IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

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

Digital Object Identifier xx.xxxx/ACCESS.2019.DOI

Spatial-Temporal Graph Attention Networks: A Deep Learning Approach for Traffic Forecasting

CHENHAN ZHANG[1], JAMES J.Q. YU[1], (Member, IEEE), AND YI LIU[1], (Student Member, IEEE)
[1]Department of Computer Science and Engineering, Southern University of Science and Technology

Corresponding author: James J.Q. Yu (e-mail: yujq3@sustech.edu.cn).

This work is supported by General Program of Guangdong Basic and Applied Basic Research Foundation No. 2019A1515011032.

ABSTRACT Traffic speed prediction, as one of the most important topics in Intelligent Transport Systems (ITS), has been investigated thoroughly in the literature. Nonetheless, traditional methods show their limitation in coping with complexity and high nonlinearity of traffic data as well as learning spatial-temporal dependencies. Particularly, they often neglect the dynamics happening to traffic network. Attention-based models witnessed extensive developments in recent years and have shown its efficacy in a host of fields, which inspires us to leverage graph-attention-based method to handling traffic network speed prediction. In this paper, we propose a novel deep learning framework, Spatial-Temporal Graph Attention Networks (ST-GAT). A graph attention mechanism is adopted to extract the spatial dependencies among road segments. Additionally, we introduce a LSTM network to extract temporal domain features. Compared with previous related research, the proposed approach is able to capture dynamic spatial dependencies of traffic networks. A series of comprehensive case studies on a real-world dataset demonstrate that ST-GAT supersedes existing state-of-the-art results of traffic speed prediction. Furthermore, outstanding robustness against noise and on reduced graphs of the proposed model has been demonstrated through the tests.

INDEX TERMS Traffic speed prediction, Graph attention, Deep learning, Intelligent Transportation System, Spatio-temporal domain feature

I. INTRODUCTION

TRAFFIC, as a canonical topic with regards to livelihood, never fail to arouse people's attention. According to a survey in 2017, the driving population of America had exceeded 200 million [1]. Under this background, accurate real-time prediction of traffic conditions is very helpful for governments and related institutes to develop the Intelligent Transportation System (ITS) which can grossly improve the people's travel experience. Traffic Speed Prediction (TSP), as a branch of traffic state prediction in the domain of ITS, has been verified to be useful for many traffic applications such as route guidance, flow control and navigation [2], [3].

TSP has been investigated for decades and the related methods can be roughly divided into two categories, i.e., model-driven approaches and data-driven approaches. Model-driven approaches handle TSP using computational simulation combining with various mathematical theories such as queuing theory [4]. However, the complex simulation process and impractical assumption usually develop massive computational consumption and degenerative results on practical scenarios [5]. In the meantime, the massive acquisition of traffic data and advanced data processing technologies make data-driven approaches an outstanding paradigm in handling the TSP problem [6]. Existing data-driven approaches can be classified into two main categories: parametric and non-parametric models. While classical statistic-based parametric approaches, e.g., Autoregressive Integrated Moving Average (ARIMA) and Kalman Filter (KF) have been widely adopted in the literature [7]–[10], these approaches fail to handle non-linear traffic data since they follow a stationary assumption of time-series [11]. To address this problem, non-parametric machine learning methods such as K-Nearest Neighbors algorithm (KNN) [12] and Support Vector Regression (SVR) [13] are used to model the complex data characteristics considering both high-dimensionality and non-linearity properties. Non-parametric methods are capable of automatically learning their model parameters by capturing the latent information from time-series data and

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.2019.2953888, IEEE Access

remarkable results have been shown in various tasks.

In recent years, the development of deep learning has enabled an increasing number of researchers to adopt deep neural networks for high-accuracy traffic prediction [6], [14]. Huang et al. [15] employ a Deep Belief Network (DBN) to learn effective features for traffic flow prediction in an unsupervised manner. Jia et al. [16] proposed a DBN and Multi-layer Perceptron (MLP) hybrid model for speed prediction. Lv et al. [17] apply Stacked Autoencoder (SAE) to extract traffic features for traffic flow prediction. All of the aforementioned deep learning approaches achieved good results. However, they mainly aim at modelling a single sequence, which is constrained to consider only the time-series dependencies on traffic networks.

To extract the spatial feature of traffic data, researchers introduce Convolutional Neural Networks (CNN) into the traffic prediction tasks. Ma et al. proposed an image-based method that treats the traffic networks as images and use CNN to learn the spatial features [18]. Yu et al. [19] have shown a good result by combining CNN with Long Short-term Memory (LSTM) network for TSP. Wang et al. introduced an error feedback mechanism in their CNN models to meet predictive challenges rising from sudden traffic events [20]. Traditional CNNs are restricted to only process grid-like spatial structures such as images. However, data are often sampled in non-Euclidean spaces such as graphs. To address this issue, Geometric Deep Learning (GDL) is proposed by [21]. Graph Convolutional Networks (GCN) is one of its developments that generalize CNN to graph domains [22], [23]. For traffic data-related problems, GCN is widely adopted to handle various tasks by treating traffic networks as graphs that can fully take advantage of spatial information in traffic [11], [19], [24], [25]. Li et al. [11] proposed a hybrid GCN-based model that captures the spatial dependency with random walks on the traffic network and the temporal dependency with LSTM. Yu et al. [24] proposed a Spatio-Temporal Graph Convolutional Networks (STGCN) that employ convolutional structures on both spatial and time axis. Currently, GCN-based approaches are among the most advanced techniques in traffic prediction research [26].

Although these models achieved outstanding prediction accuracy, most of them tend to extract static spatial dependencies in traffic, while these dependencies may evolve over time [27]. Furthermore, the "black box" nature of current deep learning models renders them bad interpretability [6]. A better comprehending of spatial dependencies of traffic networks extracted from the models would be useful for traffic allocation. Besides, previous research devotes little attention to the noise-tolerance capability of the deep learning traffic speed predictors, although measurement noise and missing data usually happen to the process of data collection.

Therefore, in this paper, we propose a novel deep learning framework, i.e., Spatial-Temporal Graph Attention Networks (ST-GAT). ST-GAT is a hybrid model integrating a spatial dependency extraction block and a temporal feature extraction block. The spatial dependency extraction block comprises a graph attention-based network based on Graph Attention Network (GAT) [28] to extract the spatial dependencies in traffic networks. Deep learning models with attention mechanisms have been verified to be effective in various graph-based tasks [28]–[31]. Among them, GAT especially inspires us since it shows an effective approach to compute the pair-wise attentional correlations, which would be useful to exploit spatial dependencies. Additionally, we design a method to construct time-series speed observations into feature representations, called Speed2Vec, to adapt time-series traffic data to GAT. In the temporal feature extraction block, we employ a LSTM network to learn time-series feature. Our model is validated via a real-world dataset, PeMSD7, collected by California Department of Transportation. Compared with state-of-the-art baselines, our model has superior performance on multiple preset prediction lengths. Additional tests demonstrate the robustness of the proposed model against noise and reduced graphs. Moreover, the analyses of the result shed light on our model's capability in understanding the spatial-temporal traffic dependencies.

