Spatiotemporal Attention-Based Graph Convolution Network for Segment-Level Traffic Prediction

Duo Li, *Senior Member, IEEE*, and Joan Lasenby

***Abstract*— Traffic prediction, as a core component of intelligent transportation systems (ITS), has been investigated thoroughly in the literature. Nevertheless, timely accurate traffic prediction still remains an open challenge due to the nonlinearities and complex patterns of traffic flows. In addition, most of the existing traffic prediction methods focus on grid-based computing problems (e.g., crowd in-out flow prediction) and point-based computing problems (e.g., traffic detector data prediction), ignoring the segment-based traffic prediction tasks. In this study, we propose an attention-based spatiotemporal graph attention network (AST-GAT) for segment-level traffic speed prediction. In particular, a multi-head graph attention block is designed to capture the spatial dependencies among road segments. Then, a component fusion block is built for speed, volume, and weather information integration. Finally, an attention-based Long short-term memory (LSTM) block is constructed for temporal dependency learning as well as segment-based speed prediction. Experiments on a real-world dataset from Highways England demonstrate that the proposed AST-GAT model outperforms the state-of-the-art baselines, providing an efficient tool for segment-based traffic prediction and therefore filling the gap between point-based and grid-based predictions.**

***Index Terms*— Traffic prediction, deep learning, graph convolution, attention mechanism.**

I. Introduction

TRAFFIC prediction is of great importance to many real-world applications. For example, timely accurate traffic prediction can help traffic operators take adequate preventive measures against congestion and road users take better-informed decisions. In the past few decades, numerous approaches were proposed to enhance the accuracy and efficiency of such predictions. These approaches can be broadly classified into two categories: parametric and nonparametric approaches.

The structures of parametric approaches are predetermined based on certain theoretical assumptions, and their parameters are usually computed using historical data. Some prominent examples of parametric methods include the Auto-Rregressive Integrated Moving Average method (ARIMA) [1], the Seasonal Auto-Regressive Integrated Moving Average method (SARIMA) [2], [3], and the Kalman filter [4], [5]. Nevertheless, models with fixed structures and parameters encounter difficulties in describing the stochastic characteristics of traffic flows. To overcome the limitations of the parametric approaches, researchers apply various nonparametric approaches, such as Support Vector Regression (SVR) [6], and Bayesian modelling [7], in which model structures and parameters depend on concrete issues. Although nonparametric approaches have showed promising capabilities in traffic forecasting studies, their potential for traffic prediction had not been fully utilized until the rise of deep learning models.

With the development of traffic detection technologies, traffic data at high temporal and spatial resolutions has become available. Conventional nonparametric methods with shallow architecture show limited capabilities in handling such high-dimensional and complicated spatiotemporal data. Recently, with the prevalence of deep learning, many deep leaning models such as Convolutional Neural Networks (CNNs) [8], [9], Recurrent Neural Networks (RNNs) [10], and Deep Belief Network (DBN) [11], have recorded considerable success in spatiotemporal traffic data mining tasks due to their powerful hierarchical feature learning ability. RNNs and their variations, e.g., Long Short-Term Memory (LSTM) [12], are applied to capture the temporal features of traffic states. Some techniques originally designed for Natural Language Processing (NLP) tasks, such as Sequence to Sequence (Seq2Seq) [13], [14] and Attention Mechanism [15], [16], have been introduced to enhance the prediction performance of RNNs. In terms of spatial feature extraction, conventional CNNs have proven to be efficient in extracting spatial relationships in Euclidean spaces that are usually represented by images and two-dimensional matrices [17]–[19]. Geometric Deep Learning (GDL) [20] is proposed to process data sampled in non-Euclidean spaces such as graphs. Graph Convolutional Networks (GCNs) [21] are one such technique, which have been widely used to address traffic data-related problems by treating traffic networks as graphs [22]–[24]. As CNNs and RNNs can only respectively capture the spatial and temporal dependencies, most of the existing studies have proposed models by combining CNNs and RNNs to fulfil spatiotemporal traffic data prediction tasks [9], [25]–[29]. Although existing deep learning methods have achieved outstanding prediction performance, they have two limitations that could be overcome.

Manuscript received August 7, 2020; revised January 26, 2021 and March 16, 2021; accepted April 17, 2021. This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) under Project EP/R035199/1. The Associate Editor for this article was S. Siri. *(Corresponding author: Duo Li.)*

Duo Li is with the Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, U.K., and also with the Highway School, Chang'an University, Xi'an 710064, China (e-mail: duoli0725@gmail.com).

Joan Lasenby is with the Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, U.K.

Digital Object Identifier 10.1109/TITS.2021.3078187

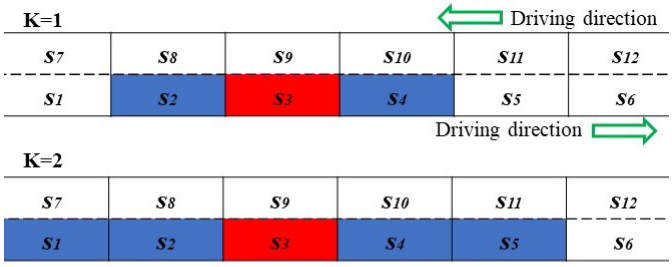
First, the majority of literature focuses on grid-based computing problems (e.g., crowd in-out flow prediction, taxi demand prediction, and traffic accident prediction) and point-based computing problems (e.g., traffic detector data prediction and sharing-bike station demand prediction). Limited attempts have been made to investigate traffic data available at the segment level (e.g., average speed and volume of a road segment). Such data is usually collected from point sources (e.g., traffic detectors and probe vehicles) and preprocessed by transport agencies. Compared to detector-based prediction which may yield redundant and inhomogeneous information, segment-based prediction can provide more straightforward and ready-to-use information for both road operators and users.

Second, GCNs heavily depend on the Laplacian matrix of a graph, which is defined as the difference between the adjacency matrix and diagonal matrix of node degrees. It is assumed that the adjacency matrix of the input graph is constant. Nevertheless, previous studies [23] observed that the adjacency matrix of a road graph could be time-varying and generally intractable due to different traffic patterns during different time spans. In addition, in previous graph-CNN works, the adjacency matrix of a road graph is usually computed based on geographical distances among detection locations. However, using geographical distances might not be able to properly describe spatial correlations among different locations. For example, imaging in a two-way road, detector A is placed immediately upstream of detector C, while detector B is installed in the opposite direction. Although both detector A and B have the same geographical distance to the detector C, the spatial correlations of these two detectors to the detector C can be significantly different.

