Transportation Research Part C 105 (2019) 297–322

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies

Zhengchao Zhang[a], Meng Li[a,b], Xi Lin[a,b], Yinhai Wang[c,a], Fang He[d,b,*]

[a] Department of Civil Engineering, Tsinghua University, Beijing 100084, PR China
[b] Tsinghua-Daimler Joint Research Center for Sustainable Transportation, Tsinghua University, Beijing 100084, PR China
[c] Department of Civil and Environmental Engineering, University of Washington, Seattle, WA 98195, USA
[d] Department of Industrial Engineering, Tsinghua University, Beijing 100084, PR China

ARTICLE INFO

Keywords:
Traffic forecasting
Deep learning
Attention mechanism
Graph convolution
Multistep prediction
Sequence-to-sequence model

ABSTRACT

Multistep traffic forecasting on road networks is a crucial task in successful intelligent transportation system applications. To capture the complex non-stationary temporal dynamics and spatial correlations in multistep traffic-condition prediction, we propose a novel deep learning framework named attention graph convolutional sequence-to-sequence model (AGC-Seq2Seq). In the proposed deep learning framework, spatial and temporal dependencies are modeled through the Seq2Seq model and graph convolution network separately, and the attention mechanism along with a newly designed training method based on the Seq2Seq architecture is proposed to overcome the difficulty in multistep prediction and further capture the temporal heterogeneity of traffic pattern. We conduct numerical tests to compare AGC-Seq2Seq with other benchmark models using two real-world datasets. The results indicate that our model yields the best prediction performance in terms of various prediction error measures. Furthermore, the variations of spatio-temporal correlations of traffic conditions under different perdition steps and road segments are revealed.

1. Introduction

Automobile use has significantly increased in the past few decades owing to the steady development in both technology and economy. However, the increased automobile use has resulted in a series of social problems such as traffic congestion, traffic accidents, energy overconsumption, and carbon emissions (Gao et al., 2011). The intelligent transportation system (ITS) has been considered as a promising solution to improve transportation management and services (Qureshi and Abdullah, 2013; Lin et al., 2017). The success of ITS applications relies on accurate and timely traffic status information.

It has been proved that system-level travelers' dynamic route guidance based on the traffic prediction can effectively reduce peak-hour congestions through guiding the traffic flow to realize balanced distribution on road networks (Liebig et al., 2017). The T-Drive system (Yuan et al., 2011; 2013), which provides the real-time routing server for end users according to the 15-min time slot traffic prediction, can save about 16% of trip time validated by the GPS log data from 33,000 taxis in Beijing. As for the traffic accidents, the proactive traffic management strategy incorporating the future traffic condition could improve traffic safety through optimizing the adaptive traffic control models on the premise of traffic safety (Hashemi and Abdelghany, 2015, 2016; Fang and Tu, 2018). The applications with high-precision traffic prediction (e.g., advanced traffic management systems and advanced traveler information

* Corresponding author at: Department of Industrial Engineering, Tsinghua University, Beijing 100084, PR China.
E-mail address: fanghe@tsinghua.edu.cn (F. He).

https://doi.org/10.1016/j.trc.2019.05.039
Received 23 October 2018; Received in revised form 22 April 2019; Accepted 29 May 2019
Available online 11 June 2019

systems) not only benefit travelers' route planning and departure time scheduling but also provide insightful information for proactive traffic management strategy to improve traffic efficiency and safety. Therefore, short-term traffic flow forecasting has always attracted many scholars' interest (Vlahogianni et al., 2004).

Substantial efforts have been conducted to develop methods for traffic prediction in the literature; however, some major challenges remain. Vlahogianni et al. (2014) provided a comprehensive review of the entire spectrum of the short-term traffic forecasting literature up to 2014 and reported the following potential directions for future research:

- Network-level short-term traffic prediction using data-driven approaches remains a challenging task, which deserves more investigation such as exploring how to incorporate network topology into prediction paradigm.
- Multistep prediction to obtain a relative long-term future traffic condition is more adapted to practical ITS applications.
- Research on incorporating both temporal characteristics of traffic flow and spatial dependencies on traffic network still deserves more comprehensive investigation.

According to the above-mentioned directions, we devote to multistep traffic forecasting on road networks through simultaneously considering the spatial and temporal dependencies of traffic conditions. This topic is challenging primarily due to the non-Euclidean topology structure of traffic networks, the stochastic characteristic of the time-varying traffic patterns, and inherent difficulty in multistep prediction. Hence, we propose a novel deep learning structure, named the **a**ttention **g**raph **c**onvolutional **seq**uence-**to**-**seq**uence model (AGC-Seq2Seq). Specifically, we integrate the graph convolutional network and attention mechanism into a Seq2Seq framework to develop the prediction model that can depict the spatio-temporal correlations in multistep traffic prediction. Furthermore, considering that the existing training method for the Seq2Seq model is not suitable for time-series problems, we hereby design a new training method in our proposed framework. To summarize, the primary contributions of this paper are listed as follows:

- We propose a novel deep learning framework, named AGC-Seq2Seq, which extracts the features from temporal and spatial domains collaboratively through the Seq2Seq model and graph convolution layer. To overcome the multistep prediction challenge and capture the temporal heterogeneity of urban traffic pattern, the attention mechanism is further incorporated into the model.
- We design a new training method for the Seq2Seq framework aiming at multistep traffic prediction to replace the existing ones (e.g., teacher forcing and scheduled sampling). It coordinates multidimensional features (e.g., historical statistic information and time-of-day) with spatio-temporal speed variables in one end-to-end deep learning structure and enables the input for the testing periods to agree with the training periods.
- Validated by two real-world datasets provided by A-map, the proposed model yields a significant improvement over other state-of-the-art benchmarks in terms of various major error measures under different prediction intervals. As the byproducts of the proposed model, we explore the variation of spatial and temporal correlations for traffic conditions under different prediction steps and road segments. It reveals the capability of the proposed model to capture the spatio-temporal dependencies and also illustrates the model's interpretability.

The remainder of this paper is organized as follows. Section 2 first reviews the existing research. Section 3 formulates the short-term traffic speed forecasting problem, and describes the structure and mathematical formulation of the proposed AGC-Seq2Seq model. Section 4 compares the prediction performances of the proposed model with other benchmark models based on the two real-world datasets and presents model interpretation. Finally, Section 5 concludes the paper.

2. Literature review

Traffic flow/condition forecasting has been studied for decades, and various emerging methods have been constantly introduced to model traffic characteristics. With the rapid development of real-time traffic data collection methods, data-driven approaches through mass historical data to capture similar traffic patterns prevail in recent years. The data-driven methods in previous work can be divided into three major representative categories, i.e. statistical models, shallow machine learning models and deep learning models.

Statistical models can predict future values based on previously observed values by time-series analysis. The autoregressive integrated moving average (ARIMA) model (Ahmed and Cook, 1979), Kalman filter (Okutani and Stephanedes, 1984), and their variations (Williams and Hoel, 2003; Guo et al., 2014) are among the most consolidated approaches. However, simple time-series models typically rely on the stationary assumption, which is inconsistent with non-stationary characteristics of urban traffic dynamics. Specifically, for multistep prediction, the posterior predicted values are based on the prior predicted values; thus, the prediction errors could propagate step by step. In this context, it is difficult to satisfy the high-precision requirement using simple time-series models.

Meanwhile, machine learning methods have shown promising capabilities in traffic forecasting studies. The artificial neural network model (Vlahogianni et al., 2005), Bayesian networks (Fusco et al., 2016), support vector machine model (Castro-Neto et al., 2009), K-nearest neighbors model (Zhang et al., 2013; Habtemichael and Cetin, 2016; Cai et al., 2016) and random forest model (Hamner, 2011) all yield satisfactory results in traffic flow forecasting. However, the performances of machine learning models depend heavily on manually selected features, and well-recognized guidelines to choose the appropriate features are not available in general since the key features are problem-wise. Therefore, using elementary machine learning approaches may not yield the prospective outcomes for complicated prediction tasks.

More recently, deep learning models have been widely and successfully employed in computer science; meanwhile, it has drawn substantial attention in the transportation field. Huang et al. (2014) employed the deep belief network for unsupervised feature learning, which was proven efficient in traffic flow prediction. Lv et al. (2015) applied a stacked auto encoder model to learn generic traffic flow features. Ma et al. (2015) used the long short-term memory neural network (LSTM) to capture nonlinear traffic dynamics effectively. Polson and Sokolov (2017) combined L_1 regularization and a multilayer network activated by the tanh function to detect the sharp nonlinearities of traffic flow. However, the models with deep architectures above do not distinguish spatial variables across topological adjacency (e.g. the different effects of upstream and downstream neighbors), which will definitely compromise the effects of capturing spatial correlations.

Meanwhile, convolutional neural networks (CNN) offer an efficient architecture to extract meaningful statistical patterns in large-scale, and high-dimensional datasets. The capability of CNNs in learning local stationary structures (the statistical properties of any part of the structure should be homogenous, e.g., 2D images) resulted in breakthroughs in image and video recognition tasks (Defferrard et al., 2016). In transportation, efforts have been conducted to apply CNN structures (variants of classical CNN model, which usually reserves the convolution layer and modifies the other parts according to specific tasks) to extract spatial correlations on traffic networks.

Ma et al. (2017) proposed a deep convolutional neural network for traffic speed prediction, where spatio-temporal traffic dynamics are converted to images. Wang et al. (2017) processed an expressway as a band image, and subsequently proposed the error-feedback recurrent convolutional neural network structure for continuous traffic speed prediction. Ke et al. (2017) partitioned the urban area into uniform grids and subsequently combined a convolutional layer with an LSTM layer to predict the on-demand passenger demand in each grid. All of the aforementioned research converted traffic network to regular grids because the CNNs are restricted to processing Euclidean-structured data. However, the time series on road networks in traffic forecasting are continuous sequences distributed over a topology graph, which is a typical representative of non-Euclidean-structured data (Narang et al., 2013); in this case, the original CNN structure may not be applicable. To fill this gap, the graph convolutional network (GCN) was developed to generalize the convolution on non-Euclidean domains in the context of spectral graph theory (Kipf and Welling, 2016). Several newly published studies conducted graph convolution on traffic prediction. Spectral-based graph convolution was adopted and combined with temporal convolution (Yu et al., 2018) and the recurrent neural network (RNN) (Li et al., 2017) to forecast traffic states. Later, Cui et al. (2018) applied high-order graph convolution to learn the interactions between links on the traffic network. The studies above do not directly define the graph convolution on road networks, but construct the traffic detector graph through computing the pairwise distances between sensors with threshold Gaussian kernel. Moreover, the model's interpretability with traffic domain knowledge is neglected, either.

To summarize, the evolution of traffic conditions on urban networks exhibits spatial and temporal dependencies, substantially (Ermagun et al., 2017; Jiang et al., 2017). So embedding the temporal and spatial components can effectively augment traffic forecasting. However, most previous studies select the spatially relevant links by prejudgment or correlation-coefficient analysis, and don not consider the temporal correlations in multistep, which may distort the accuracy of models or cause error accumulation (Ermagun and Levinson, 2018b). Some superior statistical methods (e.g., spatial vector auto regression and space-time autoregressive integrated moving average etc.) and initial deep learning models (e.g, convolutional LSTM etc.) are utilized to capture spatio-temporal information, but they cannot fully reveal the sophisticated traffic patterns. In this paper, we are devoted to proposing a customized deep learning framework, which integrates the attention mechanism and the graph convolutional network into a Seq2Seq model structure, to simultaneously capture the complex non-stationary temporal dynamics and spatial dependencies in multistep traffic condition prediction.

3. Deep learning framework

3.1. Preliminaries

In this subsection, we interpret the definitions and notations of the variables used herein.

(1) Road network topology

The road network is modeled as a directed graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$ according to the driving direction, where the node set $\mathcal{N}$ represents the intersections (detectors or selected demarcation points on the freeway), and the link set $\mathcal{L}$ represents the road segments, as shown in Fig. 1. $\boldsymbol{A}$ is the adjacency matrix of the link set, and the dummy variable $\boldsymbol{A}(i, j)$ denotes whether link i and link j are connected, i.e., $\boldsymbol{A}(i,j) = \begin{cases} 1, & l_i \text{ and } l_j \text{ are connected along driving direction} \\ 0, & \text{otherwise} \end{cases}$.

(2) Traffic speed

The speed at the t^{th} time slot (e.g., 5 min) of road segment l_i ($\forall\, l_i \in \mathcal{L}$) is defined as the average speed of floating cars during this time interval on the road segment, which is denoted by v_t^i. The speed of the road network at the t^{th} time slot is defined as the vector $\boldsymbol{V}_t \in \mathbb{R}^{|\mathcal{L}|}$ ($|\mathcal{L}|$ is the cardinality of link set $\mathcal{L}$ in the underlying road network), where the i^{th} element is $(\boldsymbol{V}_t)_i = v_t^i$.