The main contribution of this paper is as follows:

- We propose a novel deep learning hybrid model, the spatial-temporal graph attention networks (ST-GAT). To the best of our knowledge, it is the first time to apply GAT to extract spatial-temporal features in a traffic speed prediction study.
- We propose Speed2Vec, an approach for feature representation to convert time-series traffic data into the feature vector for attention computation.
- We conduct a comprehensive performance comparison in a traffic speed prediction task using a real-word dataset. The proposed model distinctly outperforms existing state-of-the-art methods.
- Extensive case studies analyze the performance of the proposed model, its sensitivity to parameters, generalization to simplified graphs, robustness to measurement noise, and interpretability.

The remainder of this paper is organized as follows. Section II formulates the problem of traffic speed forecasting on the graph, and introduce the mathematical formulation of GAT and multi-head attention. Section III describes the structure and main characteristics of the proposed ST-GAT model. Section IV compares the prediction performance of the proposed model with other benchmark models based on the real-world dataset PeMSD7 and presents several sensitivity analyses. Finally, Section V concludes the paper and discusses future studies.

II. PRELIMINARY

In this section, we first formalize the traffic speed prediction problem on road graphs. Then, we elaborate on the principle of GAT as well as multi-head attention mechanism, and they are closely related to the spatial dependency learning of our model.

A. TRAFFIC SPEED FORECASTING PROBLEM ON ROAD GRAPHS

Traffic speed forecasting is a time-series prediction task predicting the future traffic speed, given historical traffic speed observations from sensors at different road segments. Typically, we can formulate this process by

$$v_{t-M+1}, ..., v_t \xrightarrow{f(\cdot)} \hat{v}_{t+1}, ..., \hat{v}_{t+H}, \tag{1}$$

where $v_t \in \mathbb{R}^n$ is an observation vector of n road segments (observation stations) at time step t. The traffic speed forcasting model aims to learn a function $f(\cdot)$ to predict the traffic speed in the following H time steps given the information from past M time steps.

In this work, we represent the traffic network as an undirected graph with traffic time-series, $G_t = (V_t, \mathscr{E}, W)$, where V_t is the set of nodes each of which represents the speed observation from an arbitrary sensor at time t, $\mathscr{E}$ is the set of edges and W is the adjacency matrix of the graph. Subsequently, the traffic speed forecasting problem on road graphs can be represented by

$$[V_{t-M+1}, ..., V_t; \mathscr{E}; G] \xrightarrow{f(\cdot)} [\hat{V}_{t+1}, ..., \hat{V}_{t+H}]. \tag{2}$$

B. GRAPH ATTENTION NETWORK

In this paper, we use GAT to learn the attentions among nodes and apply them into updating hidden features. It is assumed that the updated hidden features with attention information are helpful for further time-series prediction. Therefore, we will detail the propagation rule of GAT in this subsection.

GAT extends GCN by incorporating an explicit attention mechanism. Following a self-attention strategy [32], GAT learns the hidden features of each node by iteratively using node feature for similarity computation. The key difference between GAT and GCN is on how to collect and accumulate the feature representations of neighbor nodes. In GCN, a standard convolution includes the standardized sum of the features of adjacent nodes as

$$h_i^{l+1} = \sigma\left(\sum_{j \in N(i)} \frac{1}{c_{ij}} \phi^l h_j^l\right), \tag{3}$$

where $N(i)$ is the set of adjacent nodes which are immediate neighbors of node i, σ is a non-linear activation function, c_{ij} is a standardized constant based on graph structure, l is the current layer, ϕ^l is the weight matrix for node feature transformation, h_i^{l+1} is the updated hidden feature of node i.

GAT replaces the above convolution operation in graph convolution with an attention mechanism. To better illustrate how the node features of layer l are updated to those of layer $l+1$, we first introduce the consitituting component of GAT, i.e, graph attentional layer. The input to a GAT layer is a set of node features, $h^l = \left\{h_1^l, h_2^l, ...h_N^l\right\}, h_i^l \in \mathbb{R}^F$ where N is the number of nodes and F is the number of features from each node. To transform the input features into higher-level features, a shared weight matrix, $\phi \in \mathbb{R}^{F' \times F}$, is used to cast the input to another feature space of F'-dimension. Then, a self-attention mechanism is defined and shared between along edges to calculate the attention coefficient of nodes and their neighbors:

$$e_{ij} = a\left(\phi h_i^l, \phi h_j^l\right), a : \mathbb{R}^F \times \mathbb{R}^{F'} \to \mathbb{R}, \tag{4}$$

where $a(\cdot, \cdot)$ is the attention mechanism, e_{ij} is the computed attention coefficient. Note that to retain topological information of the graph, only the attention coefficients of the node and its first-hop neighbors are computed. A softmax function is used to normalize the attention coefficients into a easily comparable form. Finally, a Leaky Rectified Linear Units (LeakyReLU) activation function [33] is applied the final normalized attention coefficients α_{ij} is obtained as

$$\alpha_{ij} = \text{softmax}\left(\text{LeakyReLU}\left(e_{ij}\right)\right). \tag{5}$$

Consequently, these coefficients are employed to update model features utilizing the GCN convolution rule [22]:

$$h_i^{l+1} = \sigma\left(\sum_{j \in \text{N}(i)} \alpha_{ij} \phi^l h_j^l\right). \tag{6}$$

a: *Multi-head Attention Mechanism*

Multi-head attention mechanism enables the model to learn an attention coefficient through multiple representation subspaces. In order to make the self-attention learning process robust, multi-head attention mechanism strategies are usually adopted [32], [34]. Specifically, take the adopted multi-head attention mechanism in [28] as an example, K independent attention mechanisms perform the above transformation across in K heads (i.e., K independent attention processes) and their resulting features are concatenated together to develop an output feature representation. Subsequently, the final output is obtained by averaging the concatenation of feature representation. This process is formally defined as

$$\begin{cases} h_i^{l+1} = \Big\Vert_{K=1}^{K} \sigma\left(\sum_{j \in \text{N}(i)} \alpha_{ij}^K \phi^K h_j^l\right), \text{Concatenation} \\ h_i^{l+1} = \sigma\left(\frac{1}{K} \sum_{K=1}^{K} \sum_{j \in \text{N}(i)} \alpha_{ij}^K \phi^K h_j^l\right), \text{Averaging} \end{cases}, \tag{7}$$

III. PROPOSED MODEL

In this paper, we propose a hybrid traffic speed predictor: ST-GAT. As presented in Fig. 2, ST-GAT comprises a spatial GAT block for spatial correlation extraction, an RNN block for the temporal feature learning as well as time-series prediction, and an output layer for producing the sequence output. Specifically, in the spatial GAT block, we employ the aforementioned multi-head attention mechanism that enables the model to jointly learn spatial dependencies through multiple independent attention blocks to benefit the learning process. In the RNN block, we employ a 2-layer LSTM network for extracting time-series feature. The final predictions are