To tackle these limitations, we propose an attention-based spatiotemporal graph attention network (AST-GAT) for segment-level traffic speed prediction. The proposed method is assessed on a large-scale real-world public dataset provided by Highways England.[1] Compared with existing research works that adopted attention-based spatial-temporal deep learning frameworks, our contributions are summarized as follows.

- To overcome the aforementioned limitations of GCNs, we adopt the multi-head self-attention based graph attention network (GAT) [30] to dynamically model the spatial dependencies among different segments. In the GAT-based model developed by [31], only a LSTM network is used to extract temporal domain features. We design an attention-based LSTM structure to extract temporal dependencies. Information from recent time steps, previous days, and previous weeks are taken into account to capture $recent$, $daily\text{-}periodic$, and $weekly\text{-}periodic$ temporal features.
- We formulate the segment-level traffic prediction as a graph-like computing problem. The road network is modeled as a directed graph using segment connectivity rather than geographical distance. Moreover, we design a component fusion block to assimilate time-variant features (traffic volume, speed, and weather information), and time-invariant features (Annual Average Daily Traffic and road type). The results of our study reveal that external features, such as weather and road property, help improve the prediction accuracy. However, these features are often ignored by previous attention-based spatial-temporal deep learning frameworks, e.g., [15], [16], [27], [31]–[34].

[1] http://tris.highwaysengland.co.uk/



Segment ID	s_1	s_2	s_3	s_4	s_5	s_6
$\boldsymbol{A}^1_{nh}[s_3]$	0	1	1	1	0	0
$\boldsymbol{A}^2_{nh}[s_3]$	1	1	1	1	1	0

Fig. 1. An example of K-hop neighborhood matrix.

II. Notations and Problem Formulation

In AST-GAT, a road network with $\boldsymbol{N}$ segments is modeled as a directed graph $\boldsymbol{G} = (\boldsymbol{S}, \boldsymbol{A})$, where the vertex set $\boldsymbol{S}$ represents different road segments, and the adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ depicts the connectivity between vertices. The elements $\boldsymbol{A}(i, j)$ denote whether segment i and j are connected

$$\boldsymbol{A}(i,j) = \begin{cases} 1, & \text{if } i \text{ and } j \text{ are connected} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

The adjacency matrix $\boldsymbol{A}$ described above is a typical *one*-hop neighborhood matrix. In order to identify spatial correlations among distant segments, we introduce the K-hop neighborhood matrix. For each road segment $i \in \boldsymbol{S}$, its K-hop neighborhoods can be defined as

$$\boldsymbol{Hop}_i(K) = \left\{ j \in \boldsymbol{S} | d(i,j) \le K \right\} \tag{2}$$

where $d(i, j)$ indicates the minimum number of steps needed to move between segments i and j. The K-hop neighborhood matrix $\boldsymbol{A}^K_{nh}$ can be obtained by

$$\boldsymbol{A}^K_{nh} = \mathrm{Ci}(\boldsymbol{A}^K + \boldsymbol{I}) \tag{3}$$

where Ci(.) is a clip function for the matrix by converting each nonzero element to 1, $\boldsymbol{A}^K$ is the K-hop neighborhood matrix without the self-connection, the identity matrix $\boldsymbol{I}$ is added to render the self-connection, and "nh" is an abbreviation of "neighborhood". Thus, $\boldsymbol{A}^K_{nh}(i, j) = 1$ for $j \in \boldsymbol{Hop}_i(K)$ or $i = j$. Fig. 1 illustrates an example of $\boldsymbol{A}^K_{nh}[i]$, where the segment i and its neighbors are in red and blue, respectively.

We define the mean speed and volume of the road segment i at time step t as $v_{i,t}$ and $f_{i,t}$, respectively. The weather observation of the road network at time step t is defined as $weather_t$. Traffic speed forecast is a typical time-series prediction task, i.e., using the information from the past M time steps to predict the traffic speed of each road segment

at time step $t+p$. In this study, the segment-based speed prediction problem is formulated as

$$\hat{\boldsymbol{V}}_{t+p} = \underset{\boldsymbol{V}_{t+p}}{\operatorname{argmax}} \Pr(\boldsymbol{V}_{t+p}|\boldsymbol{V}_t, \ldots, \boldsymbol{V}_{t-M}, \boldsymbol{F}_t, \ldots, \boldsymbol{F}_{t-M}, weather_t, \ldots, weather_{t-M}, RP, \boldsymbol{A}_{nh}^K) \quad (4)$$

where $\hat{\boldsymbol{V}}_{t+p}$ is the set of predicted segment speeds at time step $t+p$; $\boldsymbol{V}_t$ is the set of observed segment speeds at time step t; $\boldsymbol{F}_t$ is the set of observed segment volumes at time step t; RP is the road property, and Pr(.|.) is the conditional probability function.

III. AST-GAT Deep Learning Architecture

In this section, we elaborate on the proposed AST-GAT architecture. An AST-GAT model is comprised of multi-head GAT blocks for spatial correlation extraction, component fusion blocks for combining features from different sources, and an attention-based LSTM block for temporal dependency learning as well as segment-based speed prediction.

A. Spatial Dependency Modeling

Previous GCN-based models (e.g., [22], [25], [35]) assume that spatial dependencies are time-invariant, i.e., spatial dependencies in a road topology graph are computed once and used all the time, ignoring dynamically changing traffic patterns. To address this issue, we employ GATs that assume contributions of neighbor segments to the central segment are not predetermined. The main difference between GCNs and GATs is how the feature representations of neighbor segments are aggregated. For a GCN, a graph convolution operation yields the normalized sum of the features of neighbors

$$h_i^{l+1} = \sigma\left(\sum_{j\in S_i} \frac{1}{c_{ij}} W^l h_j^l\right) \quad (5)$$

where, S_i is the set of adjacent segments which are neighbors of segment i; σ is an activation function; c_{ij} is a standardized constant based on graph structure; l is the current layer; W^l is a shared weight matrix for segment-wise feature transformation; and h_i^l is the hidden feature of layer l for segment i.