As a classical time-series prediction problem, the nearest m-step observation data can provide valuable information for multistep traffic speed forecasting. In addition to the real-time traffic speed information, some exogenous variables such as the time-of-day,

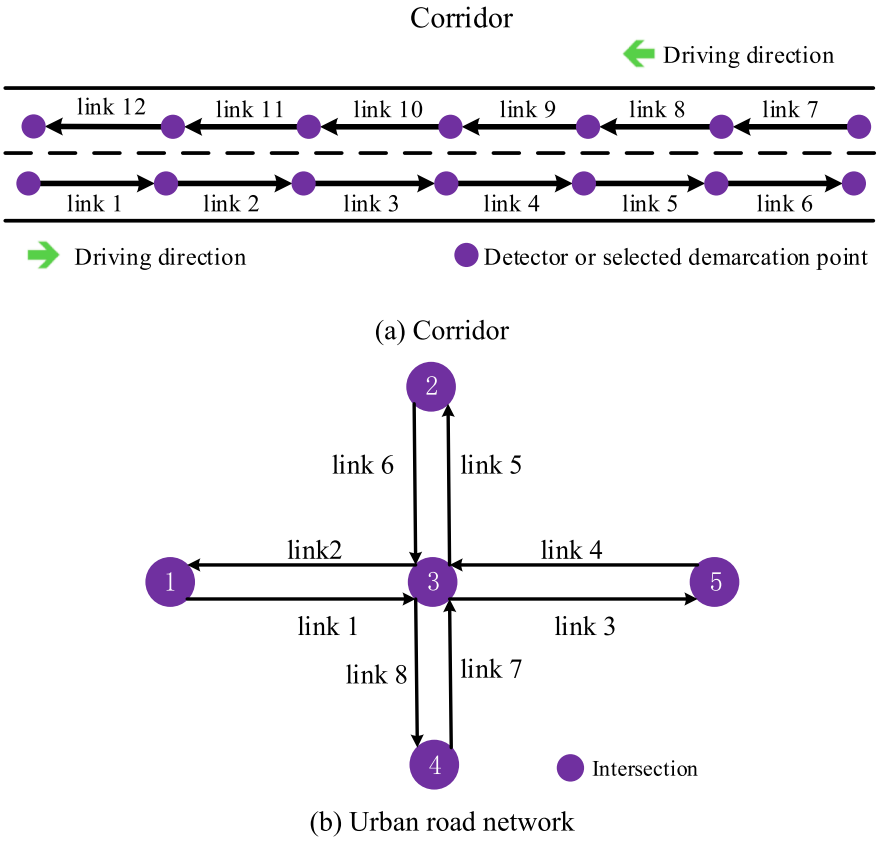


Fig. 1. Topology of traffic networks.

weekday-or-weekend, and historical statistic information are also helpful to predict the future traffic speed. We introduce these variables in the following part.

(3) Time-of-day and weekday-or-weekend

Because the speed of each road segment is aggregated as the average value in 5 min, the time-of-day is transformed into an ordered integer N, e.g., 00:00–00:05 as $N_t = 1$, and 7:00–7:05 as $N_t = 85(7 * 12 + 1)$. The weekday-or-weekend is denoted by the dummy variable p_t that distinguishes different traffic characteristics between weekdays and weekends.

(4) Historical statistic information

The daily trend of the traffic status can be captured by introducing historical statistic information into the prediction model. The historical average speed, median speed, maximum speed, minimum speed, and standard deviation at the t^{th} time slot of road segment l_i are defined as the average value, median value, maximum value, minimum value, and standard deviation in the training dataset, respectively, which are denoted by $v^i_{t,average}$, $v^i_{t,median}$, $v^i_{t,max}$, $v^i_{t,min}$ and d^i_t, respectively.

(5) Problem formulation

The task of traffic speed prediction is to use the previously observed speed records to forecast the future values of each road segment in a certain period. The multistep traffic speed problem can be formulated as

$$\widehat{\boldsymbol{V}}_{t+n} = \underset{\boldsymbol{V}_{t+n}}{\operatorname{argmax}} \ \Pr(\boldsymbol{V}_{t+n} | \boldsymbol{V}_t, \boldsymbol{V}_{t-1}, \cdots, \boldsymbol{V}_{t-m}; \mathcal{G}) \tag{1}$$

where $\widehat{\boldsymbol{V}}_{t+n}\,(n = 1, 2, 3, \cdots)$ represents the n^{th}-step predicted speed of the underlying road network, and $\{\boldsymbol{V}_t, \boldsymbol{V}_{t-1}, \cdots, \boldsymbol{V}_{t-m} | m = 1, 2, \cdots\}$ is the relevant previously observed value vector. $\Pr(\cdot|\cdot)$ is the conditional probability function.

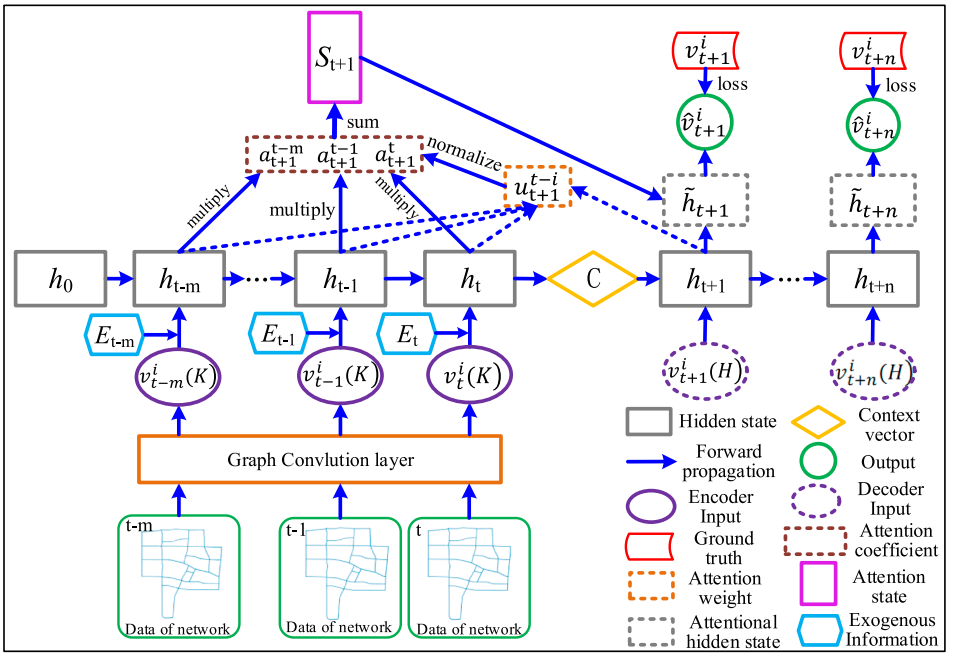


Fig. 2. Structure of the proposed model (taking $t + 1$ time step attention for example).

3.2. Attention graph convolutional sequence-to-sequence model (AGC-Seq2Seq)

In this subsection, we propose the novel AGC-Seq2Seq model that integrates spatio-temporal variables and exogenous information into the deep learning architecture for multistep traffic speed prediction.

The framework of the proposed AGC-Seq2Seq model is shown in Fig. 2. Specifically, the graph convolution operation is firstly utilized to capture the spatial characteristics based on the topology of underlying traffic network. Then the Seq2Seq model, which is composed of two connected RNN modules with independent parameters (Sutskever et al., 2014; Cho et al., 2014), encodes the spatially-fused time series as input to capture the spatio-temporal dependencies. And its decoder collaboratively produces the target multistep outputs organized by time steps from the context vector, and so it overcomes the disadvantage of fixed output timestamp of the RNN structure. We further adopt the attention mechanism to model the temporal heterogeneity of traffic pattern between sequences in encoder and decoder. The detailed contents of AGC-Seq2Seq model are as follows.

(1) Graph Convolution on Traffic Networks

Graph convolution extends the applicable scope of standard convolution from regular grids to general graphs by manipulating in the spectral domain. To introduce the general K-order graph convolution, we first define the K-hop neighborhoods for each road segment $l_i \in \mathcal{L}$ as $\mathcal{H}_i(K) = \{l_j \in \mathcal{L} | dis(l_i, l_j) \leq K\}$ in the context of road network topology (introduced in Section 3.1), where $dis(l_i, l_j)$ represents the minimum number of needed links among all the walks from l_i to l_j.

It is typical that the adjacency matrix is exactly the one-hop neighborhood matrix $\boldsymbol{A}$, and the K-hop neighborhood matrix can be acquired by calculating the K^{th} power of $\boldsymbol{A}$. To imitate the Laplacian matrix, we add the diagonal element to the neighborhood matrix, which is defined as

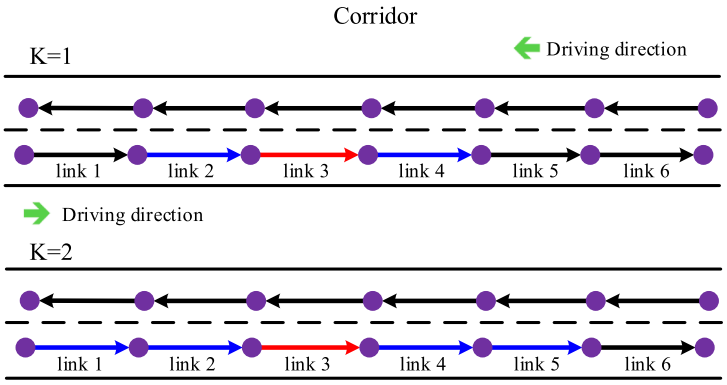
$$\boldsymbol{A}_{GC}^{K} = \text{Ci}[(\boldsymbol{A} + \boldsymbol{I})^K] \tag{2}$$

where $\text{Ci}(\cdot)$ is a clip function for the matrix by modifying each nonzero element to 1; thus, $\boldsymbol{A}_{GC}^{K}(i, j) = 1$ *for* $l_j \in \mathcal{H}_i(K)$ *or* $i = j$; otherwise, $\boldsymbol{A}_{GC}^{K}(i, j) = 0$. The identity matrix $\boldsymbol{I}$ added to $\boldsymbol{A}^K$ renders the convolution self-accessible in the topology graph.

Based on the abovementioned neighborhood matrix, a concise version of graph convolution (e.g., Cui et al., 2018) can be defined as follows.

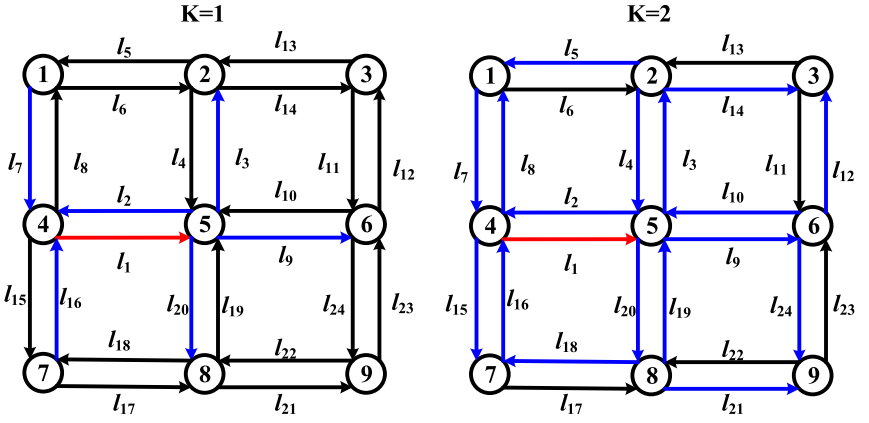
$$\boldsymbol{V}_t(K) = (\boldsymbol{W}_{GC} \odot \boldsymbol{A}_{GC}^{K}) \cdot \boldsymbol{V}_t \tag{3}$$

where $\boldsymbol{W}_{GC}$ is a trainable parameter matrix with the same size of $\boldsymbol{A}$. The operator $\odot$ refers to the Hadamard product that conducts the element-wise multiplication operation. Through the element-wise multiplication, $(\boldsymbol{W}_{GC} \odot \boldsymbol{A}_{GC}^{K})$ will produce a new matrix with trainable parameters on the K-hop neighbor positions and zero on the remaining positions. Therefore, $(\boldsymbol{W}_{GC} \odot \boldsymbol{A}_{GC}^{K}) \cdot \boldsymbol{V}_t$ can be understood as spatial discrete convolution for $\boldsymbol{V}_t$. In consequence, $\boldsymbol{V}_t(K)$ is the spatially-fused speed vector at the time t. Its i^{th} element $v_t^i(K)$ represents the spatially-fused speed of the road segment $l_i \in \mathcal{L}$ at the time t that incorporates the information of all the neighbor road segments in $\mathcal{H}_i(K)$.