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.2019.2953888, IEEE Access

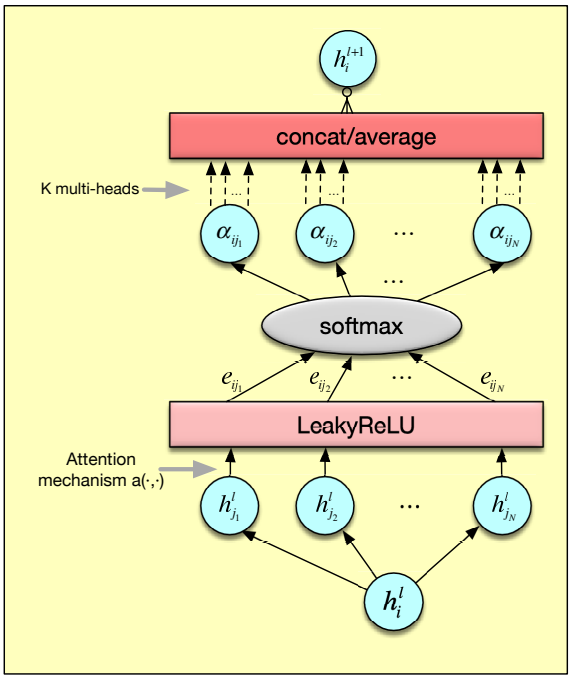


FIGURE 1. A graph attentional layer with multi-head attention mechanism, involving K heads. N denotes the number of nodes connected to node i.

generated by a fully-connected neural network in the final output layer.

A. GAT FOR CAPTURING SPATIAL DEPENDENCIES IN TRAFFIC NETWORKS

GAT leverages the node features to compute attention coefficients that represent the spatial dependency of a graph. There are some challenges when using GAT to handle traffic data. First, it is difficult to find proper feature representations of time-series data, and weak feature representations may lead to unsatisfactory training result. Second, how to represent the learned attention coefficients and apply them to update the hidden feature is crucial to our algorithm. In the following, we devise new approaches to address these problems.

a: *Speed2Vec*

In a traffic prediction task, the observed data are recorded in time-series. To define feasible feature representations on nodes from such data, we propose a Speed-to-Vector (Speed2Vec) data embedding mechanism. Specifically, we consider the speed observations of a node in a fixed historical window as its hidden feature at a time step and embed them in a vector as

$$h_t = \left[v_{t-F+1}, v_{t-F+2}, ..., v_t\right], \tag{8}$$

where $h_t \in \mathbb{R}^F$; t denotes the t-th time frame, and F is the dimension of the vector, whose physical meaning in the context is the historical window size. Then, we reshape the feature representations generated by Speed2Vec as the network-wide input to spatial GAT block, i.e.,

$$H_T^N = \begin{bmatrix} h_1^1 & h_2^1 & \cdots & h_T^1 \\ h_1^2 & h_2^2 & \cdots & h_T^2 \\ \vdots & \vdots & \ddots & \vdots \\ h_1^N & h_2^N & \cdots & h_T^N \end{bmatrix}, \tag{9}$$

where $H_T^N \in \mathbb{R}^{T\times N\times F}$; T is the length of temporal sequence, and N is the number of nodes in the traffic networks. It is worth noting that F should be reasonably large to obtain sufficient temporal features, while an overly large F renders redundant historical data in feature representation as well as an increased computational burden. In this work, we set the F to 12 to ensure that the performance comparison between our model and other baselines are under the same historical window size. Additionally, we will examine the sensitivity of F in Section IV.

Through Speed2Vec, we are able to further compute the attention coefficients using the equations given in section II. Furthermore, this mechanism enables our model to directly utilize the time-series data as the input to GAT. It eliminates the need of constructing our model into a *Sequence to Sequence* structure [11], [35] which incorporates an additional sequence encoder, rendering a more complex predicator design.

b: *Attention Adjacency Matrices*

The final step of the spatial GAT block is updating hidden features. To achieve this, we introduce the attention adjacency matrix which maps the previously learned attention coefficients into an adjacency matrix as

$$\widetilde{A} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1N} \\ \alpha_{21} & \alpha_{22} & & \alpha_{2N} \\ \vdots & & \ddots & \vdots \\ \alpha_{N1} & \alpha_{N2} & \cdots & \alpha_{NN} \end{bmatrix} \tag{10}$$

where α is the attention coefficient, and self-attention is considered. Considering the temporal sequence, we obtain a set of attention adjacency matrices over time and a corresponding 3-D variable can be represented by

$$\widetilde{A}_T = \left[\widetilde{A}_1, \widetilde{A}_2, ..., \widetilde{A}_T\right], \tag{11}$$

where $\widetilde{A}_T \in \mathbb{R}^{T\times N\times N}$; T is the length of temporal sequence, and N is the number of nodes in the traffic networks. In this way, the learned attention coefficients are allowed to multiply network-wide hidden features (i.e., H_T^N) to calculate the updated hidden features.

Furthermore, the attention adjacency matrix embodies the design principle of GAT: better interpretability. In previous work [11], [24], the edge weights are directly computed by the distance between nodes in the networks. Contributed by the adopted attention adjacency matrix, we represent the edge weights by the learned attention coefficients that the

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.2019.2953888, IEEE Access

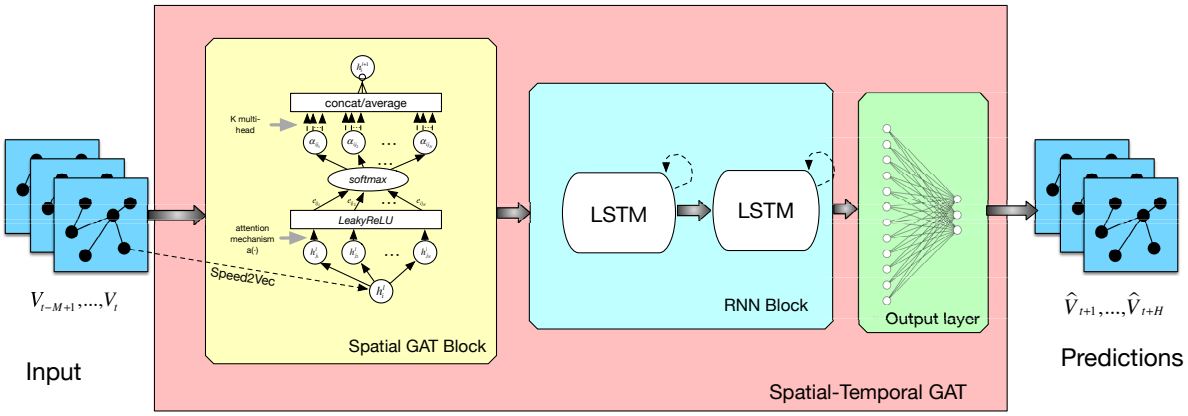


FIGURE 2. Architechture of spatial-temporal graph attention networks (ST-GAT).

spatial dependency can be intuitively presented. We can also observe the dynamic spatial dependency by the evolvement of the attention adjacency matrix. A related case study will be shown in Section IV.