GATs extend GCNs by adopting attention mechanisms to learn the relative weights between two connected segments. In a GAT, the transformation from the feature h_i^l to the higher layer feature h_i^{l+1} is realized via

$$z_i^l = W^l h_i^l \quad (6)$$

$$e_{ij}^l = \text{LeakyReLU}(a^l(z_i^l || z_j^l)) \quad (7)$$

$$\alpha_{ij}^l = \frac{\exp(e_{ij}^l)}{\sum_{k\in S_i}\exp(e_{ik}^l)} \quad (8)$$

$$h_i^{l+1} = \sigma\left(\sum_{j\in S_i} \alpha_{ij}^l z_j^l\right) \quad (9)$$

Eq. 6 performs a linear transformation of the h_i^l using a learnable weight matrix W^l. In Eq. 7, a pair-wise un-normalized attention score e_{ij}^l between segment i and segment j is computed through additive attention. It first concatenates z_i^l and z_j^l, then takes a dot product of concatenation and a learnable weight vector a^l, and finally applies a Leaky Rectified Linear Unit (LeakyReLU) activation function [36]. Eq. 8 applies a softmax activation function to normalize the attention score. In Eq. 9, attention scores from neighbors are aggregated to update the higher layer feature h_i^{l+1}, which is similar to GCNs.

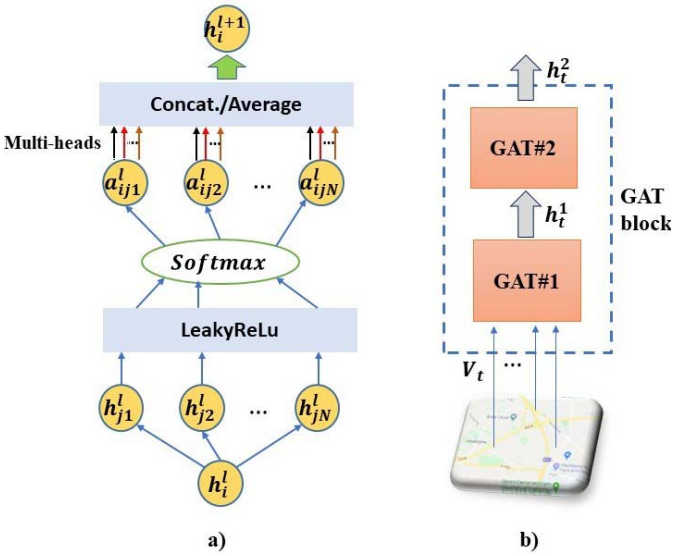


Fig. 2. a) GAT layer with multi-head attention mechanism, and b) structure of the proposed GAT block.

In order to enhance the model capacity and to stabilize the learning process of self-attention, we adopt multi-head attention mechanisms [37] which enable the model to jointly learn attention scores from multiple representation subspaces. As depicted in Fig. 2a, K_{head} independent attention mechanisms perform the GAT convolution operation simultaneously, and then their features are concatenated (for intermediary layers) or averaged (for final layers) to produce the following output feature representation

$$\begin{cases} h_i^{l+1} = \Big\Vert_{K_{head}=1}^{K_{head}} \sigma\left(\sum_{j\in S_i} \alpha_{ij}^{K_{head}} W^{K_{head}} h_j^l\right), \text{concatenate} \\ h_i^{l+1} = \sigma\left(\frac{1}{K_{head}} \sum_{K_{head}=1}^{K_{head}} \sum_{j\in S_i} \alpha_{ij}^{K_{head}} W^{K_{head}} h_j^l\right), \text{average} \end{cases} \quad (10)$$

In this study, GAT blocks are constructed to extract spatial correlations among speeds of different segments at different time steps. As shown in Fig. 2b, every block consists of two GAT layers. For each time step t, the input to the first layer is a set of speeds $\boldsymbol{V}_t = \{v_{t,1}, v_{t,2}, \ldots, v_{t,N}\}, v_{t,i} \in \mathbb{R}^{Feature}$, where $Feature$ denotes the number of features of each segment. The first layer generates a new set of segment features (of potentially different cardinality $Feature'$), $h_t^1 = \{h_{t,1}^1, h_{t,2}^1, \ldots, h_{t,N}^1\}, h_{t,i}^1 \in \mathbb{R}^{Feature'}$. Then, the output features from different attention heads are concatenated and used as the input to the second layer. The second

layer produces another new set of segment features $h_t^2 = \left\{h_{t,1}^2, h_{t,2}^2, \ldots h_{t,N}^2\right\}, h_{t,i}^2 \in \mathbb{R}^{Feature''}$, and the output features from different attention heads are averaged.

B. Component Fusion

As mentioned in the introduction, there is a bivariate equilibrium relationship between traffic volume and speed. Thus, it is necessary to include volume information in segment speed prediction. Similar to speed spatial feature extraction, we use the constructed GAT blocks to model the spatial dependencies among volumes of different segments. For each time step t, the input to the GAT block is $F_t = \left\{f_{t,1}, f_{t,2}, \ldots, f_{t,N}\right\}, f_{t,i} \in \mathbb{R}^{Feature_f}$, and the output of the block is $h_{f,t}^2 = \left\{h_{f,t,1}^2, h_{f,t,2}^2, \ldots, h_{f,t,N}^2\right\}, h_{f,t,i}^2 \in \mathbb{R}^{Feature''_f}$, which can be formulated as

$$F_t \xrightarrow{g_{concat.}^1(\cdot)} h_{f,t}^1 \xrightarrow{g_{aver.}^2(\cdot)} h_{f,t}^2 \tag{11}$$

where $g_{concat.}^1(\cdot)$ indicates the first multi-head convolution layer with concatenation operation, and $g_{aver.}^2(\cdot)$ indicates the second multi-head convolution layer with averaging operation.

After obtaining the spatial representation $h_{f,t}^2$, we apply a gating mechanism to perform component fusion. Specifically, a flow gate is added to regulate the obtained volume information. Then, the speed spatial representation $h_{v,t}^2$ multiplies (element-wise) the gated volume information, given by

$$h_t^c = (h_{v,t}^2) \otimes \sigma(h_{f,t}^2) \tag{12}$$

where $\otimes$ is the element-wise product between tensors, and σ is the sigmoid function.