Link ID	1	2	3	4	5	6
$A^1_{GC}[3]$	0	1	1	1	0	0
$A^2_{GC}[3]$	1	1	1	1	1	0

(a) Corridor



Link ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
$A^1_{GC}[1]$	1	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
$A^2_{GC}[1]$	1	1	1	1	1	0	1	1	1	1	0	1	0	1	1	1	0	1	1	1	1	0	0	1

(b) Urban road network

Fig. 3. Illustration of $\boldsymbol{A}^K_{GC}[i]$.

Further, Eq. (3) can be decomposed into a one-dimensional convolution that is flexible and suitable for parallel computing.

$$[v^i_{t-m}(K),\cdots,v^i_t(K)] = (\boldsymbol{W}_{GC}[i] \odot \boldsymbol{A}^K_{GC}[i])^T \cdot [\boldsymbol{V}_{t-m},\cdots,\boldsymbol{V}_t] \tag{4}$$

$\boldsymbol{W}_{GC}[i]$ and $\boldsymbol{A}^K_{GC}[i]$ are the i^{th} row of $\boldsymbol{W}_{GC}$ and $\boldsymbol{A}^K_{GC}$, respectively. An example of $\boldsymbol{A}^K_{GC}[i]$ on the road network is shown in Fig. 3, where the road segment i is in red line and neighbor links are in blue lines.

(2) Seq2Seq model

The encoder of Seq2Seq model takes the designed multidimensional feature vector as input, which concatenates the spatio-temporal variable $v^i_{t-j}(K)$ with exogenous variable $\boldsymbol{E}_{t-j}$ (including the information of time-of-day and weekday-or-weekend). The procedure above is demonstrated in the following equations.

$$v^i_{t-j}(K) = (\boldsymbol{W}_{GC}[i] \odot \boldsymbol{A}^K_{GC}[i])^T \cdot \boldsymbol{V}_{t-j}, \quad 0 \leqslant j \leqslant m \tag{5}$$

$$\boldsymbol{E}_{t-j} = [N_{t-j}; p_{t-j}] \tag{6}$$

$$\boldsymbol{X}_{t-j}^{i} = [v_{t-j}^{i}(K); \boldsymbol{E}_{t-j}] \tag{7}$$

where N_{t-j} and p_{t-j} are defined in Section 3.1; and the operator $[\cdot;\cdot]$ concatenates two tensors along the same dimensions

Then, in the encoder part, as demonstrated in Eqs. (8)–(9) below, at the time step $t-j$, $j \in \{0,\cdots,m\}$, the previous hidden status $\boldsymbol{h}_{t-j-1}$ is passed to the current time stamp together with input $\boldsymbol{X}_{t-j}$ to calculate $\boldsymbol{h}_{t-j}$. Therefore, the context vector $\boldsymbol{C}$ stores all the information of the encoder including the hidden states ($\boldsymbol{h}_{t-m}$, $\boldsymbol{h}_{t-m+1},\cdots,\boldsymbol{h}_{t-1}$) and input vector ($\boldsymbol{X}_{t-m}$, $\boldsymbol{X}_{t-m+1},\cdots,\boldsymbol{X}_{t}$), which is further designed as a connector between encoder and decoder parts.

$$\boldsymbol{h}_{t-j} = \begin{cases} \text{Cell}_{\text{encoder}}(\boldsymbol{h}_0, \boldsymbol{X}_{t-j}), & j = m \\ \text{Cell}_{\text{encoder}}(\boldsymbol{h}_{t-j-1}, \boldsymbol{X}_{t-j}), & j \in \{0,\cdots,m-1\} \end{cases} \tag{8}$$

$$\boldsymbol{C} = \boldsymbol{h}_t \tag{9}$$

where $\boldsymbol{h}_0$ is the initial hidden status and is typically set as a zero vector; $\text{Cell}_{\text{encoder}}(\cdot)$ is the calculation function for the encoder that is decided by the adopted RNN structure.

In the decoder part, the core idea is leveraging the context vector $\boldsymbol{C}$ as the initial hidden status, and subsequently decoding the output sequence step by step. In consequence, at the time stamp $t+j$, $j \in \{1,\cdots,n\}$, the hidden state $\boldsymbol{h}_{t+j}$ not only contains the input information, but also considers the previous output status ($\boldsymbol{h}_{t+1}$, $\boldsymbol{h}_{t+2},\cdots,\boldsymbol{h}_{t+j-1}$).

(3) Newly designed training method

The inputs of the decoder are dependent on the training method. *Teacher forcing* is a popular training strategy used in natural language processing. In the teacher-forcing training strategy, the ground truths (target sequence) are fed into the decoder for the training, and at the testing stage, the previously generated predictions are utilized as input for the later time stamp. However, this method is not suitable for the time-series problem primarily because of the discrepant distributions of the decoder inputs between the training and testing periods. Li et al. (2017) mitigated this issue by using *scheduled sampling* that randomly selects either the ground truth or the previous prediction to feed the model with the setting probability ϵ. However, it will inevitably increase the complexity of the model and calculation burden.

To overcome the issues above, we propose a new training method employing the historical statistic information and time-of-day as inputs. In the time-series prediction problem, historical statistic information can be obtained both in the training and testing stages; in this context, the distribution of decoder inputs between the training and testing periods will synchronize with each other, thus solving the dilemma of *teacher forcing*. Moreover, because historical statistic information is critical in multistep forecasting, adding it to the model is expected to enhance the prediction accuracy. Accordingly, the equations below are used to calculate the hidden state in the decoder at the time $t+j$, $j \in \{1, \cdots, n\}$.

$$\boldsymbol{v}_{t+j}^{i}(H) = [N_{t+j}; v_{t+j,average}^{i}; v_{t+j,median}^{i}; v_{t+j,max}^{i}; v_{t+j,min}^{i}; d_{t+j}^{i}] \tag{10}$$

$$\boldsymbol{h}_{t+j} = \begin{cases} \text{Cell}_{\text{decoder}}(\boldsymbol{C}, \boldsymbol{v}_{t+j}^{i}(H)), j = 1 \\ \text{Cell}_{\text{decoder}}(\boldsymbol{h}_{t+j-1}, \boldsymbol{v}_{t+j}^{i}(H)), j \in \{2,\cdots,n\} \end{cases} \tag{11}$$

where $\text{Cell}_{\text{decoder}}(\cdot)$ is the calculation function for the decoder, which is similar to that of the encoder.

(4) Adopted RNN structure

We employ the Gated Recurrent Unit (Chung et al., 2014) as the inner structure for both the encoder and decoder (shown in Fig. 4). It demonstrates competitive performance and a simpler structure than the standard LSTM. The calculation procedure of $\text{Cell}_{\text{encoder}}(\cdot)$ and $\text{Cell}_{\text{decoder}}(\cdot)$ is shown in Eqs. (12)–(17) below.

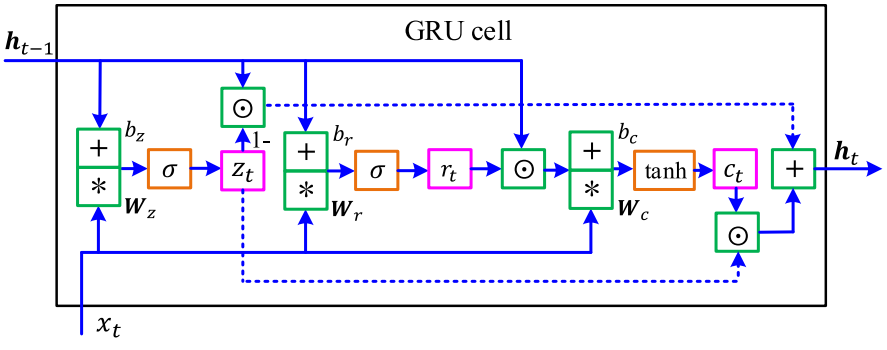


Fig. 4. Sketch of GRU.

$$z_t = \sigma(\boldsymbol{W}_z \cdot [\boldsymbol{h}_{t-1}; x_t] + b_z) \tag{12}$$

$$r_t = \sigma(\boldsymbol{W}_r \cdot [\boldsymbol{h}_{t-1}; x_t] + b_r) \tag{13}$$

$$c_t = \tanh(\boldsymbol{W}_c \cdot [r_t \odot \boldsymbol{h}_{t-1}; x_t] + b_c) \tag{14}$$

$$\boldsymbol{h}_t = (1 - z_t) \odot \boldsymbol{h}_{t-1} + z_t \odot \boldsymbol{c}_t \tag{15}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{16}$$

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{17}$$

In the equations above, z_t and r_t are the update gate and the reset gate, respectively. c_t is the candidate output. $\sigma(\cdot)$ and $\tanh(\cdot)$ are the two widely used nonlinear activation functions that map the input into (0,1) and $(-1,1)$, respectively. W_z, W_r, and W_c are the weight matrices that achieve the fully connected layer, while b_z, b_r, b_c are the corresponding bias vectors.

(5) Attention mechanism

To allow modeling of dependencies without regard to their distance in the input or output sequences, we further integrate the attention mechanism (Bahdanau et al., 2014; Luong et al., 2015) into the model. The key concept of the attention mechanism is adding the attention vector for each time step that captures the relevance of the source-side information to help predict the traffic speed. At the time step $t + j$, $j \in \{1, \cdots, n\}$, the attention function, defined by Equations (18)-(20), maps query $\boldsymbol{h}_{t+j}$ and a set of keys $(\boldsymbol{h}_{t-m}, \cdots, \boldsymbol{h}_{t-1}, \boldsymbol{h}_t)$ to the attention vector $\boldsymbol{S}_{t+j}$. As given by Eqs. (18)–(20) below, $\boldsymbol{S}_{t+j}$ is computed as a weighed sum of the keys, where the weight assigned to each key is obtained by a compatibility function of the query with the corresponding key.

$$u_{t+j}^{t-i} = \boldsymbol{q}^T \tanh(\boldsymbol{h}_{t+j} \boldsymbol{W}_f \boldsymbol{h}_{t-i}),\ i = 0,\ 1,\ \cdots, m \tag{18}$$

$$a_{t+j}^{t-i} = \mathrm{softmax}(u_{t+j}^{t-i}) = \frac{\exp(u_{t+j}^{t-i})}{\sum_{r=1}^{m} \exp(u_{t+j}^{t-r})},\ i = 0,\ 1,\ \cdots, m \tag{19}$$

$$\boldsymbol{S}_{t+j} = \sum_{i=1}^{m} a_{t+j}^{t-i} \boldsymbol{h}_{t-i} \tag{20}$$

where u_{t+j}^{t-i} can be used construed as to measure the similarity between $\boldsymbol{h}_{t+j}$ and $\boldsymbol{h}_{t-i}$, calculated by Eq. (18), and in this study, we employ the *Luong Attention* form (Luong et al., 2015) as the compatibility function with trainable weight matrix $\boldsymbol{W}_f$ and vector $\boldsymbol{q}^T$ to adjust the dimension of the result; a_{t+j}^{t-i} is the normalization of u_{t+j}^{t-i} and is further used as the weight coefficient with the corresponding encoder hidden state $\boldsymbol{h}_{t-i}$ to calculate $\boldsymbol{S}_{t+j}$.

As shown in Fig. 2, the attentional hidden state $\widetilde{\boldsymbol{h}}_{t+j}$ is composed of the attention vector $\boldsymbol{S}_{t+j}$ and original hidden state $\boldsymbol{h}_{t+j}$ through a simple concatenation, as shown in Eq. (21). Eq. (22) denotes the linear transformation from the hidden state to the output. The dimensions of the weighted parameter matrix $\boldsymbol{W}_v$ and intercept parameter b_v are consistent with the output.

$$\widetilde{\boldsymbol{h}}_{t+j} = \tanh(\boldsymbol{W}_h \cdot [\boldsymbol{S}_{t+j}; \boldsymbol{h}_{t+j}]) \tag{21}$$

$$\hat{v}_{t+j} = \boldsymbol{W}_v \widetilde{\boldsymbol{h}}_{t+j} + b_v \tag{22}$$

(6) Objective function

To jointly reduce the predictive errors in multiple step prediction, we define the loss as the mean absolute error between $(\hat{v}_{t+1}, \hat{v}_{t+2}, \cdots \hat{v}_{t+n})$ and $(v_{t+1}, v_{t+2}, \cdots v_{t+n})$, which is given by

$$loss = \frac{1}{n} \sum_{j=1}^{n} |\hat{v}_{t+j}^{i} - v_{t+j}^{i}| \tag{23}$$

All the parameters are updated by minimizing the loss function through the mini-batch gradient descent algorithm in the training stage. A detailed discussion regarding why the Seq2Seq framework is suitable for collaborative multistep prediction is presented in the Appendix A.

The training steps of the AGC-Seq2Seq model is illustrated in Algorithm 1.