B. RNN FOR TEMPORAL FEATURES LEARNING AND PREDICTION

Traffic data also has a distinct temporal dependency in addition to spatial dependency. Recurrent Neural Networks (RNNs) are usually leveraged to learn temporal dependency and realize time-series prediction [11], [19], [36]. In ST-GAT, we use LSTM, which is one of the most practical variants of RNNs [37], [38]. LSTM introduces a collection of gating units and cell states that control the flow of information to solve the vanishing gradient problem in long-term time-series prediction. Especially, the cell states are the key to LSTMs that they store the memory information and pass through all the time iterations. The gating units have three types, namely, input gate, forget gate, and output gate, which are used to decide whether to add or remove information to a cell state. Given data x_t, the cell output state c_t and the hidden layer output h_t can be computed by[1]

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{(t-1)} + b_{hi}), \tag{12}$$
$$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{(t-1)} + b_{hf}), \tag{13}$$
$$g_t = \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{(t-1)} + b_{hg}), \tag{14}$$
$$o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{(t-1)} + b_{ho}), \tag{15}$$
$$c_t = f_t * c_{(t-1)} + i_t * g_t, \tag{16}$$
$$h_t = o_t * \tanh(c_t), \tag{17}$$

where i_t, f_t, g_t, o_t are the input, forget, cell, and output gates values, respectively, W_{ii}, W_{if}, W_{ig}, W_{io}, W_{hi}, W_{hf}, W_{hg}, W_{ho} are the weight matrices connecting x_t, $h_{(t-1)}$ to three gates and the cell input, b_{ii}, b_{if}, b_{ig}, b_{io}, b_{hi}, b_{hf}, b_{hg}, b_{ho} are the corresponding biases, σ represents the sigmoid function, $\tanh$ represents the the hyperbolic tangent function, and $*$ represents element-wise multiplication here. To construct ST-GAT, we connect the RNN block with spatial GAT block as shown in Fig. 3. Note that for achieving multiple nodes ahead forecasting the LSTM in our model is generalized to a 3-D structure.

[1]With abuse of notation, h_t in this subsection exclusively denotes the hidden layer output of LSTM.

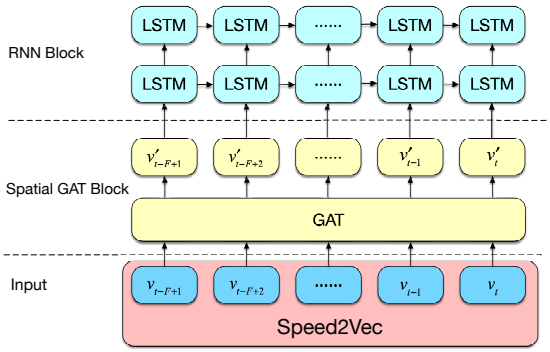


FIGURE 3. The way of connection between RNN block and spatial GAT block. v'_t denotes the updated feature representations at time t.

The last step is to forecast the future H time-step traffic speed. We employ a fully-connected layer which uses the output of LSTM as its input for linear transformation to obtain the final prediction output. Given the input, $\hat{v}_{t+H}$ is computed by

$$\hat{v}_{t+H} = w \times h_t + b, \tag{18}$$

where $w \in \mathbb{R}^{C \times H}$ is a weight matrix that maps the C-channels hidden output of LSTMs to H output and b is the bias.

The proposed model is trained using mean squared error (MSE), also known as L2 loss, which can be written as

$$L\left(\hat{V}_{t+H};\theta\right) = \sum_t \left\| \hat{V}_{t+H} \leftarrow (V_{t-F+1},...V_t;\theta) - V_{t+H} \right\|^2, \tag{19}$$

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.2019.2953888, IEEE Access

where V_{t+H} and $\hat{V}_{t+H}$ denote the network-wide ground truth and predictions, respectively; θ represents all the learnable parameters in the model.

Summarizing the aforementioned, the main characteristics of ST-GAT are threefold. First, ST-GAT can be regarded as a generalized model to handle structured time-series benefited from the Speed2Vec mechanism. It can also be applied to spatial-temporal tasks not limited to traffic speed prediction in road networks. Second, by using attention adjacency matrices, ST-GAT can represent the spatial dependencies by learned attention coefficients among nodes. Third, as a new attempt, the architecture of the proposed ST-GAT model in this paper is more simple compared to existing advanced models, e.g., [11], [27], [39], and it demonstrates a great potential for further enhancements.

IV. EXPERIMENTS

In this section, a series of comprehensive experiments are performed to evaluate the performance of the proposed approach for traffic speed prediction. We first assess its prediction accuracy and compare it with related results of baselines and benchmarks. Additionally, we investigate the sensitivity of model performance to different hyperparameters. Then, we assess its performance on reduced graphs. Furthermore, we demonstrate the interpretability of the proposed approach by visualization. Lastly, we inspect the influence of measurement noise and missing data.

A. SYSTEM CONFIGURATION

a: *Dataset*

PeMSD7 is a dataset collected from Caltrans Performance Measurement System (PeMS) by over 39,000 sensor stations in the District 7 of California. Data samples from each 30-second interval are aggregated into 5-minute periods. We choose the dataset which is sampled by [24] in a medium-scale containing 228 stations of PeMSD7. The time period of the dataset is from May 1st to June 30th of 2012 which only includes the weekdays to avoid atypical traffic.

As the time interval of data collection is set to 5 minutes, each sensor in the road network produces 288 observations per day. When there are missing data points, we use linear interpolation to recover missing data. Additionally, data are normalized by Z-Score method. The training, validation, and testing sets are correspondingly developed, each of which contains 60%, 20%, and 20% of all data.

In this paper, we build the adjacency matrix of sensors (nodes) of road network in two ways. First, an adjacency matrix is generated based on thresholded Gaussian kernel method replacing the computed weight value by 1. Meanwhile, the self-connection is considered. Thus, the adjacency matrix $W = \{w_{ij}\}$ is established by

$$w_{ij} = \begin{cases} 1, \text{ if } i \neq j \text{ and } \exp\left(-\dfrac{\text{dist}(v_i, v_j)}{\sigma^2}\right) \geqslant \varepsilon \\ 1, \text{ if } i = j \text{ (i.e., self-connection)} \\ 0, \text{ otherwise.} \end{cases}, \quad (20)$$

where w_{ij} represents the edge weight between node v_i and node v_j, which is decided by their Euclidean space distance $\text{dist}(v_i, v_j)$; ε and σ^2 are the user-controlled parameters that control the density of graph, and we set their values the same as those in [24] to make a fair comparison. Note that we define the network as an undirected graph. Therefore, the initial adjacency matrix is converted symmetrically, i.e., $w_{ij} = w_{ji}$.