Since the early 1950s, it has been recognized that weather conditions impact traffic flow and driver behavior [38]. Weather phenomena exert significant impact on traffic flow related parameters, such as free-flow speed and capacity [39], [40]. Therefore, it is reasonable to take into account the influence of weather conditions in traffic speed prediction. In addition, we include two static features relevant to road property, namely, road type (motorway or A-level road, whether a junction) and Annual Average Daily Traffic (AADT). At each time step t, network-wide weather information $weather_t$, road property features, and the fused spatial representation h_t^c are concatenated as X_t which is then used as the input to the following attention-based LSTM block. The entire component fusion process is illustrated in Fig. 3.

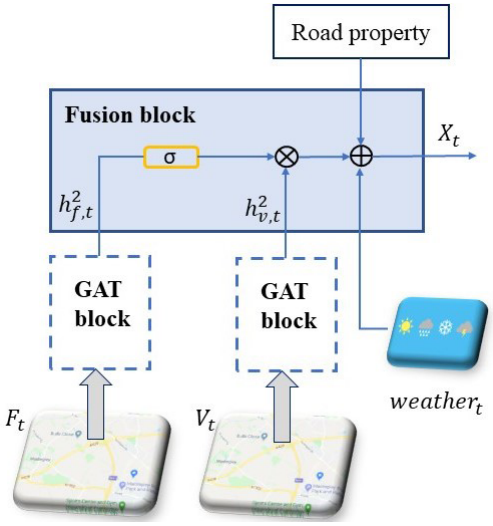


Fig. 3. Structure of component fusion block.

C. Temporal Dependency Modeling

Fig. 4 presents the structure of the proposed attention-based LSTM block. It consists of three parts: two RNNs are employed to respectively model the short-term and long-term dependencies of the historical data, while an attention mechanism is introduced to compute the weighted representation of long-term information.

Previous studies witness the importance of long-term dependency to the traffic prediction problem [15], [16], [41]. Nevertheless, It is a nontrivial task to deal with long-term information via RNNs. The increasing input length may result in vanishing gradients, which significantly reduce the effects of periodicity. Thus, it is important to explicitly construct the input time series so as to prevent vanishing gradients. We construct three types of time series X^R, X^D and X^W along the time axis to collect information from the previous r time steps, previous d days, and previous w weeks, respectively. The *recent* time series $X^R = \{X_{t-r+1}, X_{t-r+2}, \ldots, X_t\}$ consists of the samples that are r steps before the current step, which might be the most important contributor to the future traffic. The *daily-periodic* time series $X^D = \left\{X_{t-r+1}^d, \ldots, X_{t+r}^d, \ldots, X_{t-r+1}^{d-1}, \ldots, X_{t+r}^1,\right\}$ consists of the samples that are r steps before and after the current step on the past d days, which is used to model the daily periodicity of traffic data, e.g., daily peak hours. The *weekly-periodic* time series $X^W = \left\{X_{t-r+1}^w, \ldots, X_{t+r}^w, \ldots X_{t-r+1}^{w-1}, \ldots, X_{t+r}^1,\right\}$ consists of the samples that are r steps before and after the current step on the last w weeks, which have the same week attributes. This type of time series is designed to extract the weekly periodic features, e.g., similar traffic patterns on Mondays. Fig. 5 illustrates an example of time series construction, where interval=15min, $r = 4$, $d = 2$, and $w = 2$.

In this study, we select an LSTM network to perform the temporal feature extraction, which is designed to solve the exploding and vanishing gradients issue of traditional RNNs. For each segment i, the calculation procedure of the LSTM network is shown below

$$I_{t,i} = \sigma(W_{I,i}[h_{t-1,i}; x_{t,i}] + b_{I,i}) \tag{13}$$
$$O_{t,i} = \sigma(W_{O,i}[h_{t-1,i}; x_{t,i}] + b_{O,i}) \tag{14}$$
$$U_{t,i} = \sigma(W_{U,i}[h_{t-1,i}; x_{t,i}] + b_{U,i}) \tag{15}$$
$$\widetilde{C}_{t,i} = \tanh(W_{C,i}[h_{t-1,i}; x_{t,i}] + b_{C,i}) \tag{16}$$
$$C_{t,i} = U_{t,i} \otimes C_{t-1,i} + I_{t,i} \otimes \widetilde{C}_{t,i} \tag{17}$$
$$h_{t,i} = O_{t,i} \otimes \tanh(C_{t,i}) \tag{18}$$

where, I, O, and U represent input gate, output gate, and forget gate, respectively; W_I, W_O, and W_U are the weight matrices; b_I, b_O, and b_U are the corresponding bias vectors;

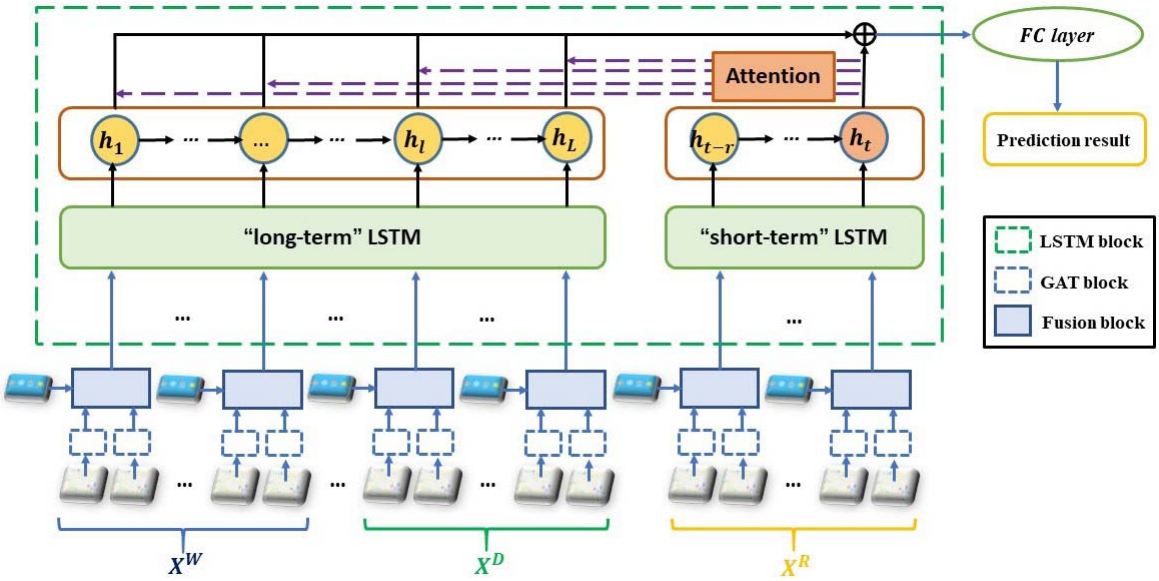


Fig. 4. Structure of the proposed attention-based LSTM block.