Algorithm 1. AGC-Seq2Seq model Training

Input The central road segment i
The link adjacency matrix $\boldsymbol{A}$ of the underlying traffic network
Observations of traffic speed $\{\boldsymbol{V}_1, \cdots, \boldsymbol{V}_T\}$ in training set
Observations of time-of-day $\{N_1, \cdots, N_T\}$ in training set
Observations of weekday-or-weekend $\{P_1, \cdots, P_T\}$ in training set
Observations of historical statistic information $\{v^i_{1,average}, \cdots, v^i_{T,average}\}$, $\{v^i_{1,median}, \cdots, v^i_{T,median}\}$, $\{v^i_{1,max}, \cdots, v^i_{T,max}\}$, $\{v^i_{1,min}, \cdots, v^i_{T,min}\}$, $\{d^i_1, \cdots, d^i_T\}$ in training set
Input time step: $m + 1$
Prediction time step: n
Output AGC-Seq2Seq model with learnt parameters
Procedure AGC-Seq2Seq model Train
1: Initialize a null set: $\boldsymbol{X} \leftarrow \varnothing$
2: **for** all available time intervals t $(1 \le t \le \mathrm{T})$ **do**
3: $\mathcal{X}_{\text{spatial}} \leftarrow [\boldsymbol{V}_{t-m}, \cdots, \boldsymbol{V}_{t-1}, \boldsymbol{V}_t]$
4: $\mathcal{X}_{\text{exogenous}} \leftarrow [\boldsymbol{E}_{t-m}, \cdots, \boldsymbol{E}_{t-1}, \boldsymbol{E}_t]$, where $\boldsymbol{E}_{t-j} = (N_{t-j}, p_{t-j})$
5: $\mathcal{X}_{\text{decoder}} \leftarrow [\boldsymbol{v}^i_{t+1}(H), \cdots, \boldsymbol{v}^i_{t+n}(H)]$, where $\boldsymbol{v}^i_{t+j}(H) = (N_{t+j}, v^i_{t+j,max}, d^i_{t+j}, v^i_{t+j,min}, v^i_{t+j,average}, v^i_{t+j,median})$ ► where $\mathcal{X}_{\text{spatial}}$ is the input set of graph convolution, $\mathcal{X}_{\text{exogenous}}$ is the input set of exogenous information for encoder, $\mathcal{X}_{\text{decoder}}$ is the input set of decoder
6: A training sample $(\mathcal{X}_{\text{spatial}}, \mathcal{X}_{\text{exogenous}}, \mathcal{X}_{\text{decoder}})$ is put into $\boldsymbol{X}$
7: **end for**
10: Initialize the initial hidden status of encoder, all the weight and bias parameters
11: **repeat**
12: Randomly extract a batch of samples $\boldsymbol{X}^b$ from $\boldsymbol{X}$
13: Update the parameters by minimizing the objective function shown in Eq. (23) through the mini-batch gradient descent algorithm within $\boldsymbol{X}^b$
14: **until** convergence criterion met
15: **end procedure**

4. Numerical examples

4.1. Dataset

The datasets utilized in this study were collected from the users of A-map, which is a smartphone-based navigation application with the most active users in China (Sohu, 2018). The studied sites contain freeway network and urban road network.

(a) Freeway network

The freeway network is selected as the entire 2nd ring road, which is the most congested among the ring roads in Beijing (The position is shown in Fig. 5(a)). As shown in Fig. 5(b), we partition the 33KM-in-length 2nd ring road into 163 road segments with 200 m in length. Furthermore, we calculate the 5-min average speed for each link using the collected trajectory points of anonymous users. The plots of the traffic speed in the 2nd ring road on weekdays and weekends are shown in Fig. 5(c) and (d) with the x-axis for the longitude, y-axis for the latitude, z-axis for the time and color map for speed.

(b) Urban road network

We choose an area with the size of 10 km^2 in Tianjin Nankai district as the case study of urban road network, shown in Fig. 6 (a). Fig. 6(b) shows the topology of the network consisting of 242 links, and the detailed information is presented in Appendix B. It should be noted that the road segments between two intersections in the arterial road are partitioned into several links by local streets which leading to the nearby point of interest (e.g. residential area, shopping mall, etc.). We calculate the 5-min average speed for each link through trajectory data after filtering the outliers.

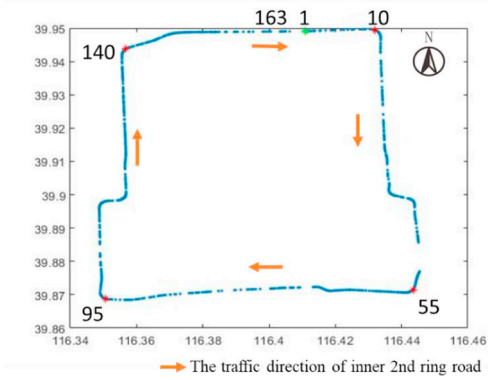
The detailed experimental descriptions of two datasets are summarized in Table 1.

4.2. Model comparisons

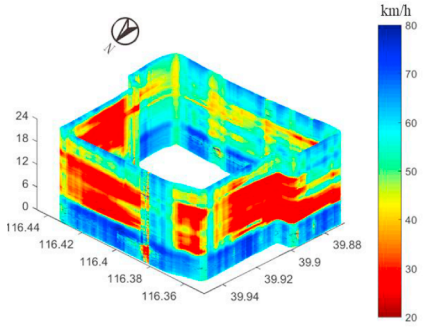
In this subsection, the proposed model is compared with other benchmark models, including the traditional time-series analysis approaches (i.e., HA and ARIMA) and state-of-the-art machine learning (i.e., ANN, KNN, SVR, and XGBOOST) /deep learning models (i.e., LSTM, GCN, and Seq2Seq-Att).

- HA: The historical average model predicts the future speed in the testing dataset based on the empirical statistics in the training set. For example, the average speed during 8:00–8:05 of road segment $l_i \in \mathcal{L}$ is estimated by the mean of all historical speeds in the training dataset during 8:00–8:05 of the same link. The calculation formula is given by

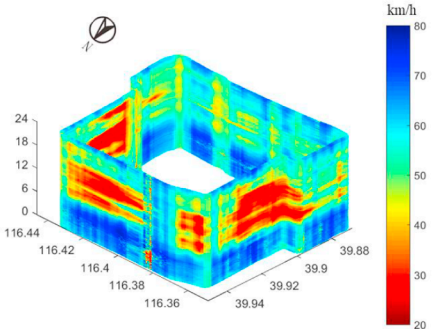
(a) Position of 2nd ring road



(b) Sketch of road segments in GIS map



(c) Traffic speed of 2nd ring road on weekdays in July 2016



(d)Traffic speed of 2nd ring road on weekends in July 2016

Fig. 5. Results of freeway data preprocess.

$$v^i_{t,average} = \frac{1}{\mathcal{D}} \sum_{r=1}^{\mathcal{D}} v^i_{t,r} \tag{24}$$

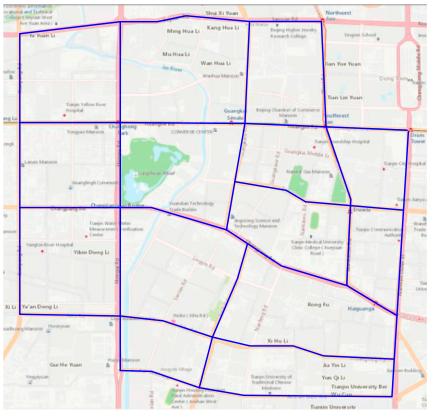
where $v^i_{t,r}$ is the speed at the tth time slot of road segment l_i on rth day; and $\mathcal{D}$ is the number of days in the training set.

- ARIMA: For the autoregressive integrated moving average (p, d, q) model (Box and Pierce, 1970), the degree of differencing is set as $d = 1$, and the order of autoregressive part and moving average part(p, q) are determined through computing the corresponding Akaike information criterion of the training dataset with $p \in [7, 12]$, $q \in [0, 2]$. The calculation formula is given by

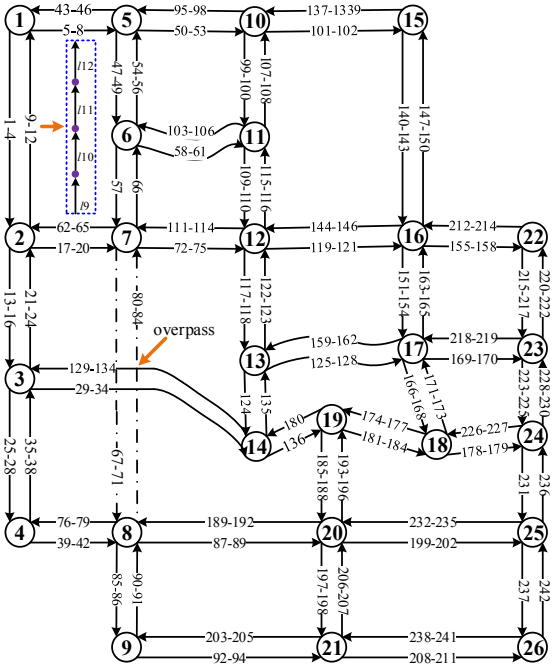
$$\left(1 - \sum_j^p \phi_j L^j\right)(1 - L)^d v^i_t = \left(1 + \sum_j^q \theta_j L^j\right) \epsilon_t \tag{25}$$

where L is the lag operator and ϵ_t is the white noise sequence.

- ANN: We establish a three-layer artificial neural network (Rumelhart et al., 1988) activated by the sigmoid function, and set the number of hidden neurons twice the dimension of the feature vector. Because the ANN does not differentiate variables across time steps, it fails to capture the temporal dependencies.
- KNN: K-nearest neighbor (Denoeux, 1995) is a lazy learning algorithm that obtains the K-most similar observations in the training set through the Euclidean distance between feature vectors. The predicted value is calculated through the weighted summation of the corresponding future values belonging to the selected observations. The hyper parameter K is chosen through cross validation from 5 to 25.
- SVR: In support vector regression (Suykens and Vandewalle, 1999), the fitting curve is calculated through the mapping feature vectors into the high-dimensional space aided by the kernel function. The kernel function and hyper parameters in the model are selected through cross validation.
- XGBOOST: XGBOOST (Chen and Guestrin, 2016) yields outstanding performance in a broad range of machine learning tasks; it is a scalable end-to-end boosting system based on the tree structure. All the features are reshaped to a vector and fed into XGBOOST for training.



(a) Map of urban road network



(b) Topology of road network

Fig. 6. Information of urban road network.

Table 1
The experimental descriptions of two datasets.

Item\Type	Freeway network	Urban road network
Date of training	Oct 1, 2016-Nov 20 2016	Aug 1, 2018-Sep 23 2018
Date of testing	Nov 21, 2016-Nov 27 2016	Sep 24, 2018- Sep 30 2018
Prediction time horizon	06:00–22:00	07:00–19:00
Size of dataset	58 × 192 × 163 ≈ 1.8million	61 × 144 × 242 ≈ 2.1million

- LSTM: In LSTM (Hochreiter and Schmidhuber, 1997), all the features of each road segment are reshaped to a matrix with one axis as the time steps, and the other axis as the feature category. LSTM takes temporal dependencies into account, but does not capture spatial dependencies.
- GCN: In GCN, the features of all road segments in the underlying traffic network are reshaped to a matrix with one axis as each road segment, and the other axis as the feature category. GCN generalizes the convolution to non-Euclidean domains by the Laplacian matrix of the graph; therefore, it considers spatial correlations, but does not capture temporal dependencies.
- Seq2Seq-Att: In Seq2Seq-Att, the attention mechanism based on the Seq2Seq structure is utilized for traffic prediction along with

Table 2
Illustration of feature vector.

No.	Notation	No.	Notation	No.	Notation
f0	N_{t+n}	f7	$v^i_{t-11}(\boldsymbol{v}_{t-11})$	f13	$v^i_{t-5}(\boldsymbol{v}_{t-5})$
f1	$v^i_{t+n,average}$	f8	$v^i_{t-10}(\boldsymbol{v}_{t-10})$	f14	$v^i_{t-4}(\boldsymbol{v}_{t-4})$
f2	$v^i_{t+n,median}$	f9	$v^i_{t-9}(\boldsymbol{v}_{t-9})$	f15	$v^i_{t-3}(\boldsymbol{v}_{t-3})$
f3	d^i_{t+n}	f10	$v^i_{t-8}(\boldsymbol{v}_{t-8})$	f16	$v^i_{t-2}(\boldsymbol{v}_{t-2})$
f4	$v^i_{t+n,max}$	f11	$v^i_{t-7}(\boldsymbol{v}_{t-7})$	f17	$v^i_{t-1}(\boldsymbol{v}_{t-1})$
f5	$v^i_{t+n,min}$	f12	$v^i_{t-6}(\boldsymbol{v}_{t-6})$	f18	$v^i_t(\boldsymbol{v}_t)$
f6	p_{t+n}				

the new proposed training method. The only difference between the Seq2Seq-Att and AGC-Seq2Seq models is the graph convolution layer.