Furthermore, we also develop reduced graphs by only connecting each node with its K nearest neighbors. In this way, we attempt to explore the model performance on the graph with different K values. The related test will be shown later.

b: *Experiment Setting*

All experiments are conducted on a NVIDIA GeForce RTX 2080 GPU and an Intel(R) Xeon(R) E5-2620 v4 CPU. The past time window is 60 minutes (12 observed data points) and they are used to forecast traffic speed in the next 15, 30 and 45 minutes.

ST-GAT is trained based on the optimizer Adam [40] for 150 epochs. The initial learning rate is $2e^{-4}$ with a weight decay of $5e^{-4}$; the batch size is set to 50. The dropout [41] and early stopping are used to prevent overfitting. In addition, we employed batch normalization [42] and Xavier parameter [43] initialization to stable the learning process.

To balance the trade-off between model performance and computational complexity, the adopted architecture setting of ST-GAT is specified as follows by executing grid search strategy. We employ a single graph attentional layer with 8 attention heads to achieve the multi-head attention mechanism. The number of hidden units of the two-layer LSTMs is set to 32 and 128, respectively.

ST-GAT is compared with the following classic and the state-of-the-art machine learning models: (1) HA Historical Average, which models the traffic speed as a seasonal pattern and uses the average of previous seasons as the prediction; (2) Auto-Regressive Integrated Moving Average (ARIMA); (3) Linear Support Victor Regression (LSVR) (4) Feed-Forward Neural Network (FNN); (5) Full-Connected LSTM (FC-LSTM) [35]; (6) Diffusion Convolutional Recurrent Neural Network (DCRNN) [11]; and (7) Spatio-Temporal Graph Convolutional Networks (STGCN) [24].

We use Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) to evaluate all learning models, which are defined as follows

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i}^{n}\left(X_i - \hat{X}_i\right)^2}, \quad (21)$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|X_i - \hat{X}_i\right|, \quad (22)$$

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{X_i - \hat{X}_i}{X_i} \times 100\%\right|, \quad (23)$$

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.2019.2953888, IEEE Access

where X_i denotes the observed traffic speed at time i, and $\hat{X}_i$ denotes the forecast at i. Moreover, we consider MAPE as the most referable one among the three metrics, as did in [19], [44].

B. EXPERIMENT RESULT

a: Comparison with State-of-the-art Methods

Table 1 shows the results of ST-GAT and aforementioned baselines on dataset PeMSD7 for 15 minutes, 30 minutes and 45 minutes ahead predictions where the baseline results are adopted from [24]. We draw the following observations from the comparison. (1) ST-GAT achieves the best prediction accuracy regarding all the metrics for the three forecasting windows with a significance level of 99% (two-sided T-test, $P < 0.01$), even compared with the state-of-the-art graph convolution-based model such as STGCN and GCGRU. Specifically, the MAPE of ST-GAT outperforms STGCN by 0.5% (15 min), 0.76% (30 min), and 0.83% (45 min). This illustrates the effectiveness of the attention mechanism on graph-based spatial feature learning. (2) Traditional statistical and machine learning methods have been greatly outperformed, especially for long-term forecasting. For example, comparing the results of LSVR and FC-LSTM, LSVR achieves better performance in 15 min ahead forecasting. However, in terms of 45 min ahead forecasting, it performs worse than FC-LSTM. This is partly due to their incapability of long-sequence memorization and spatial-temporal learning on complex data.

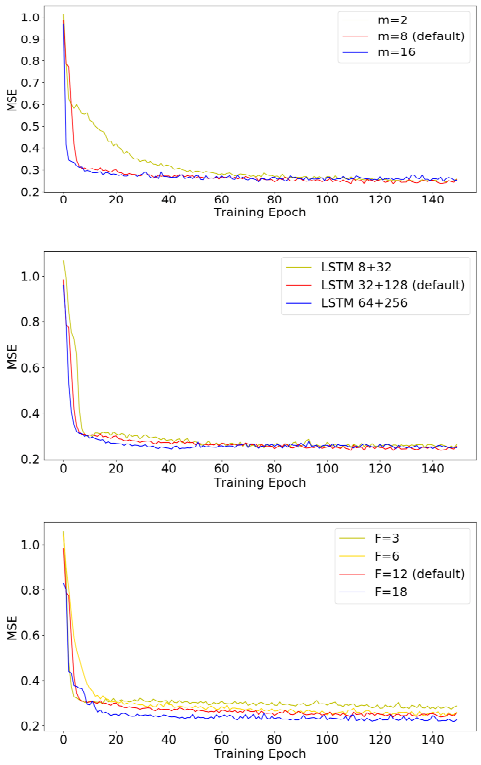


FIGURE 4. Validation MSE versus the number of training epochs.

b: Sensitivity of Hyperparameters

In this work, a hybrid graph-attention-based recurrent neural network is adopted. We empirically select a number of hyperparameters and parameters when constructing this traffic speed predictor. In this subsection, the sensitivity and influence of these settings to the prediction accuracy as well as training speed are investigated.

We test the performance of our model with three architectural hyperparameters, namely, the number of attention heads in the graph attentional layer, the number of hidden units in each LSTM layer and the dimension of Speed2Vec. Specifically, these values are set to 2/8/16, 8+32/32+128/64+256, and 3/6/12/18, respectively, whose default values are presented in Table 2. 45 min ahead prediction is utilized as the benchmark test. We first compare their prediction accuracy. It can be observed that the prediction accuracy is improved with larger hyperparameters. To better illustrate the convergence rate of the training process, we demonstrate the MSE convergence in Fig. 4. From this figure, we observe that convergence speed is also enhanced with larger hyperparameters. Particularly, an obvious increase of training convergence speed can be observed when the number of attention heads is added to 8 from 2, which is contributed by the efficacy of multi-head attention mechanism in ST-GAT.

Furthermore, the larger hyperparameters (e.g., $m = 16$, numbers of neurons in the two LSTM layers: 64 and 256) cannot be simultaneously utilized due to the limitation of the hardware in this experiment. Therefore, we finally present the performance over default hyperparameters as shown in Table 1. Large-scale networks will be investigated in future work.

c: Efficacy of ST-GAT on Reduced Graphs

As mentioned before, in our original work, the traffic network is constructed by building an adjacency matrix based on the distances between road segments. Nonetheless, the influence of distance cannot be reflected since the constructed adjacency matrix is unweighted, i.e., the adjacency weights are all represented by 1 regardless of their actual distance between nodes in the graph. Furthermore, the graph generated by this adjacency matrix is so dense that the average degree of each node reaches 200 when there are 228 nodes in total. The hypothesis is that the graph incorporates unnecessary edges that develop redundant topological information, and the spatial features learned by this cannot help the model improve its prediction performance effectively. Therefore, we are particularly interested in investigating the model performance on reduced graphs which discard this redundant information.