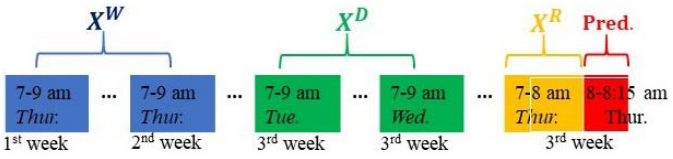


Fig. 5. An example of time series construction.

$\widetilde{C}$ represents a candidate for cell state, which is created by a *tanh* layer; C is the cell state, and h is the output of an LSTM block.

For each segment i, the *recent* time series X_i^R is used as the input to the short-term feature extraction LSTM. The *daily-periodic* time series X_i^D and *weekly-periodic* time series X_i^W are concatenated along the time axis as X_i^{long} which is used as the input to the long-term feature extraction LSTM. The reason behind this is that the importance of long-term features may vary at different time steps. We therefore concatenated them and used an attention layer to capture their dynamic importance. Then, the following calculations are conducted

$$h_{t,i}^{short} = \text{LSTM}(X_{t,i}^R, h_{t-1,i}^R) \tag{19}$$

$$h_{l,i}^{long} = \text{LSTM}(X_{l,i}^{long}, h_{l-1,i}^{long}) \tag{20}$$

where, l is the time index of the long-term time series.

An attention mechanism is incorporated to capture the temporal heterogeneity of traffic patterns. The importance value $\alpha_{l,i}$ is derived by comparing the query $h_{t,i}^{short}$ from Eq. 19 with a set of keys $\left\{h_{1,i}^{long}, \ldots, h_{l,i}^{long}, \ldots, h_{L,i}^{long}\right\}$ from Eq. 20. Formally, the weight $\alpha_{l,i}$ is defined as

$$\alpha_{l,i} = \frac{\exp(\text{score}(h_{t,i}^{short}, h_{l,i}^{long}))}{\sum_{l \in L} \exp(\text{score}(h_{t,i}^{short}, h_{l,i}^{long}))} \tag{21}$$

The score function in Eq. 21 is regarded as a content-based function [42] defined as

$$\text{score}(h_{t,i}^{short}, h_{l,i}^{long}) = q_i^T \tanh(h_{t,i}^{short} W_{X,i} + h_{l,i}^{long} W_{L,i} + b_{X,i}) \tag{22}$$

where, W_L, W_X and b_X are trainable parameters, and q^T is to adjust the dimension of the output. The attention vector $A_{t,i}$ is obtained via a weighted sum of the keys. Then, $A_{t,i}$ and $h_{t,i}^{short}$ are concatenated to preserve both short-term and long-term dependencies for predicting segment and time step. Finally, fully connected layers are added to produce the predicted speed $\widetilde{v}_{t+p,i}$. The above calculations are formulated as

$$A_{t,i} = \sum_{l \in L} h_{l,i}^{long} \alpha_{l,i} \tag{23}$$

$$\widetilde{h}_{t+p,i} = \tanh(W_{fc1,i}[A_{t,i}; h_{t,i}^{short}] + b_{fc1,i}) \tag{24}$$

$$\widetilde{v}_{t+p,i} = W_{fc2,i}\widetilde{h}_{t+p,i} + b_{fc2,i} \tag{25}$$

where, $W_{fc1}, W_{fc2}, b_{fc1}$, and b_{fc2} are trainable parameters.

The proposed AST-GAT model is jointly trained using mean squared error as the loss function, which can be written as

$$loss = \frac{1}{N} \sum_{i \in \boldsymbol{S}} |v_{t+p,i} - \widetilde{v}_{t+p,i}|^2 \tag{26}$$

where, N is the total number of segments, $v_{t+p,i}$ denotes the observed speed of segment i at time step $t+p$, and the vertex set $\boldsymbol{S}$ represents different road segments within the road network.

IV. Case Study

A. Data Collection and Preprocessing

We assess the proposed model on a large-scale real-world public dataset provided by Highways England. All data collected is for a network of roads around Cambridge, U.K. with a total of 60 segments (see Fig. 6), which consists of four

Fig. 6. Layout of the study area, taken from google map[3].

roads, namely, A11(13 miles, 4 junctions), A14 (18 miles, 7 junctions), A428 (2.4 miles) and M11(17 miles, 5 junctions). The size of the segments ranges from 200 to 1300 m, which is predefined in the original dataset. The data contains traffic speed and volume information for each road segment at 15-min intervals. Weather data is obtained from the weather station of the Digital Technology Group (DTG),[2] University of Cambridge. The weather data contains wind speed and rainfall information at 30-min intervals, which is upsampled to 15-min intervals in order to be consistent with the Highways England data format.

In this study, three-month data from 01/10/2019 to 31/12/2019 is collected and divided into two subsets. The data for the last week of each of the three months is used as the testing set, and the rest of the data is used as the training set. 10% of the training set is selected to be the validation set. We use Z-Score normalization that describes the position of a raw score in terms of its distance from the mean, to convert traffic speed, volume, wind speed and rainfall to a [-1, 1] scale. After prediction, the predicted values are denormalized and then used for evaluation.

B. Settings

In this study, all experiments are conducted on GPUs in Google Colaboratory [43] or more commonly referred to as "Google Colab". The GPUs available in "Google Colab" vary over time and often include Nvidia K80s, T4s, P4s and P100s. We choose $r = 4$, $d = 2$, and $w = 1$ to construct $recent$, $daily\text{-}periodic$, and $weekly\text{-}periodic$ time series, and a 2-hop neighborhood adjacency matrix is built.