To ensure fairness, we choose the most appropriate inputs for the aforementioned benchmark models. The traditional time-series models (HA and ARIMA) utilize the whole time-series of speed records in the training set based on the corresponding formulas. The machine learning models (ANN, XGB, KNN, SVR) take the designed features as input which has the same feature category and look-back time windows as those of the AGC-Seq2Seq model, while GCN extra takes the spatial speed observations into account. We consider the look-back time windows as 12 (i.e., $m = 11$), implying that the speed records in the past hour are adopted to predict the future value. The designed vector containing speed observations in the past hour, time-of-day, weekday-or-weekend, and historical statistic information is shown in Table 2 (the spatial speed observations for GCN is in the brackets). As for the temporal deep learning approaches, LSTM utilizes the series $\{v^i_{t-11}, v^i_{t-10}, \cdots, v^i_t\}$, while Seq2Seq-Att additionally exploits $\{\boldsymbol{v}^i_{t+1}(H), \cdots, \boldsymbol{v}^i_{t+n}(H)\}$ for decoder.

All the notations are defined in Section 3.1. n is fixed according to the prediction step.

In this experiment, the structure of AGC-Seq2Seq model is comprised of one bidirectional GRU layer with 20 hidden units for encoder and one unidirectional GRU layer with 20 hidden units for decoder. The graph convolution layer consists of two filters to fully catch up spatial information. The order of graph convolution is set as 1. The initial learning rate is 1e-2, and decays with rate of 0.9 after every 50 training steps. Early stopping on the validation dataset is used to avoid overfitting. All the benchmarks are under fine-tuned. Before model training and validation, the speed observations, time-of-day, and historical information are standardized to the range [0, 1], through the max-min standardization, respectively.

We evaluate the models via three classical error indexes: mean absolute percentage error (MAPE), mean absolute error (MAE), and root mean squared error (RMSE), given by $\text{MAPE} = \frac{1}{Q}\sum_{i=1}^{Q}\frac{|v_i - \hat{v}_i|}{v_i}$, $\text{MAE} = \frac{1}{Q}\sum_{i=1}^{Q}|v_i - \hat{v}_i|$, and $\text{RMSE} = \sqrt{\frac{1}{Q}\sum_{i=1}^{Q}(v_i - \hat{v}_i)^2}$, where v_i and $\hat{v}_i$ are the i^{th} ground truth and prediction value of the traffic speed, respectively; Q is the size of the testing dataset.

Our experimental platform is on the server with two CPUs (Intel(R) Xeon(R) CPU E5-2673 v3 @2.40Ghz, 24 cores), 256-GB RAM, and four GPUs (NVIDIA Quadro P5000, 16 GB memory). All the algorithms are coded in the parallel computation structure.

Tables 3 and 4 show the comparisons of the proposed model and benchmark algorithms for 5 min, 15 min, and 30 min ahead forecasting on the testing datasets of freeway network and urban road network respectively. The prediction performances of congested western 2nd ring road (link 100-link 130) and 46 congested links of urban road network (speed in peak hours under 15 km/h) are also added in the brackets. The following phenomena can be observed from the experimental results.

i. AGC-Seq2Seq model outperforms the other benchmarks in terms of all the metrics under all prediction intervals.
ii. The performance of HA is invariant to the increases in the forecasting horizon because it depends only on the historical data.
iii. The performances of all the models under the 5-min forecasting horizons are similar because the traffic status is relatively stable within 5 min.
iv. The deep-learning approaches yield better predictive performances but longer computational time than the traditional machine-learning models.
v. The GCN (which models spatial correlations) outperforms LSTM (which captures the temporal characteristics), providing verification that the consideration of spatial correlations is important in traffic speed forecasting.
vi. The AGC-Seq2Seq model exhibits a distinct improvement over the GCN and Seq2Seq-Att; this emphasizes the importance of capturing the spatio-temporal characteristics simultaneously for the traffic speed forecasting. The running time of the AGC-Seq2Seq model is only slightly higher than those of the GCN and Seq2Seq-Att; it primarily benefits from the advanced parallel computation technology in the GPU module, which owns thousands of kernels and skills in doing a variety of intensive calculations.
vii. The aggregated results for each model across all links are relatively close while the difference on congested links is obvious, which is consistent with our general knowledge. The previous research also holds the same view (Fusco et al., 2016; Ermagun and Levinson, 2018c).

To investigate the effect of attention mechanism, we test the Seq2Seq-GC model on urban road network which keeps the same structure with AGC-Seq2Seq except for dropping the attention mechanism. Table 5 shows the results. AGC-Seq2Seq consistently

Table 3
Prediction performance comparison on freeway network.

Model	MAPE(congested)	MAE(congested)	RMSE(congested)	Time(s)[b]
(a) 5-min prediction horizon (one step)				
HA	30.32% (46.06%)	7.89 (10.02)	10.39 (12.8)	/
ARIMA	10.65% (13.49%)	3.64 (3.78)	5.40 (5.77)	686
ANN	10.69% (13.56%)	3.54 (3.67)	5.17 (5.47)	71
XGBOOST	10.38% (13.12%)	3.41 (3.52)	5.02 (5.26)	58
KNN	12.26% (15.33%)	3.88 (4.07)	5.84 (6.23)	50
SVR	11.54% (15.28%)	3.74 (3.93)	5.26 (5.60)	127
LSTM	10.25% (13.95%)	3.48 (3.77)	5.23 (5.59)	219
GCN[a]	10.06% (12.38%)	3.39 (3.44)	5.01 (5.18)	263
Seq2Seq-Att	10.10% (12.65%)	3.40 (3.52)	5.11 (5.43)	223
AGC-Seq2Seq[a]	**9.57% (11.36%)**	**3.25 (3.28)**	**4.85 (4.96)**	280
(b) 15-min prediction horizon (three steps)				
Model	*MAPE(congested)*	*MAE(congested)*	*RMSE(congested)*	*Time(s)[b]*
HA	30.32% (46.06%)	7.89 (10.02)	10.39 (12.76)	/
ARIMA	16.71% (22.20%)	5.31 (5.98)	8.25 (9.63)	698
ANN	16.45% (22.05%)	4.94 (5.62)	7.45 (8.43)	72
XGBOOST	16.07% (21.29%)	4.82 (5.40)	7.34 (8.13)	62
KNN	16.83% (21.72%)	5.01 (5.61)	7.66 (8.61)	53
SVR	16.99% (23.11%)	5.14 (5.83)	7.61 (8.74)	125
LSTM	16.17% (21.52%)	5.01 (5.74)	7.99 (9.40)	279
GCN[a]	14.99% (19.30%)	4.62 (5.11)	7.32 (8.23)	363
Seq2Seq-Att	15.00% (19.43%)	4.62 (4.89)	7.38 (8.38)	350
AGC-Seq2Seq[a]	**14.46% (17.86%)**	**4.47 (4.59)**	**7.12 (7.78)**	390
(c) 30-min prediction horizon (six steps)				
Model	*MAPE(congested)*	*MAE(congested)*	*RMSE(congested)*	*Time(s)[b]*
HA	30.32% (46.06%)	7.89 (10.02)	10.39 (12.76)	/
ARIMA	22.82% (31.50%)	6.99 (8.26)	10.6 (12.98)	701
ANN	20.55% (28.25%)	5.91 (6.94)	8.67 (10.02)	72
XGBOOST	20.98% (26.94%)	5.78 (6.68)	8.68 (9.65)	63
KNN	20.05% (26.31%)	5.79 (6.68)	8.71 (9.94)	55
SVR	21.02% (28.55%)	6.08 (7.01)	8.86 (10.21)	157
LSTM	20.70% (27.81%)	6.40 (7.01)	10.03 (12.01)	287
GCN[a]	18.64% (23.78%)	5.54 (6.37)	8.81 (10.11)	510
Seq2Seq-Att	18.40% (23.63%)	5.36 (6.47)	8.64 (9.93)	520
AGC-Seq2Seq[a]	**17.94% (21.10%)**	**5.25 (5.61)**	**8.36 (9.03)**	600

outperforms Seq2Seq-GC which intuitively validates the effect of attention mechanism in traffic prediction.

Fig. 7 shows the multistep prediction values of XGBOOST and AGC-Seq2Seq in the peak hours under the 30-min and 15-min horizons respectively. It is obvious that the XGBOOST model learns a false pattern in such a sharp falling period with a distinct time lag, shown by green arrows. Hence its prediction values lag behind the ground truth and cause non-negligible errors. Meanwhile, AGC-Seq2Seq alleviates this problem by capturing the accurate trend when the traffic status oscillates seriously.

We select ARIMA, XGBOOST, LSTM and GCN as the representatives for the time-series models, machine learning models, and deep learning approaches, respectively, to compare their performances with AGC-Seq2Seq under the 5–30-min prediction intervals, as shown in Fig. 8. AGC-Seq2Seq tends to demonstrate better performance than other models with the increase in the prediction horizon. Additionally, the ARIMA model performs the worst because of the step-by-step error accumulation in the multistep forecasting scenario.

Figs. 9 and 10 further show the prediction metrics of selected for each link and time-of-day on freeway network, respectively. Obviously, AGC-Seq2Seq performances better than other models for each link and time point. The results also demonstrate that the difference between each model in peak hours and congested links is more distinct (shown in local enlarged drawings), while it is relatively close in nom-peak hours and non-congested links.

4.3. Model interpretation

(1) Feature importance

Fig. 11 shows the F score (the number of times a feature is used to split the data across all trees, and a higher score indicates the corresponding feature being more import) of the feature vector (shown in Table 2) in the XGBOOST model under 15-min prediction horizon, which is used widely to assess the importance of the features (Ke et al., 2017). To evaluate the trend of feature importance under different prediction intervals, we divide the feature vector into two major categories: (1) speed records T1 in the past hour (f7–f18) and exogenous information T2 (f0–f6). The F score of feature fi, $i = 0, 1, \cdots, 18$ is denoted as $F(fi)$. The relative importance of T1 and T2 can be calculated as $\frac{\sum_{i=7}^{18} F(fi)}{\sum_{i=0}^{18} F(fi)}$ and $\frac{\sum_{i=0}^{6} F(fi)}{\sum_{i=0}^{18} F(fi)}$, respectively. Fig. 12 shows the relative importance of T1 and T2 under

Table 4
Prediction performance comparison on urban road network.

Model	MAPE(congested)	MAE(congested)	RMSE(congested)	Time(s)[b]
(a) 5-min prediction horizon (one step)				
HA	24.49% (48.25%)	4.08 (6.63)	5.27 (8.27)	/
ARIMA	15.37% (21.92%)	3.01 (4.03)	3.99 (5.26)	617
ANN	16.15% (24.59%)	3.04 (4.18)	4.02 (5.39)	170
XGBOOST	16.38% (26.18%)	3.03 (4.22)	4.02 (5.43)	72
KNN	20.15% (34.25%)	3.57 (5.18)	4.78 (6.78)	125
SVR	17.19% (25.33%)	3.09 (4.21)	4.01 (5.33)	89
LSTM	15.38% (22.25%)	2.98 (4.04)	3.96 (5.24)	306
GCN[a]	15.35% (23.47%)	2.99 (4 1 1)	3.99 (5.36)	342
Seq2Seq-Att	15.35% (23.28%)	2.99 (4.11)	4.02 (5.39)	318
AGC-Seq2Seq[a]	**15.09% (21.72%)**	**2.93 (3.95)**	**3.91 (5.15)**	365
(b) 15-min prediction horizon (three steps)				
Model	*MAPE(congested)*	*MAE(congested)*	*RMSE(congested)*	*Time(s)*[b]
HA	24.49% (48.25%)	4.08 (6.63)	5.27 (8.27)	/
ARIMA	21.76% (36.30%)	3.92 (5.68)	5.21 (7.38)	640
ANN	21.83% (39.96%)	3.83 (5.76)	5.12 (7.53)	176
XGBOOST	21.83% (38.66%)	3.79 (5.70)	5.00 (7.32)	76
KNN	22.83% (41.48%)	3.94 (6.01)	5.28 (7.89)	130
SVR	22.76% (39.53%)	3.85 (5.76)	5.07 (7.39)	91
LSTM	21.76% (37.10%)	3.84 (5.69)	5.11 (7.36)	362
GCN[a]	20.93% (38.86%)	3.74 (5.62)	5.05 (7.39)	472
Seq2Seq-Att	21.13% (37.28%)	3.77 (5.67)	5.12 (7.53)	460
AGC-Seq2Seq[a]	**19.92% (33.82%)**	**3.62 (5.32)**	**4.87 (6.96)**	507
(c) 30-min prediction horizon (six steps)				
Model	*MAPE(congested)*	*MAE(congested)*	*RMSE(congested)*	*Time(s)*[b]
HA	24.49% (48.26%)	4.08 (6.63)	5.27 (8.27)	/
ARIMA	24.47% (42.13%)	4.21 (6.33)	5.64 (8.33)	665
ANN	23.10% (42.93%)	3.95 (6.11)	5.30 (8.05)	174
XGBOOST	23.24% (43.43%)	3.93 (6.12)	5.21 (7.91)	78
KNN	23.60% (44.36%)	4.01 (6.26)	5.39 (8.26)	180
SVR	24.10% (44.38%)	3.99 (6.17)	5.27 (8.03)	106
LSTM	23.33% (43.61%)	3.98 (6.21)	5.40 (8.25)	373
GCN[a]	22.37% (41.76%)	3.89 (6.05)	5.27 (8.03)	668
Seq2Seq-Att	22.50% (41.82%)	3.91 (6.08)	5.34 (8.19)	676
AGC-Seq2Seq[a]	**21.38% (38.18%)**	**3.74 (5.65)**	**5.07 (7.50)**	760

[a] The order of graph convolution in this experiment is set as $K = 1$. The high-order situation is discussed in Section 4.3.
[b] The computational time in this table includes training datasets loading, hyper-parameters optimization, model training and model prediction. It mainly serves as an indicator to measure the complexity of each model. We note that in the industrial-scale applications, pre-training method and cloud computing technology can greatly shorten the computational time.