In this subsection, we compare the performance of the proposed model on reduced graphs. Specifically, we obtain reduced graphs by dropping redundant topological connections among nodes in the graphs. For each node in the graph, only the adjacencies with K nearest neighbors are

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.2019.2953888, IEEE Access

TABLE 1. Performance comparison of ST-GAT and baselines on the PeMSD7 dataset.

Model	15 min			30 min			45 min		
	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
HA	7.20	4.01	10.61	7.20	4.01	10.61	7.20	4.01	10.61
ARIMA	9.00	5.55	12.92	9.13	5.86	13.94	9.38	6.27	15.20
LSVR	4.55	2.50	5.81	6.67	3.63	8.88	8.28	4.54	11.50
FNN	4.75	2.74	6.38	6.98	4.02	9.72	8.58	5.04	12.38
FC-LSTM	6.20	3.57	8.60	7.03	3.94	9.55	7.51	4.16	10.10
GCGRU	4.21	2.37	5.54	5.96	3.31	8.06	7.13	4.01	9.99
STGCN	4.04	2.25	5.26	5.70	3.03	7.33	6.77	3.57	8.69
ST-GAT	**3.45**	**2.01**	**4.76**	**4.68**	**2.76**	**6.57**	**5.30**	**3.20**	**7.86**

TABLE 2. ST-GAT hyperparameter test. m denotes the number of multi-heads, LSTM denotes the number of hidden units in each LSTM layer (1st/2nd), and F denotes the dimension of the Speed2Vec.

Hyperparameter	45 min		
	RMSE	MAE	MAPE (%)
m=2	5.55	3.31	8.38
m=8 (default)	5.30	3.20	7.86
m=16	5.22	3.08	7.40
LSTM=8/32	5.38	3.24	7.89
LSTM=32/128 (default)	5.30	3.20	7.86
LSTM=64/256	5.24	3.06	7.38
F=3	5.44	3.28	8.03
F=6	5.37	3.22	7.99
F=12 (default)	5.30	3.20	7.86
F=18	5.20	3.03	7.51

TABLE 3. Accuracy of ST-GAT with measurement noise and missing data

Disturbance		45 min		
		RMSE	MAE	MAPE (%)
Noise	Original	5.30	3.20	7.86
	0.5%	5.31	3.22	7.87
	1%	5.35	3.27	7.94
	2%	5.48	3.35	7.99
Missing	Original	5.30	3.20	7.86
	0.5%	6.83	3.73	8.15
	1%	8.01	3.88	8.49
	2%	9.91	4.91	8.64

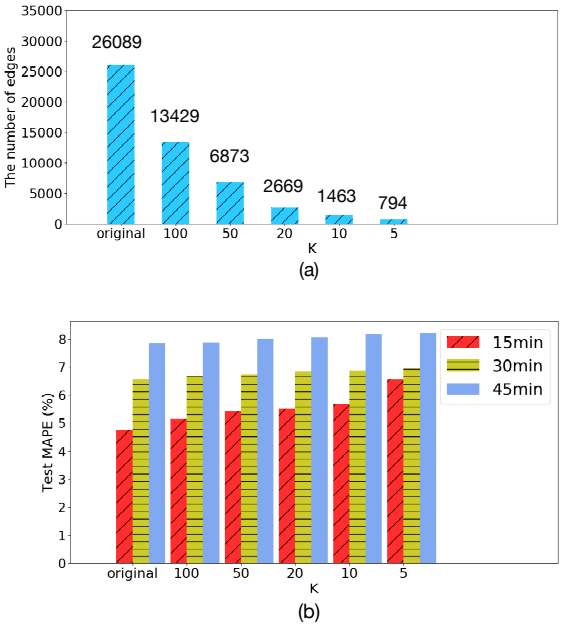


FIGURE 5. (a) The total number of edges in each reduced graph. (b) Model performance on reduced graphs. K corresponds to the number of the nearest neighbors to connect for each node to generate the graph.

retained[2]. A suite of reduced graphs with K equaling to $5/10/20/50/100$, respectively, are constructed. Fig. 5 (a) indicates the total number of edges in each reduced graph and Fig. 5 (b) shows the comparison of model performance on the original and reduced graphs. Overall, we observe that it does cause degenerated performance with the smaller K. However, the performance degradation is minuscule when compared to the degree to the reduction of the graph. For example, when $K = 20$, namely approximately 90% of the edges in the graph are discarded compared with the original graph, the model performance is still quite close to that on the original graph where the prediction accuracy only suffers from a $0.2 - 0.7\%$ (MAPE) penalty. This case study demonstrates the effectiveness of the proposed model on reduced graphs, which develop good prediction results with relatively less graph information.

It is also worth noting that we observe that the reduced graphs seem to have a greater effect on the model performance of short-term prediction than long-term prediction. Comparing the performance of the model on the graph ($K = 5$) and the original graph, the penalties of the performance are $1.7\%/0.4\%/0.3\%$ corresponding to 15 min/30 min/45 min, respectively. This phenomenon will be further investigated in future work.

d: *Model Interpretation*

Traditional methods such as ARIMA, FC-LSTM are not able to exploit spatial dependency. However, by employing the attention mechanism on traffic networks, our model ST-GAT has the compatibility to extract spatial feature from new traffic data. To better understand the model interpretation, we first visualize the learned attention coefficients and conduct an empirical study. Fig. 6 shows the heatmaps of learned attention coefficients of three sensor stations (each for one road segment) and their arbitrary five neighbors, respectively, which are sampled from PeMSD7 dataset. In

[2]With abuse of notation, K in this subsection exclusively denotes the number of connected nodes of each node in a reduced graph.

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.2019.2953888, IEEE Access

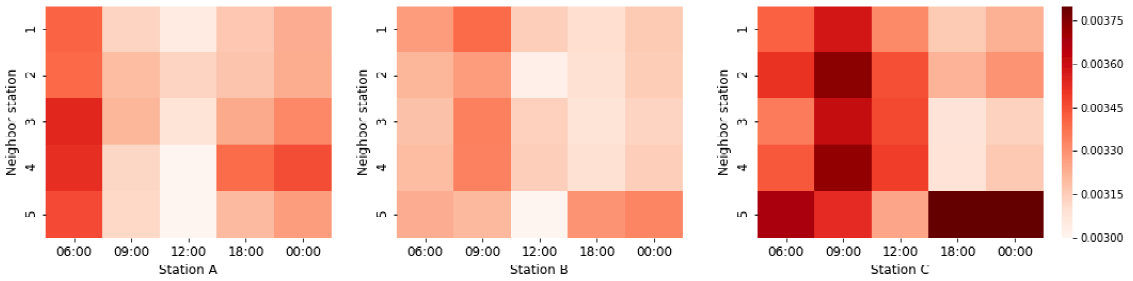


FIGURE 6. Heatmaps of learned attention coefficients varied over time and different neighbors. Three different road segment sensors are included.