We set the hyperparameters based on the model performance on the validation set. The number of attention heads in a GAT is set to 8, and the hidden units of an LSTM is set to 128. AST- GAT is optimized via the Adam optimizer [44]. The batch size is set to 64. A cosine decay learning rate schedule with restarts [45] is applied, in which the initial learning rate is 0.01; the number of steps of the first decay is 20; the iteration number coefficient is 2, and the learning rate coefficient is 0.5. We also use early-stop and dropout [46] (dropout rate = 0.5) in all the experiments to prevent overfitting.

[2]https://www.cl.cam.ac.uk/research/dtg/weather/
[3]https://www.google.com/maps

C. Evaluation Metric and Baselines

To verify the effectiveness of the proposed AST-GAT model, we compare our proposed model with the following classic and state-of-the-art machine learning models, including: 1) Historical Average (HA) that models the traffic speed as a seasonal pattern and uses the average of previous seasons as the prediction; 2) Auto-Rregressive Integrated Moving Average method (ARIMA) [1]; 3) Full-Connected LSTM (FC-LSTM) [47]; 4) Diffusion Convolutional Recurrent Neural Network (DCRNN) [22]; 5) Spatio-Temporal Graph Convolutional Networks (STGCN) [19]; and 6) Attention based Spatial-Temporal Graph Convolutional network (ASTGCN) [15]. Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) are chosen as evaluation metrics, which are defined as

$$\text{RMSE} = \sqrt{\frac{1}{N_p}\sum_{t=1}^{N_p}\left(X_t - \hat{X}_t\right)^2} \tag{27}$$

$$\text{MAPE} = \frac{1}{N_p}\sum_{t=1}^{N_p}\left|\frac{X_t - \hat{X}_t}{X_t} \times 100\%\right| \tag{28}$$

where N_p is the number of predictions; X_t and $\hat{X}_t$ respectively denote the observed and predicted traffic speeds at time t.

D. Result Analysis

Table I shows the results of AST-GAT and aforementioned baselines on the Highways England dataset for 15-min, 30-min and 45-min ahead predictions. Note that all the evaluation metrics are calculated using the predicted and observed speeds (in km/h). The following phenomena are observed from performance comparison. First, traditional statistical and machine learning methods have been greatly outperformed by deep learning methods (DCRNN, STGCN, ASTGCN and AST-GAT) adopting a CNN+RNN structure, especially for long-term forecasting. This might be due to the fact that temporal dependency becomes increasingly non-linear with the growth of the horizon. Besides, methods like HA, ARIMA and FC-LSTM can only process time series, ignoring spatial correlations among different road segments. Second, our proposed method records the best performance among all the tested approaches in terms of all evaluation metrics, indicating the effectiveness of our method. It is worth noting that the benefits due to the proposed model rise with the increase of the prediction horizon, even when compared against to the graph convolution-based methods like STGCN and DCRNN. For example, the RMSE of AST-GAT outperforms DCRNN by 0.45 (15 min), 0.50 (30 min), and 0.65 (45 min). This might be due to the application of the attention mechanism. The proposed model slightly outperforms ASTGCN which also adopts the attention mechanism for both spatial and temporal feature extraction. This improvement might be attributed to additional information provided by external features, such as weather and road property.

In Fig. 7a, we show the prediction values of DCRNN and AST-GAT for segment 199049801 on October 24 under the

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT APPROACHES FOR TRAFFIC SPEED FORECASTING

Model	RMSE			MAPE (%)		
	15-min	30-min	45-min	15-min	30-min	45-min
HA	9.53	9.53	9.53	7.48	7.48	7.48
ARIMA	11.95	12.35	12.67	9.22	9.44	9.68
FC-LSTM	8.26	8.94	10.14	6.30	6.85	7.50
DCRNN	5.64	8.02	9.14	4.28	6.17	6.95
STGCN	5.37	7.65	8.96	4.11	5.95	6.84
ASTGCN	5.21	7.55	8.54	3.98	5.82	6.63
AST-GAT	**5.19**	**7.52**	**8.49**	**3.96**	**5.78**	**6.57**

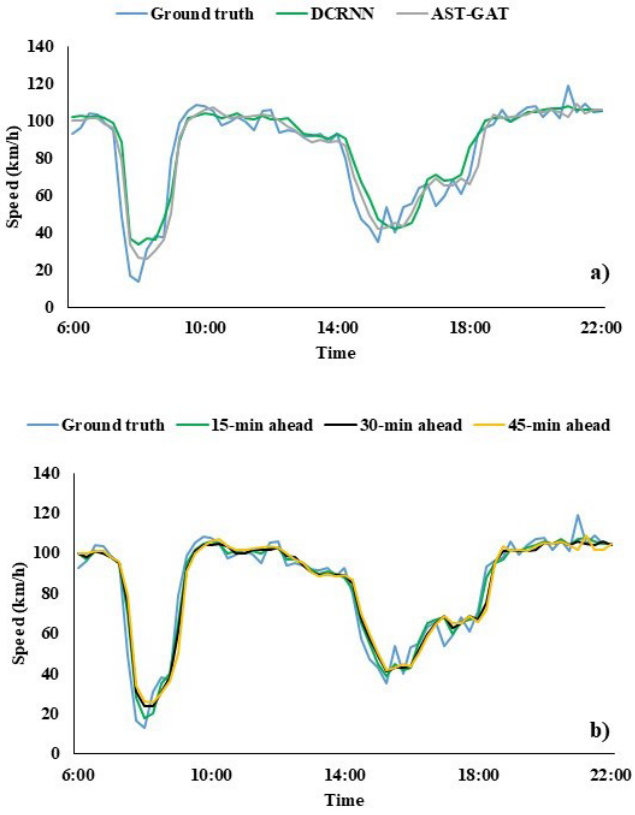


Fig. 7. Prediction vs. ground truth for segment 199049801 on October 24: a) 45-min ahead prediction using AST-GAT and DCRNN, and b) AST-GAT with different predction horizons.