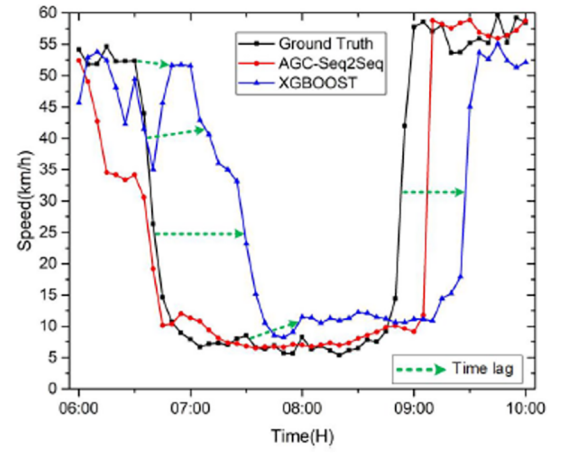
Table 5
Comparison for AGC-Seq2Seq and Seq2Seq-GC on the urban network.

Interval	Model	MAPE	MAE	RMSE
5-min	Seq2Seq-GC	15.50% (24.37%)	2.98 (4.11)	4.00 (5.39)
	AGC-Seq2Seq	**15.09% (21.72%)**	**2.93 (3.95)**	**3.91 (5.15)**
15-min	Seq2Seq-GC	20.56% (35.18%)	3.69 (5.46)	4.96 (7.13)
	AGC-Seq2Seq	**19.92% (33.82%)**	**3.62 (5.32)**	**4.87 (6.96)**
30-min	Seq2Seq-GC	21.97% (40.56%)	3.79 (5.80)	5.12 (7.69)
	AGC-Seq2Seq	**21.38% (39.31%)**	**3.74 (5.65)**	**5.07 (7.50)**

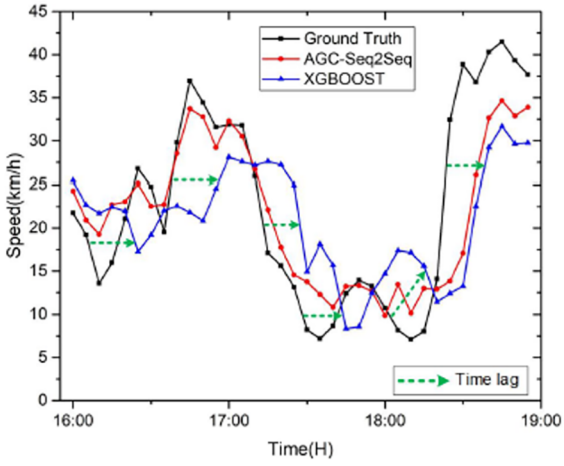
different prediction intervals. The results indicate that the exogenous variables are more important in the long-term prediction than in the short-term prediction. This is because the value of look-back observations degrades gradually with the increase in the forecasting interval. For this reason, it is necessary to design the new training method for Seq2Seq architecture in the case of multistep traffic prediction, which incorporates the exogenous information as input for decoder.

(2) Effect of spatial features in multistep prediction

Fig. 13 shows the curves of prediction errors varying with K-hop neighbors in the AGC-Seq2Seq model under 5-min, 15-min and 30-min time intervals on the freeway network and urban road network, respectively ($k = 0$ represents the case of Seq2Seq-Att model). The best k under 5-min, 15-min and 30-min is 5, 6 and 7 respectively for freeway network and 1, 2 and 3 for urban road network, implying that the spatial correlation of traffic condition becomes more substantial as the forecasting horizon increases. The best k on



(a) 30-min prediction of morning peak hours on freeway link 106 (Nov 21st 2016)



(b) 15-min prediction of evening peak hours on urban link 231 (Sep 26th 2018)

Fig. 7. Multistep prediction values of XGBOOST and AGC-Seq2Seq on peak hours.

the urban network is obviously smaller than that on the freeway network mainly due to the characteristic of interrupted traffic flow and the influence of traffic lights.

(3) Interpretation of graph convolution weights

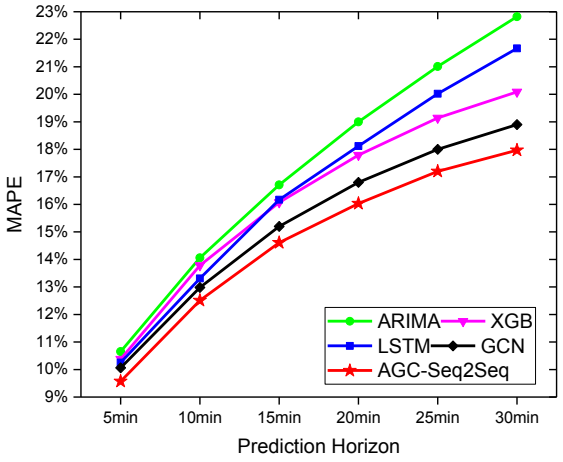
To better demonstrate the AGC-Seq2Seq model's capability of revealing spatial dependencies, we analyze the trained graph convolution weights of freeway network and urban road network, respectively.

(a) Freeway network

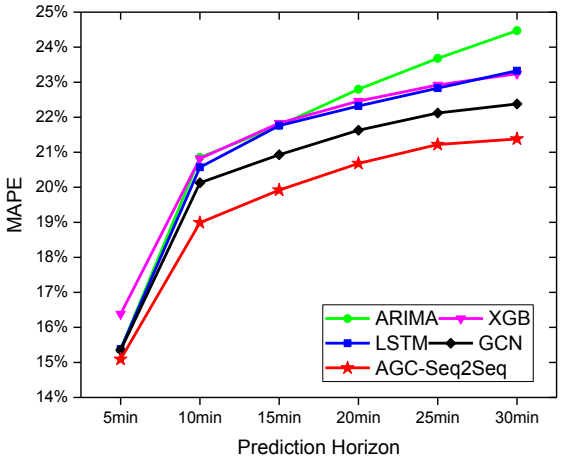
We set the topology of freeway network at different locations as target links, upstream links and downstream links for different order neighbors and compare the graph convolution weights of them. The locations with higher absolute values of graph convolution weights contribute more to the spatially-fused speed $v_i^i(K)$ (according to the Eq. (4)). Thus the absolute value of graph convolution weights can measure the relative spatial importance (Cui et al., 2018; Wang et al., 2018). The absolute values of graph convolution weights for r^{th} order upstream/downstream neighbor links are the sum of weights on corresponding links on the underlying traffic networks and the relative spatial importance of the r^{th} order upstream/downstream neighbor links is defined as

$$R_{spatial}^{\{r,\mu\}} = \sum_i \frac{\| \widetilde{\boldsymbol{W}_{GC}^K}[i]\{r,\mu\}\|_1}{\| \widetilde{\boldsymbol{W}_{GC}^K}[i]\|_1} \quad r \le K, \mu = up \quad or \quad down \tag{26}$$

$$\widetilde{W_{GC}^K} = \boldsymbol{W}_{GC} \odot \boldsymbol{A}_{GC}^K \tag{27}$$



(a) Performance on freeway network



(b) Performance on urban road network

Fig. 8. Performance of selected models with varying prediction intervals.

where $\widetilde{\boldsymbol{W}}_{GC}^{K}$ denotes the parameter matrix of K-hop graph convolution after training stage; $\widetilde{\boldsymbol{W}}_{GC}^{K}[i]$ denotes the i^{th} row of $\widetilde{\boldsymbol{W}}_{GC}^{K}$; $\widetilde{\boldsymbol{W}}_{GC}^{K}[i]\{r, \mu\}$ denotes the vector representing the values of the r^{th} order upstream/downstream neighbor links of link i in the matrix $\widetilde{\boldsymbol{W}}_{GC}^{K}$ and $||\cdot||_1$ represents the L_1 norm.

The relative spatial importance of graph convolution weight for target link $l_i \in \mathcal{L}$ is defined as

$$R_{spatial}^{t} = \sum_i \frac{|\widetilde{\boldsymbol{W}}_{GC}^{K}(i,i)|}{||\widetilde{\boldsymbol{W}}_{GC}^{K}[i]||_1} \tag{28}$$

where $\widetilde{\boldsymbol{W}}_{GC}^{K}(i,i)$ denotes the diagonal element of $\widetilde{\boldsymbol{W}}_{GC}^{K}$.

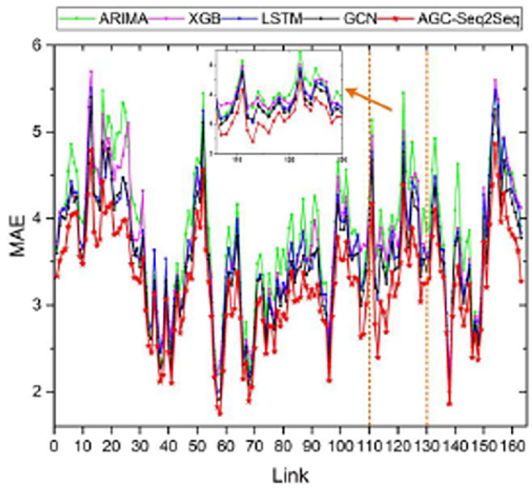
Based on Eqs. (26)–(28), the histogram of relative spatial importance under 15-min prediction interval and $K = 3$ of graph convolution is shown as Fig. 14.

1st/2nd/3rd up (down) denotes the 1st\2nd\3rd order neighbor upstream (downstream) links

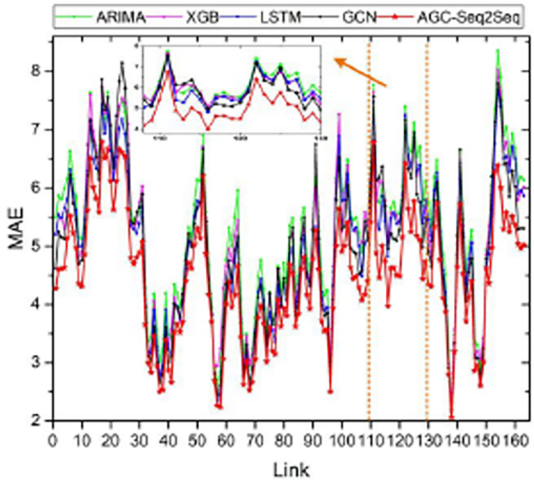
The following observations can be drawn from the results:

i. The target link itself plays the most important role in spatial dependencies because the traffic evolution of the target link is more correlated with itself than other links.
ii. The relative importance of downstream links is higher than that of upstream links. Since the traffic congestion propagates from downstream to upstream along the traffic network, the downstream links are expected to exert higher impacts.
iii. The spatial importance reduces as the order of neighbor increases because the spatial importance decrease with physical distance.