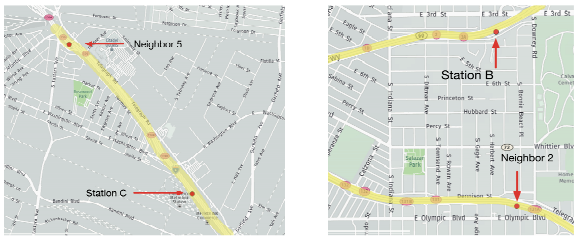


FIGURE 7. **Left:** Two stations located in the upstream and downstream of the same road. **Right:** Two stations located on different roads which are without intersection.

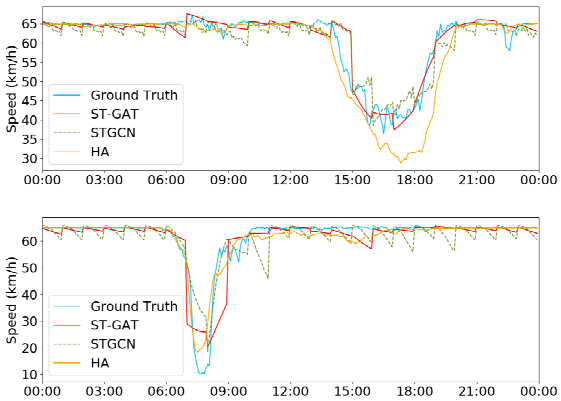


FIGURE 8. Traffic speed forecasting in a day of the dataset PeMSD7. The results of ST-GAT and STGCN are conducted by one hour ahead forecasting.

this figure, the value of attention coefficients can be regarded as the degree of correlations of any two road segments. We can observe that the attention coefficients vary for different neighbors, which suggests the discrepancy of correlations between one road segment and its multiple neighbors. Further investigation indicates some factors affecting the attention coefficients. For example, the attention coefficients between Station C and its Neighbor 5 are relatively large since the two stations are located at the upstream and downstream of the same road as shown in Fig. 7 (left). However, for Station B and its Neighbor 2, their attention coefficients are relatively small since they are located in different roads while they have a close spatial distance as shown in Fig. 7 (right). Moreover, the learned attention coefficients are not fixed but varied in time. For Station A, the attention coefficients with its neighbors at 06:00 in the morning are distinctly larger than those at other times. It can be concluded that spatial dependencies change when traffic status changes over time. In other words, it demonstrates that our model is able to extract dynamic spatial dependencies. In summary, we can intuitively perceive the traffic spatial dependency among these adjacent road segments by learned attentions.

To further demonstrate the effect of graph attention on spatial-temporal feature learning, we compare the prediction performance of our model and the state-of-the-art graph convolution-based method STGCN through showing their one hour ahead forecasting. As shown in Fig. 8, we observe the following phenomena: (1) In general, ST-GAT is more likely to accurately predict the traffic speed value with smaller deviation. A relatively better imitative effect can be observed from the figure. (2) ST-GAT predicts the start and end of the peak hours more accurately (e.g. 15:00 to 21:00). Benefited from the attention mechanism, ST-GAT predictions of each sensor station are more sensitive to speed changes of its neighborhoods. (3) ST-GAT develops predictions with small oscillation in non-peak periods (e.g. 00:00 to 06:00). Comparatively, the oscillation happens to STGCN is more violent in this period. By incorporating LSTM, our model achieves promising temporal feature learning and develops notable long-term prediction accuracy.

e: *Influence of Missing Data and Measurement Noise*

Incomplete data coverage and measurement noise usually happen to traffic data collection on account of limited device deployment, failure of involved sensors, and data transmission errors. According to [45], in the data collection process, the missing rate of raw data can be as high as 15%. Both the measurement noise and missing data introduce unknown influence to speed prediction. Therefore, we are interested in how these two factors influence the performance of the proposed model.

First, we investigate the influence of measurement noise on model performance. In this work, noise is sampled from a Gaussian distribution for each observation value. Specifically, according to IEEE Standard [46], we select the Gaus-

sian distributions with variances which are $0.5\%/1\%/2\%$ of the mean values of data. The final noisy data are generated by imposing the sampled noise on the observation values. New learning models are trained and tested individually using the generated noisy data. Table 3 demonstrates the performance of our model on noisy data. The prediction accuracy only degenerates by a 0.03% (MAPE) when data are disturbed by 2% Gaussian noise. Second, we investigate the tolerance of our model to missing data. We construct datasets incorporating $0.5\%/1\%/2\%$ missing data by randomly selecting $0.5\%/1\%/2\%$ observations and replace their values by 0. Similarly, we independently train and test the new model with the generated datasets. Nonetheless, as shown in Table 3, noticeable performance degradation is observed: a 0.3% (MAPE) penalty is developed when only 0.5% data is missing.

This test demonstrates the robustness of our model on noisy data, which is greatly contributed by the noise-tolerance capability of the deep learning model incorporated. However, the under-performing against missing data expose the deficiency of the proposed model, which may suggest the significance of data integrity to the graph-based attention mechanism. We aim to address this drawback in future work.

V. CONCLUSION AND FUTURE WORK

In this paper, inspired by the research findings of applying attention mechanism on graphs, we propose a novel graph-based deep learning framework ST-GAT for traffic speed forecasting. This model integrates the graph attention network (GAT) and recurrent neural network to jointly learn spatial-temporal dependencies on traffic networks. Specifically, we utilize the attentional graph convolution of GAT on spatial feature learning and regard the learned attention coefficients as the spatial dependency. A LSTM network is integrated to capture the temporal dynamics and improve the performance for relatively long-term forecasting. This framework inherits the advantages of both GAT and LSTM. Experiments on a real-world dataset show that the proposed framework supersedes existing state-of-the-art methods in the literature, which indicates the data potential of graph-based attention mechanism on spatial-temporal learning. In addition, the proposed model develops notable performance on simplified graphs as well as noisy data, which demonstrates its scalability and robustness. These advantages will be practical for both industrial use and scientific research.

In the future, we plan to integrate additional factors such as the road directions, traffic control, and the weather into the prediction to further improve the performance. Additionally, new technologies to enhance the capability of addressing missing raw data will be incorporated.

REFERENCES

[1] I.Wagner, "Car drivers - statistics and facts." https://www.statista.com/topics/1197/car-drivers/. Accessed July 4, 2019.

[2] M. Veres and M. Moussa, "Deep learning for intelligent transportation systems: A survey of emerging trends," IEEE Transactions on Intelligent Transportation Systems, 2019, in press.

[3] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 4, pp. 1624–1639, 2011.

[4] E. Cascetta, Transportation systems engineering: theory and methods, vol. 49. Springer Science & Business Media, 2013.

