 17 and 19, we plot the speed curves of these nodes during one day in Figure 4. We find an interesting phenomenon that the time series sequence of all nodes have the same trend. The possible explanation is that our self-adaptive adjacency matrix can connect the nodes which have the same temporal patterns at close distance range.

VI. DISCUSSION AND CONCLUSION

In this paper, we propose a novel spatial-temporal deep learning framework STGAT for traffic flow prediction. STGAT is a general framework to process spatial-temporal forecasting tasks. To take into account both potential and existing spatial dependencies, we design a dual path networks architecture and each subnetwork share the same structure. Integrating the graph attention mechanism and gated temporal convolution through spatial-temporal blocks. By adopting different dilation factor of different layer in gated temporal convolution, STGAT is able to handle long temporal sequence. Through adopting graph attention mechanism into our spatial-temporal framework, the model can be directly applied to inductive learning problems. Finally, our STGAT achieves state-of-the-art results on two public real-world traffic network datasets. In the future, we plan to investigate the proposed model on other spatial-temporal prediction problems. What's more, we will consider more external factors, such as weather and event, and use all of them to improve the performance of traffic flow predictions.

ACKNOWLEDGMENT

The authors would like to thank the National Natural Science Foundation of China (61876017, 61876018, 61906014, 61702031) for their support in this research.

REFERENCES

[1] A. M. Nagy and V. Simon, "Survey on traffic prediction in smart cities," Pervasive and Mobile Computing, pp. S1 574 119 217 306 521–, 2018.

[2] Y. Wang, D. Zhang, Y. Liu, B. Dai, and L. H. Lee, "Enhancing transportation systems via deep learning: A survey," Transportation Research Part C: Emerging Technologies, vol. 99, pp. 144 – 163, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X18304108

[3] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," pp. 3634–3640, 2018. [Online]. Available: http://dl.acm.org/citation.cfm?id=3304222.3304273

[4] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," CoRR, vol. abs/1906.00121, 2019. [Online]. Available: http://arxiv.org/abs/1906.00121

[5] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in Neural Information Processing, L. Cheng, A. C. S. Leung, and S. Ozawa, Eds. Cham: Springer International Publishing, 2018, pp. 362–373.

[6] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," 2018. [Online]. Available: https://openreview.net/forum?id=SJiHXGWAZ

[7] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 922–929, Jul. 2019. [Online]. Available: https://wvvw.aaai.org/ojs/index.php/AAAI/article/view/3881

[8] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 1720–1730.

[9] C. Park, C. Lee, H. Bahng, K. Kim, S. Jin, S. Ko, J. Choo et al., "Stgrat: A spatio-temporal graph attention network for traffic forecasting," arXiv preprint arXiv:1911.13181, 2019.

[10] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," arXiv preprint arXiv:1911.08415, 2019.

[11] M. S. Ahmed and A. R. Cook, Analysis of freeway traffic time-series data by using Box-Jenkins techniques, 1979, no. 722.

[12] 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.

[13] Y. Kamarianakis and P. Prastacos, "Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches," Transportation Research Record, vol. 1857, no. 1, pp. 74–84, 2003.

[14] D. Nikovski, N. Nishiuma, Y. Goto, and H. Kumazawa, "Univariate short-term prediction of road travel times," in Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005. IEEE, 2005, pp. 1074–1079.

[15] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 2, pp. 865–873, 2014.

[16] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," Transportation Research Part C: Emerging Technologies, vol. 54, pp. 187–197, 2015.

[17] R. W. Liu, J. Chen, Z. Liu, Y. Li, Y. Liu, and J. Liu, "Vessel traffic flow separation-prediction using low-rank and sparse decomposition," in 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2017, pp. 1–6.

[18] Y. Li, R. W. Liu, Z. Liu, and J. Liu, "Similarity grouping-guided neural network modeling for maritime time series prediction," IEEE Access, vol. 7, pp. 72 647–72 659, 2019.

[19] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in IEEE International Joint Conference on Neural Networks, 2005.

[20] S. Franco, G. Marco, T. Ah Chung, H. Markus, and M. Gabriele, "The graph neural network model," IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 61–80, 2009.

[21] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," CoRR, vol. abs/1901.00596, 2019. [Online]. Available: http://arxiv.org/abs/1901.00596

[22] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," 2013.