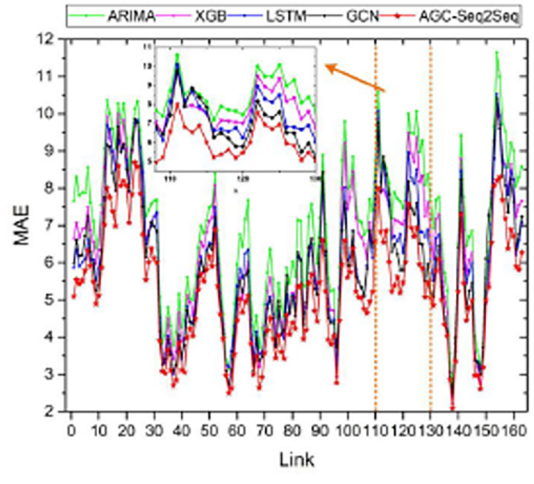
(b) Urban road network



(a) 5-min prediction horizon (one step)

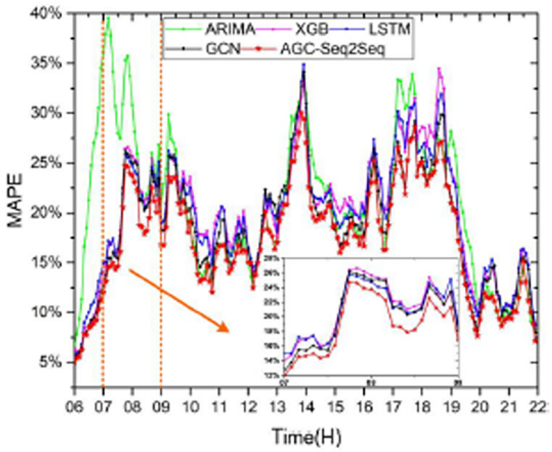


(b) 15-min prediction horizon (three steps)

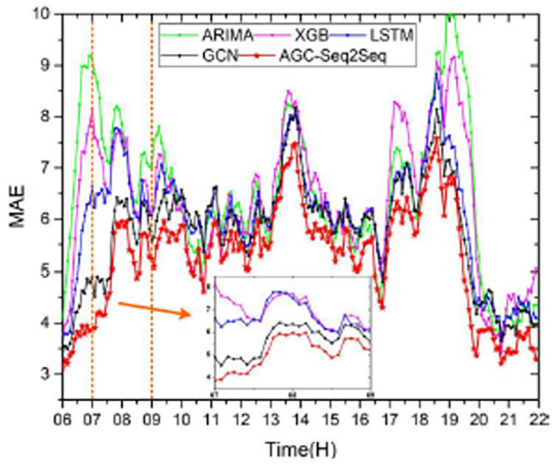


(c) 30-min prediction horizon (six steps)

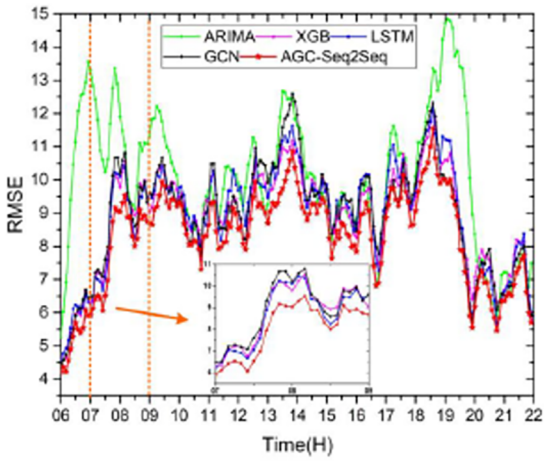
Fig. 9. Prediction performance for each link on freeway network.



(a) MAPE by time-of-day under 30-min prediction



(b) MAE by time-of-day under 30-min prediction



(c) RMSE by time-of-day under 30-min prediction

Fig. 10. Performance metrics by time-of-day under 30-min on freeway network.

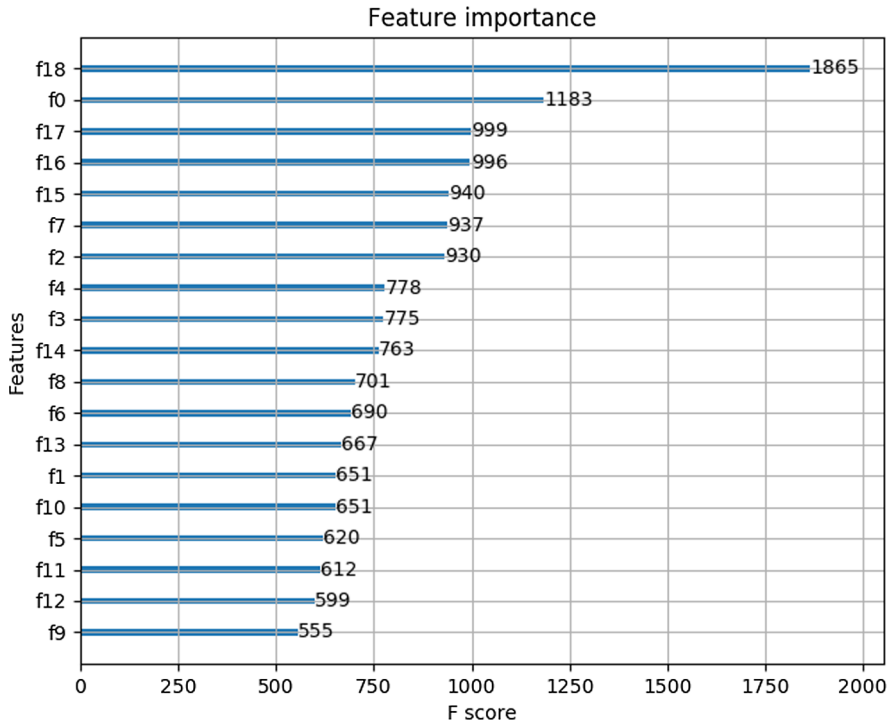


Fig. 11. F score of feature vector in XGBOOST model under 15-min horizon.

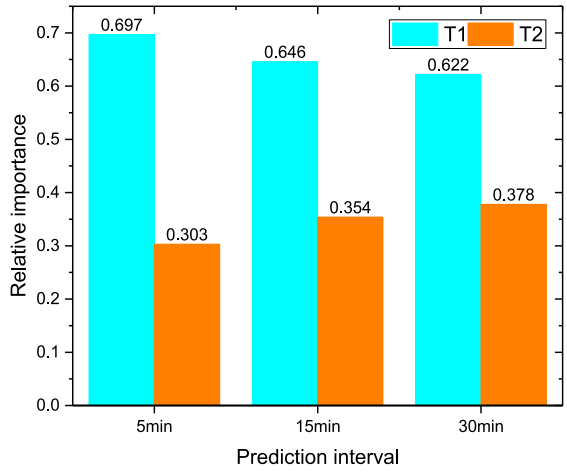
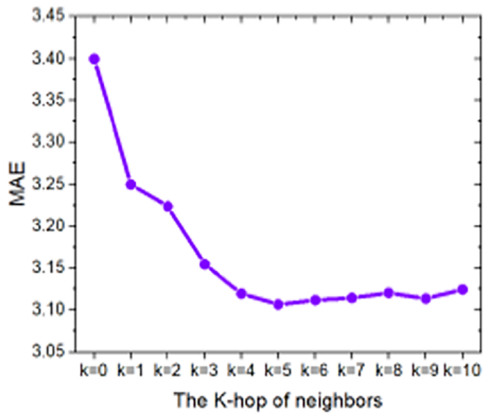


Fig. 12. Relative importance of feature under different prediction intervals.

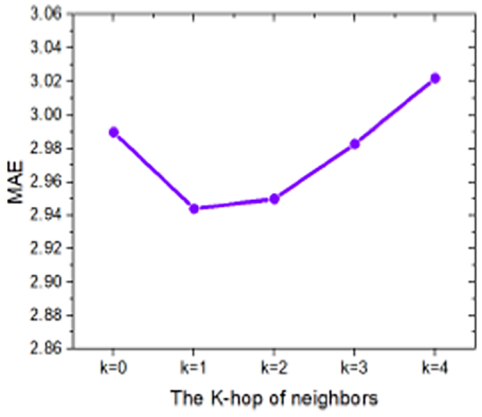
We visualize the graph convolution weights for each link $l_i \in \mathcal{L}$ (define as $\sum_j |\widetilde{\boldsymbol{W}}_{GC}^{K}(i,j)|$ of urban road network in the GIS map, as shown in Fig. 15. Comparing Figs. 15 to 16, which depicts the traffic condition in peak hours, we observe that the trained graph convolution weight matrix tends to provide higher values on the congestion links (some of them are marked by black boxes). The spatial correlations in congested regions are relatively high due to the propagation of traffic waves.

(4) Relevance between temporal traffic pattern and attention coefficient

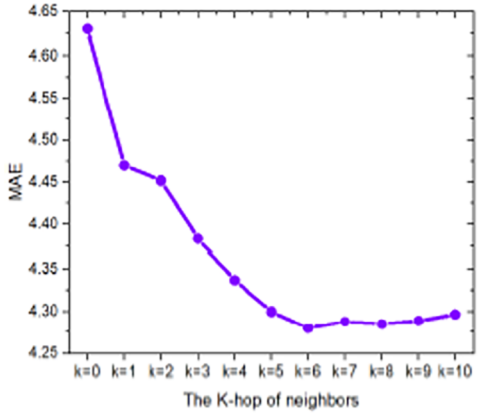
In the attention mechanism, the coefficient a_{i+j}^{t-i} provides a criterion to measure the relevance between the target and source-side information. Fig. 17 visualizes the attention coefficients under two typical scenarios. The attention heatmap of the road segment with drastic status changes (Scenario I) exhibits high values on the look-back observations within the past 15 min, while the corresponding attention coefficients of the smoothly changed traffic status (Scenario II) distribute uniformly among the temporal dimension. This indicates that the prediction model tends to rely on the recent information (within past 15 min) when the traffic status oscillates severely.



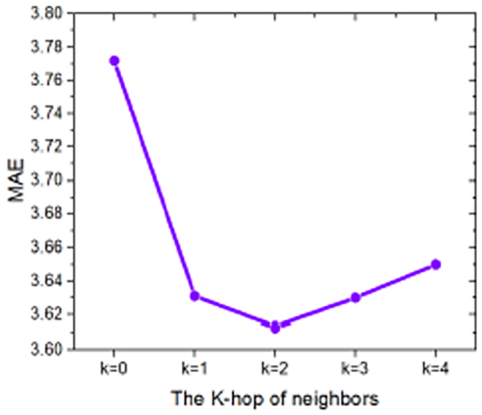
(a1) 5-min prediction on freeway network



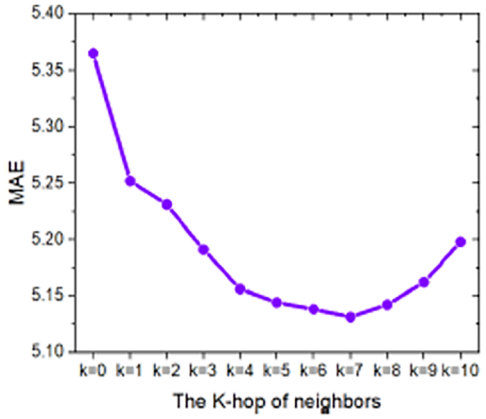
(b1) 5-min prediction on urban network



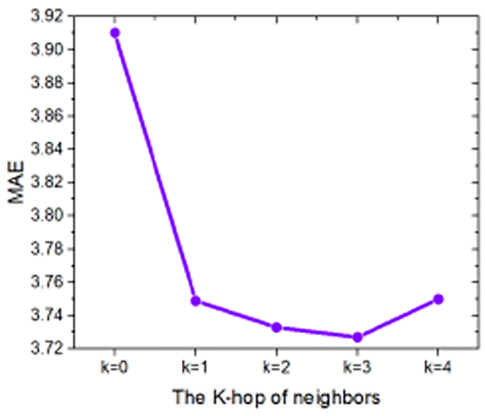
(a2) 15-min prediction on freeway network



(b2) 15-min prediction on urban network



(a3) 30-min prediction on freeway network



(b3) 30-min prediction on urban network

Fig. 13. Curves of prediction error varying with k-hop neighbors.

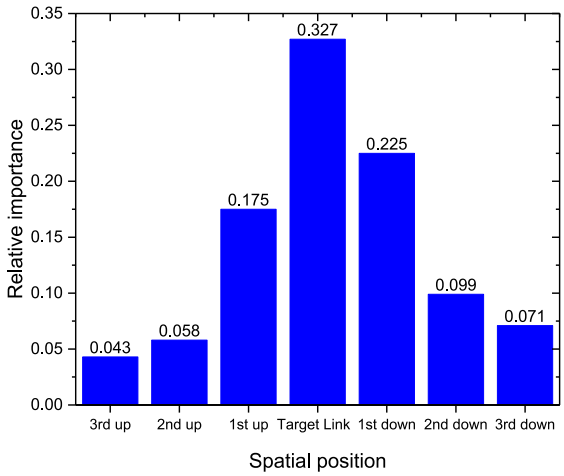


Fig. 14. The relative spatial importance.