TABLE II
IMPACT OF EXTERNAL FEATURES AND SERIES SIZES

	RMSE	MAPE (%)
WS+RF+RP, w=1	8.49	6.57
WS, w=1	8.61	6.68
RF, w=1	8.52	6.63
RP, w=1	8.70	6.75
WS+RF+RP, w=2	8.47	6.54
WS+RF+RP, w=3	8.46	6.52

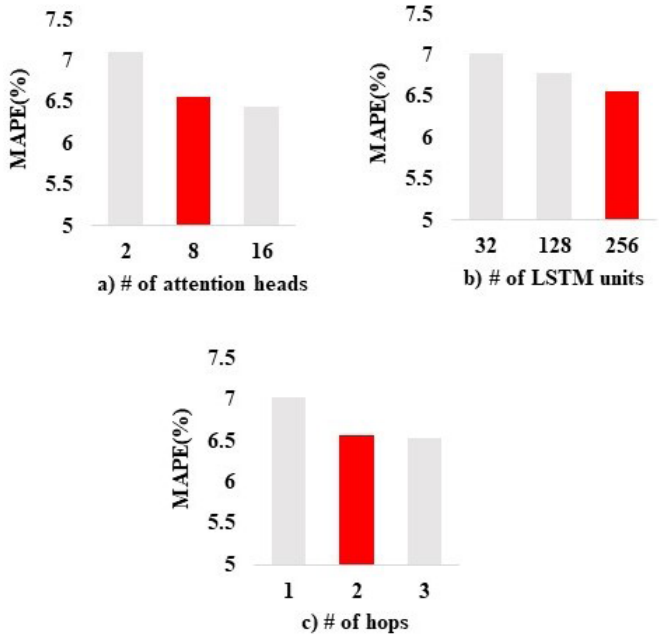


Fig. 8. Effects of different parameters on model performance: a) attention head, b) hidden unit, c) neighbouring hop.

45-min horizon. It is shown that the prediction values of DCRNN slightly lags behind the ground truth when the traffic status oscillates seriously, while the proposed model is more likely to accurately predict abrupt changes in the traffic speed. Fig. 7b shows the actual series and predicted series of AST-GAT with different horizons. It can be observed that, for different horizons, the predicted series can closely follow the actual series.

We investigate the impact of different external features and series sizes on the results, as shown in table II. In our original model (the first row of the table), wind speed (WS), rainfall (RF), road property (RP), and previous one-week ($w = 1$) data are provided. We can observe that generally more features lead to lower errors. The rainfall information has the most significant effect on the results among all the external features. Although increasing the size of long-term time series also helps improve prediction accuracy, this improvement is not remarkable.

Fig. 8 illustrates the effects of different parameter settings on model performance. A 45-min ahead prediction is used as the benchmark test. We test three critical parameters, namely, the number of attention heads in a GAT, the number of hidden units in an LSTM and the number of hops used to generate an

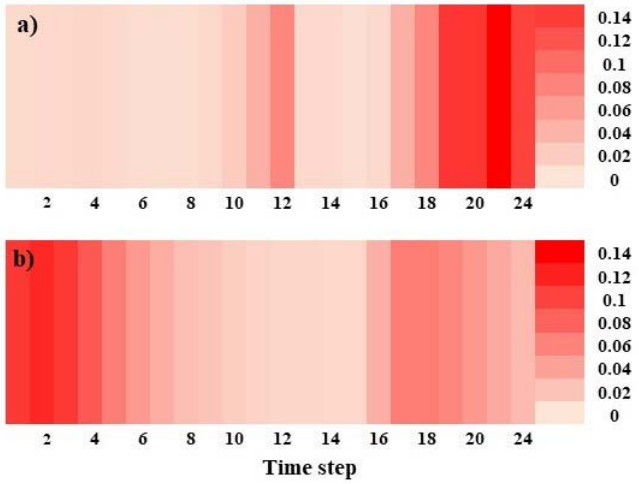


Fig. 9. Temporal attention coefficients for a) segment 199137191, and b) segment 199049101.

adjacency matrix. Specifically, these values are set to 2/8/16, 32/128/256, and 1/2/3, respectively, whose default values are highlighted in red. It is shown that larger hidden units and attention heads improve the prediction accuracy. A significant reduction in RMSE is observed when the number of attention heads is increased from 2 to 8, demonstrating the efficiency of the adopted multi-head attention mechanism. The increase of the number of hops also leads to performance improvement; this might be because more hops (i.e., more neighbours) enable the model to capture broader spatial dependency. It should be noted that we select $k_{head} = 8$ and $K = 2$ rather than larger values in this study, as the selected settings can produce similar prediction accuracy with much less computational cost when compared against higher parameter values.

As shown in Table I, applying attention mechanism can improve the effectiveness of our model for spatiotemporal data prediction. To better understand the model interpretation, we visualize the learned temporal attention coefficients. Fig. 9 presents the temporal attention coefficients for two randomly selected road segments. It is observed that the temporal attention mechanism can generate different attention coefficient distributions for different road segments. In Fig. 9a (segment 199137101), distant steps have higher attention scores; whereas, in Fig. 9b (segment 199049101), recent steps are more important for prediction.

V. Conclusion

In this paper, a novel attention-based spatial-temporal graph attention network (AST-GAT) is proposed and successfully applied to segment-based speed prediction. The AST-GAT model is comprised of three types of blocks, namely, a multi-head GAT block for spatial correlation extraction, a component fusion block for combing derived speed, volume and weather features, and an attention-based LSTM block for temporal dependency learning as well as segment-based speed prediction. Such architecture inherits the advantages of GATs, LSTMs and attention mechanisms. Experiments on a large-scale real-world dataset show that our model outperforms existing state-of-the-art methods in the literature. In the future, we will generalize our framework to address a much broader set of traffic spatiotemporal prediction tasks.

References

[1] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining Kohonen maps with ARIMA time series models to forecast traffic flow," *Transp. Res. C, Emerg. Technol.*, vol. 4, no. 5, pp. 307–318, Oct. 1996.

[2] G. Shi, J. Guo, W. Huang, and B. M. Williams, "Modeling seasonal heteroscedasticity in vehicular traffic condition series using a seasonal adjustment approach," *J. Transp. Eng.*, vol. 140, no. 5, May 2014, Art. no. 04014012.

[3] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, Nov. 2003.