[23] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in Advances in Neural Information Processing Systems 29, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 3844–3852. [Online]. Available: http://papers.nips.cc/paper/6081-convolutional-neural-networks-on-graphs-with-fast-localized-spectral-filtering.pdf

[24] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," CoRR, vol. abs/1609.02907, 2016. [Online]. Available: http://arxiv.org/abs/1609.02907

[25] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in International Conference on Learning Representations, 2018. [Online]. Available: https://openreview.net/forum?id=rJXMpikCZ

[26] S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations," in Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, ser. AAAI'16. AAAI Press, 2016, pp. 1145–1152. [Online]. Available: http://dl.acm.org/citation.cfm?id=3015812.3015982

[27] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 1225–1234. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939753

[28] S. Wang, Z. Xu, C. Yan, and J. Huang, "Graph convolutional nets for tool presence detection in surgical videos," in Information Processing in Medical Imaging, A. C. S. Chung, J. C. Gee, P. A. Yushkevich, and S. Bao, Eds. Cham: Springer International Publishing, 2019, pp. 467–478.

[29] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," CoRR, vol. abs/1801.07455, 2018. [Online]. Available: http://arxiv.org/abs/1801.07455

[30] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," in Advances in Neural Information Processing Systems 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 5165–5175. [Online]. Available: http://papers.nips.cc/paper/7763-link-prediction-based-on-graph-neural-networks.pdf

[31] X. Geng, X. Wu, L. Zhang, Q. Yang, Y. Liu, and J. Ye, "Multi-modal graph interaction for multi-graph convolution network in urban spatiotemporal forecasting," CoRR, vol. abs/1905.11395, 2019. [Online]. Available: http://arxiv.org/abs/1905.11395

[32] K. Guo, Y. Hu, Z. Qian, H. Liu, K. Zhang, Y. Sun, J. Gao, and B. Yin, "Optimized graph convolution recurrent neural network for traffic prediction," IEEE Transactions on Intelligent Transportation Systems, 2020.

[33] H. V. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, and C. Shahabi, "Big data and its technical challenges," Communications of the Acm, vol. 57, no. 7, pp. 86–94, 2014.

[34] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "Signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular data domains," CoRR, vol. abs/1211.0053, 2012. [Online]. Available: http://arxiv.org/abs/1211.0053

[35] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D. Yeung, "Gaan: Gated attention networks for learning on large and spatiotemporal graphs," CoRR, vol. abs/1803.07294, 2018. [Online]. Available: http://arxiv.org/abs/1803.07294

[36] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256. [Online]. Available: http://proceedings.mlr.press/v9/glorot10a.html

[37] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 156–165.

[38] T. Mallick, P. Balaprakash, E. Rask, and J. Macfarlane, "Transfer learning with graph neural networks for short-term highway traffic forecasting," arXiv preprint arXiv:2004.08038, 2020.

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/ACCESS.2020.3011186, IEEE Access

XIANGYUAN KONG received the B.E. degree in software engineering from Beijing Jiaotong Univer- sity, Beijing, China, in 2017. He is currently pursuing the Ph.D. degree with the School of Software Engineering, Beijing Jiaotong University. His research interests include machine learning and traffic flow forecasting in intelligent transportation systems.

WEIWEI XING received the B.S. degree in computer science and technology and the Ph.D degree in signal and information processing from Beijing Jiaotong University, Beijing, China, in 2001 and 2006, respectively. She is currently a Professor with the School of Software Engineering, Beijing Jiaotong University. Her research interests include intelligent information processing and machine learning.

XIANG WEI received his ph.D degree in Software Engineering from Beijing Jiaotong University in 2019. From 2019 to now, he is currently a lecturer with the School of Software Engineering, Beijing Jiaotong University. He has hold one search grants from National Science Foundation of China (NSFC). His research interest focuses on intelligent information processing, especially for semi-supervised deep learning and GANs.

PENG BAO received the B.E. and M.E. degrees in software engineering from Beijing Jiaotong University, Beijing, China, in 2004 and 2008 respectively, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, in 2014. In 2015, he joined the faculty of the School of Software Engineering, Beijing Jiaotong University, where he is currently an Associate Professor. From 2016 to 2017, he was a visiting scholar in the Department of Computer Science in the University of Illinois Urbana-Champaign. His main research interests include data mining, machine learning, science of science, and intelligent transportation systems.

JIAN ZHANG received the B.S degree in Information Science from China University of Mining and Technology, Xuzhou, China in 2012, the M.E degree in software engineering from Beijing Jiao Tong University, Beijing, China in 2015, and the Ph.D. degree from the Centre de Recherceh en Informatique, Signal et Automatique de Lille, Ecole Centrale de Lille, France in 2018. In 2019, he jointed the faculty of the School of Software Engineering, Beijing Jiao Tong University, where he is currently a lecturer. His main research interests include machine learning, deep learning and intelligent transportation systems.

WEI LU received the B.S. degree in computer science from Fushun Petroleum Institute, Fushun, China, in 1985 and the M.S. degree in computer science and the Ph.D. degree in information and communication engineering from Sichuan University, Chengdu, China, in 1998 and 2006, respectively. He is currently a Professor with the School of Software Engineering, Beijing Jiaotong University, Beijing, China. His current research interests include computer networks and information systems, and multimedia information processing.