[5] M. Ben-Akiva, M. Bierlaire, H. Koutsopoulos, and R. Mishalani, "Dynamit: a simulation-based system for traffic prediction," in DACCORD Short Term Forecasting Workshop, pp. 1–12, Delft, The Netherlands, 1998.

[6] 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.

[7] X. Li, G. Pan, Z. Wu, G. Qi, S. Li, D. Zhang, W. Zhang, and Z. Wang, "Prediction of urban human mobility using large-scale taxi traces and its applications," Frontiers of Computer Science, vol. 6, no. 1, pp. 111–121, 2012.

[8] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "Predicting taxi–passenger demand using streaming data," IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 3, pp. 1393–1402, 2013.

[9] S. H. Hosseini, B. Moshiri, A. Rahimi-Kian, and B. N. Araabi, "Short-term traffic flow forecasting by mutual information and artificial neural networks," in 2012 IEEE International Conference on Industrial Technology, pp. 1136–1141, 2012.

[10] J. Guo, W. Huang, and B. M. Williams, "Adaptive kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," Transportation Research Part C: Emerging Technologies, vol. 43, pp. 50–64, 2014.

[11] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," arXiv preprint arXiv:1707.01926, 2017.

[12] H. Chang, Y. Lee, B. Yoon, and S. Baek, "Dynamic near-term traffic flow prediction: system-oriented approach based on past experiences," IET intelligent transport systems, vol. 6, no. 3, pp. 292–305, 2012.

[13] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, "Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions," Expert systems with applications, vol. 36, no. 3, pp. 6164–6173, 2009.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," nature, vol. 521, no. 7553, pp. 436–444, 2015.

[15] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: deep belief networks with multitask learning," IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 5, pp. 2191–2201, 2014.

[16] Y. Jia, J. Wu, and Y. Du, "Traffic speed prediction using deep learning method," in 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), pp. 1217–1222, 2016.

[17] 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.

[18] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction," Sensors, vol. 17, no. 4, p. 818, Apr. 2017.

[19] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks," Sensors, vol. 17, no. 7, p. 1501, Jun. 2017.

[20] J. Wang, Q. Gu, J. Wu, G. Liu, and Z. Xiong, "Traffic speed prediction and congestion source exploration: A deep learning method," in 2016 IEEE 16th International Conference on Data Mining (ICDM), pp. 499–508, 2016.

[21] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," IEEE Signal Processing Magazine, vol. 34, no. 4, pp. 18–42, 2017.

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

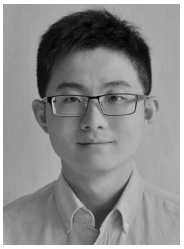
[23] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," arXiv preprint arXiv:1312.6203, 2013.

[24] 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.

[25] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Graph convolutional recurrent neural network: Data-driven traffic forecasting," arXiv preprint arXiv:1707.01926, 2017.

[26] G. Li, M. Müller, A. Thabet, and B. Ghanem, "Can gcns go as deep as cnns?," arXiv preprint arXiv:1904.03751, 2019.
[27] L. N. Do, H. L. Vu, B. Q. Vo, Z. Liu, and D. Phung, "An effective spatial-temporal attention based neural network for traffic flow prediction," Transportation Research Part C: Emerging Technologies, vol. 108, pp. 12–28, 2019.
[28] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," arXiv preprint arXiv:1710.10903, 2017.
[29] E. Choi, M. T. Bahadori, L. Song, W. F. Stewart, and J. Sun, "Gram: graph-based attention model for healthcare representation learning," in Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 787–795, ACM, 2017.
[30] J. Feng, M. Huang, Y. Yang, et al., "Gake: graph aware knowledge embedding," in Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 641–651, 2016.
[31] H. Zhou, T. Young, M. Huang, H. Zhao, J. Xu, and X. Zhu, "Commonsense knowledge aware conversation generation with graph attention.," in IJCAI, pp. 4623–4629, 2018.
[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in Neural Information Processing Systems 30, pp. 5998–6008, Curran Associates, Inc., 2017.
[33] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," arXiv preprint arXiv:1505.00853, 2015.
[34] D. Busbridge, D. Sherburn, P. Cavallo, and N. Y. Hammerla, "Relational graph attention networks," arXiv preprint arXiv:1904.05811, 2019.
[35] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in Advances in Neural Information Processing Systems 27, pp. 3104–3112, Curran Associates, Inc., 2014.
[36] Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, "A hybrid deep learning based traffic flow prediction method and its understanding," Transportation Research Part C: Emerging Technologies, vol. 90, pp. 166–180, 2018.
[37] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," in Artificial Neural Networks, 1999, vol. 2, pp. 850–855(5), 1999.
[38] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[39] Z. Cui, R. Ke, and Y. Wang, "Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction," arXiv preprint arXiv:1801.02143, 2018.
[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.
[41] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," The journal of machine learning research, vol. 15, no. 1, pp. 1929–1958, 2014.
[42] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167, 2015.
[43] 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, pp. 249–256, 2010.
[44] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li, "Deep multi-view spatial-temporal network for taxi demand prediction," in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[45] H. M. Al-Deek, C. Venkata, and S. R. Chandra, "New algorithms for filtering and imputation of real-time and archived dual-loop detector data in i-4 data warehouse," Transportation research record, vol. 1867, no. 1, pp. 116–126, 2004.
[46] W. Gao, Y. Tian, T. Huang, S. Ma, and X. Zhang, "The ieee 1857 standard: Empowering smart video surveillance systems," IEEE Intelligent Systems, vol. 29, no. 5, pp. 30–39, 2013.

CHENHAN ZHANG received the B.Eng. degrees in Telecommunication Engineering from University of Wollongong, Wollongong, Australia, and Zhengzhou University, Zhengzhou, China in 2017 and 2018, respectively. He received the M.S degree in Engineering Management from City University of Hong Kong in 2019. He is currently a research assistant at Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. His research interests include deep learning, intelligent transportation systems, text mining and social network analysis.



JAMES J.Q. YU (S'11–M'15) received the B.Eng. and Ph.D. degree in Electrical and Electronic Engineering from the University of Hong Kong, Pokfulam, Hong Kong, in 2011 and 2015, respectively. He was a post-doctoral fellow at the University of Hong Kong from 2015 to 2018. He is currently an assistant professor at Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China, and an honorary assistant professor at Department of Electrical and Electronic Engineering, the University of Hong Kong. He is also the chief research consultant of GWGrid Inc., Zhuhai and Fano Labs, Hong Kong. His research interests include smart city and urban computing, deep learning, intelligent transportation systems and smart energy systems. He is an Associate Editor of the IET Smart Cities journal.

YI LIU (S'19) received the B.Eng degree in Network Engineering from Heilongjiang University, Harbin, China, in 2019. (Early Graduation Base On Academic Excellence) He is currently a visiting student at Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. His research interests include security and privacy in smart city and urban computing, deep learning, intelligent transportation systems and blockchain.