[4] L. L. Ojeda, A. Y. Kibangou, and C. C. de Wit, "Adaptive Kalman filtering for multi-step ahead traffic flow prediction," in *Proc. Amer. Control Conf.*, Jun. 2013, pp. 4724–4729.

[5] J. Guo, W. Huang, and B. M. Williams, "Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 50–64, Jun. 2014.

[6] 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 Syst. Appl.*, vol. 36, no. 3, pp. 6164–6173, Apr. 2009.

[7] J. Wang, W. Deng, and Y. Guo, "New Bayesian combination method for short-term traffic flow forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 79–94, Jun. 2014.

[8] 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.

[9] X. Liang, G. Wang, M. R. Min, Y. Qi, and Z. Han, "A deep spatio-temporal fuzzy neural network for passenger demand prediction," in *Proc. SIAM Int. Conf. Data Mining*, 2019, pp. 100–108.

[10] J. W. C. van Lint, S. P. Hoogendoorn, and H. J. van Zuylen, "Freeway travel time prediction with state-space neural networks: Modeling state-space dynamics with recurrent neural networks," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1811, no. 1, pp. 30–39, Jan. 2002.

[11] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] D. Chai, L. Wang, and Q. Yang, "Bike flow prediction with multi-graph convolutional networks," in *Proc. 26th ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, Nov. 2018, pp. 397–400.

[14] T. Pamuła, "Impact of data loss for prediction of traffic flow on an urban road using neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 1000–1009, Mar. 2019.

[15] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 922–929.

[16] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 5668–5675.

[17] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1–7.

[18] X. Ma, H. Zhong, Y. Li, J. Ma, Z. Cui, and Y. Wang, "Forecasting transportation network speed using deep capsule networks with nested LSTM models," *IEEE Trans. Intell. Transp. Syst.*, early access, Apr. 16, 2020, doi: 10.1109/TITS.2020.2984813.

[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] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017.

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

[22] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," 2017, *arXiv:1707.01926*. [Online]. Available: http://arxiv.org/abs/1707.01926

[23] Z. Diao, X. Wang, D. Zhang, Y. Liu, K. Xie, and S. He, "Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 890–897.
[24] X. Geng *et al.*, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 3656–3663.
[25] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," 2017, *arXiv:1709.04875*. [Online]. Available: http://arxiv.org/abs/1709.04875
[26] 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 *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1720–1730.
[27] Z. Zhang, M. Li, X. Lin, Y. Wang, and F. He, "Multistep speed prediction on traffic networks: A graph convolutional sequence-to-sequence learning approach with attention mechanism," 2018, *arXiv:1810.10237*. [Online]. Available: http://arxiv.org/abs/1810.10237
[28] Z. Yuan, X. Zhou, and T. Yang, "Hetero-ConvLSTM: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 984–992.
[29] S. Zhang, S. Li, X. Li, and Y. Yao, "Representation of traffic congestion data for urban road traffic networks based on pooling operations," *Algorithms*, vol. 13, no. 4, p. 84, Apr. 2020.
[30] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*. [Online]. Available: http://arxiv.org/abs/1710.10903
[31] C. Zhang, J. J. Q. Yu, and Y. Liu, "Spatial-temporal graph attention networks: A deep learning approach for traffic forecasting," *IEEE Access*, vol. 7, pp. 166246–166256, 2019.
[32] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 1, pp. 1234–1241.
[33] X. Shi, H. Qi, Y. Shen, G. Wu, and B. Yin, "A spatial-temporal attention approach for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, early access, Apr. 13, 2020, doi: 10.1109/TITS.2020.2983651.
[34] H. Zheng, F. Lin, X. Feng, and Y. Chen, "A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 9, 2020, doi: 10.1109/TITS.2020.2997352.
[35] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial-temporal graph modeling," 2019, *arXiv:1906.00121*. [Online]. Available: http://arxiv.org/abs/1906.00121
[36] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 2015, *arXiv:1505.00853*. [Online]. Available: http://arxiv.org/abs/1505.00853
[37] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[38] J. Tanner, "Effect of weather on traffic flow," *Nature*, vol. 169, no. 4290, p. 107, 1952.
[39] K. M. Kockelman, "Changes in flow-density relationship due to environmental, vehicle, and driver characteristics," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1644, no. 1, pp. 47–56, Jan. 1998.
[40] R. Hranac, E. D. Sterzin, D. Krechmer, H. Rakha, and M. Farzaneh, "Empirical studies on traffic flow in inclement weather," Federal Highway Admin., Washington, DC, USA, Tech. Rep. FHWA-HOP-07-073, 2006.
[41] A. Zonoozi, J.-J. Kim, X.-L. Li, and G. Cong, "Periodic-CRN: A convolutional recurrent model for crowd density prediction with recurring periodic patterns," in *Proc. IJCAI*, 2018, pp. 3732–3738.
[42] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," 2015, *arXiv:1508.04025*. [Online]. Available: http://arxiv.org/abs/1508.04025
[43] E. Bisong, *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Berkeley, CA, USA: Springer, 2019.
[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[45] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," 2016, *arXiv:1608.03983*. [Online]. Available: http://arxiv.org/abs/1608.03983
[46] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[47] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

Duo Li (Senior Member, IEEE) received the B.Eng. degree in civil engineering from the Huazhong University of Science and Technology, China, in 2010, the M.S. degree in civil engineering from The University of Queensland, Australia, in 2011, and the Ph.D. degree in civil engineering from The University of Auckland, New Zealand, in 2015.

From 2018 to 2019, he was a Humboldt Research Fellow with the Institute of Traffic Systems, German Aerospace Center (DLR). From 2015 to 2018, he was a Lecturer with the Highway School, Chang'an University, where he has been an Associate Professor, since 2019. Since 2020, he has been a Research Associate with the Department of Engineering, University of Cambridge.

Joan Lasenby received the B.A. degree in mathematics and the Ph.D. degree in radio astronomy from the University of Cambridge, in 1982 and 1986, respectively.

She is currently a Professor of image and signal analysis with the Signal Processing and Communications Group, the Deputy Head of the Department (Graduate Studies) with the Department of Engineering, University of Cambridge, and a fellow and the Director of Studies at Trinity College Cambridge.