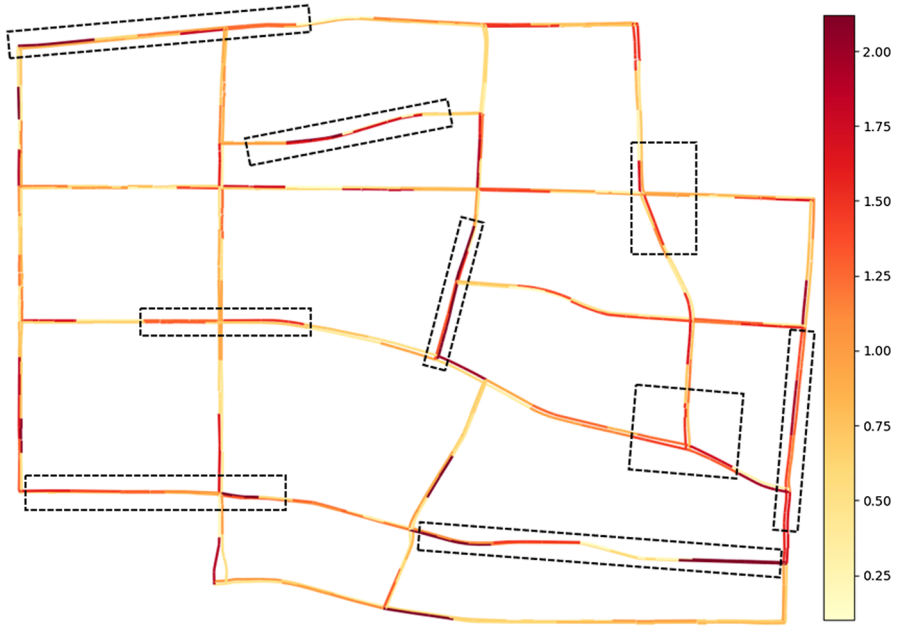


Fig. 15. Graph convolution weights in urban road network.

5. Conclusions

To tackle the challenges of multistep traffic speed prediction, we are devoted to proposing a sophisticated deep learning approach, i.e., the attention graph convolutional sequence-to-sequence model (AGC-Seq2Seq). The Seq2Seq architecture and graph convolutional operators are combined to learn the spatio-temporal dependencies on traffic networks. The attention mechanism is integrated into the model to capture the temporal heterogeneity of traffic patterns, and the entire architecture is trained with a newly designed method. To validate the effectiveness of the proposed model, we compare it with several benchmark models including the HA, ARIMA, XGBOOST, ANN, LSTM, SVR, KNN, GCN, and Seq2Seq-Att, based on two real-world datasets. The results indicate that the proposed model outperforms the benchmark models in terms of the MAPE, MAE, and RMSE under different prediction intervals.

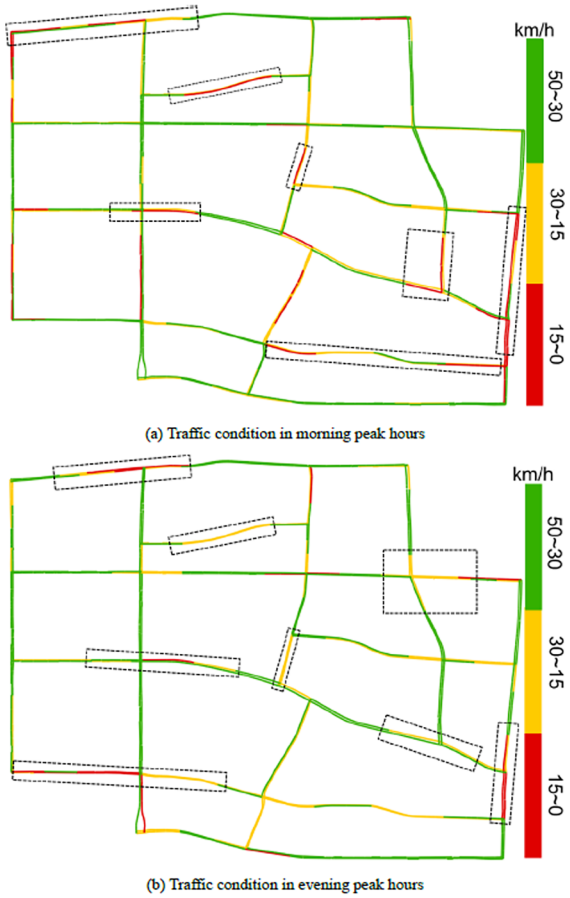


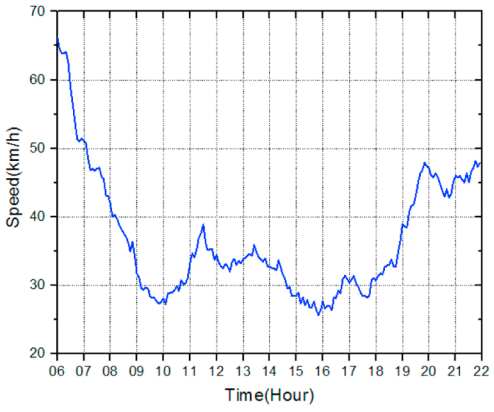
Fig. 16. Traffic condition in peak hours.

Based on the proposed model, we further explore the feature importance, effect of spatial information on multistep prediction, interpretation of graph convolution weights, and relevance between traffic temporal pattern and attention coefficients. The evidence from the experiment implies that both the relative importance of the features regarding exogenous information and the effect of increasing spatial information strengthen with the increase in the prediction intervals. The analysis of graph convolution weights shows the capacity of AGC-Seq2Seq to model spatial dependencies in the underlying traffic networks. For the road segments whose traffic condition changes rapidly, the corresponding attention coefficients take high values for the look-back observations within the past 15-min.

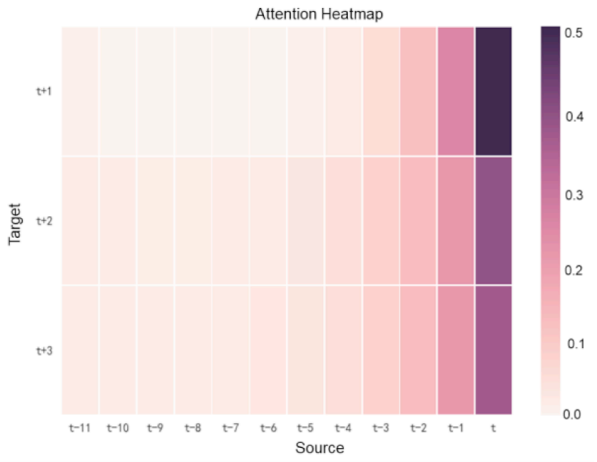
Future studies could include further integrating the traffic flow theories into the prediction model, e.g., utilizing the propagation waves of traffic flow (Zhang et al., 2018) and considering the traffic demand configuration (Ermagun and Levinson, 2018a) to determine the influential spatial neighbors in a more sophisticated model. From the application perspective, the proposed framework can be integrated with advanced transportation management systems, e.g., providing system-level real-time routing services to reduce peak-hour congestions.

Acknowledgments

This research is supported partially by grants from National Natural Science Foundation of China (71871126, 51622807, U1766205), National Key Research and Development Program of China (2018YFB1601601). The authors are grateful to A-map (https://www.amap.com/) for providing their anonymous users' GPS trajectory data. The research is supported in part by the Center for Data-Centric Management in the Department of Industrial Engineering at Tsinghua University and Tsinghua University-Toyota Research Center.

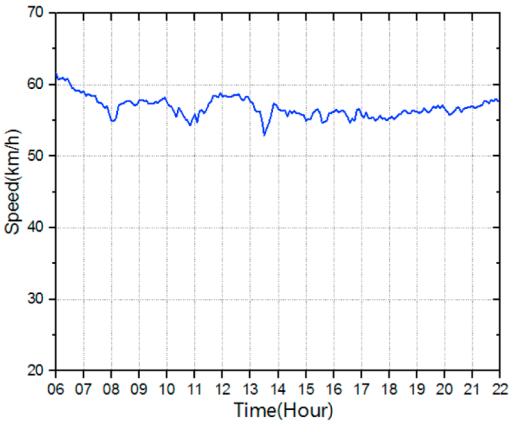


Traffic speed of link 6 (northern 2nd ring road)

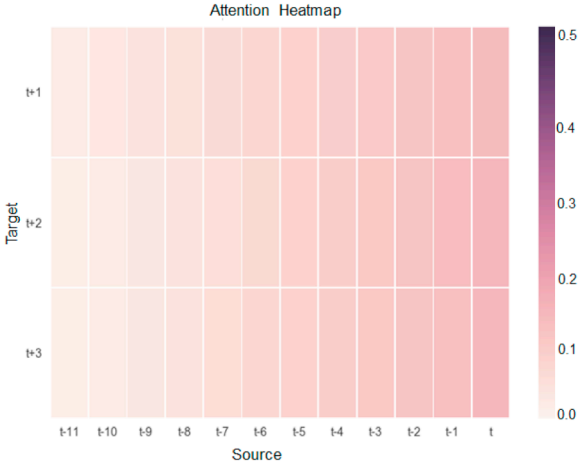


Attention heatmap of link 6 under 15-min horizon

(a) Scenario I: Traffic status with drastic changes



Traffic speed of link 57 (southern 2nd ring road)



Attention heatmap of link 57 under 15-min horizon

(b) Scenario II: Traffic status with smooth changes

Fig. 17. Visualization of attention coefficient matrix.

Appendix A

The well-known approaches for multistep prediction could be divided into three categories.

Category 1

This category of methods rolls the prediction horizon step-by-step as shown in Eq. (A1) below, and it is widely adopted by time-series models (e.g. AR and ARIMA). Since the posterior predicted values are based on prior predicted ones, the error will accumulate through this process.

$$\begin{aligned}
\hat{v}_{t+1} &= \underset{v_{t+1}}{\operatorname{argmax}} \ \Pr(v_{t+1}|v_t, v_{t-1}, \cdots, v_{t-m}) \\
\hat{v}_{t+2} &= \underset{v_{t+2}}{\operatorname{argmax}} \ \Pr(v_{t+2}|\hat{v}_{t+1}, v_t, v_{t-1}, \cdots, v_{t-m+1}) \\
&\vdots \\
\hat{v}_{t+n} &= \underset{v_{t+n}}{\operatorname{argmax}} \ \Pr(v_{t+n}|\hat{v}_{t+n-1}, \hat{v}_{t+n-2}, \cdots, \hat{v}_{t+1}, v_t, v_{t-1}, \cdots, v_{t-m+n-1})
\end{aligned} \tag{A1}$$

Category 2

The second category directly learns the dependencies of n^{th} prediction value and look-back observations as given by Eq. (A2). Some classical machine learning models (e.g. SVR, XGBOOST and KNN) utilize this way to make multistep prediction. Since the temporal relevance diminishes with the increase of prediction intervals, the models may fail to capture the variation pattern when the traffic status oscillates seriously.

$$\hat{v}_{t+n} = \underset{v_{t+n}}{\text{argmax}} \ \Pr(v_{t+n}|v_t, v_{t-1}, \cdots, v_{t-m}) \tag{A2}$$

Category 3

The third category makes multistep prediction by training the parameters to cooperatively reduce errors from 1^{st} to n^{th} prediction values given by Eq. (A3). Kuznetsov and Mariet (2018) provided one theoretical proof for the advantage of such way in time series modeling.

$$(\hat{v}_{t+1}, \cdots, \hat{v}_{t+n}) = \underset{v_{t+n}, \cdots, v_{t+1}}{\text{argmax}} \ \Pr(v_{t+n}, \cdots, v_{t+1}|v_t, v_{t-1}, \cdots, v_{t-m}) \tag{A3}$$

Appendix B

The information of the links in urban road network

#	L(m)	Lanes	W (m)	#	L(m)	Lanes	W(m)	#	L(m)	Lanes	W(m)
1	242	3	9	41	205	2	6	81	460	4	12
2	194	3	9	42	253	2	6	82	476	2	6
3	175	3	9	43	220	3	9	83	249	4	12
4	220	5	15	44	195	3	9	84	296	6	18
5	235	3	9	45	198	3	9	85	264	4	12
6	264	3	9	46	356	3	9	86	282	5	15
7	253	3	9	47	215	4	12	87	330	2	6
8	205	4	12	48	244	4	12	88	243	2	6
9	209	4	12	49	226	4	12	89	335	2	6
10	197	3	9	50	277	3	9	90	293	4	12
11	203	3	9	51	329	3	9	91	232	4	12
12	234	3	9	52	260	3	9	92	313	5	15
13	177	3	9	53	332	4	12	93	232	3	9
14	224	3	9	54	215	3	9	94	262	4	12
15	183	3	9	55	246	4	12	95	212	3	9
16	210	5	15	56	229	4	12	96	301	3	9
17	238	3	9	57	252	6	18	97	361	3	9
18	263	3	9	58	308	2	6	98	321	4	12
19	222	3	9	59	317	2	6	99	345	2	6
20	199	5	15	60	336	2	6	100	182	2	6
21	201	3	9	61	277	2	6	101	355	2	6
22	201	3	9	62	244	2	6	102	358	4	12
23	165	3	9	63	221	3	9	103	311	2	6
24	226	5	15	64	245	3	9	104	356	3	9
25	213	3	9	65	224	4	12	105	263	2	6
26	228	3	9	66	263	4	12	106	306	2	6
27	258	3	9	67	255	4	12	107	184	2	6
28	314	3	9	68	303	4	12	108	347	2	6
29	261	2	6	69	474	2	6	109	151	2	6
30	318	2	6	70	317	4	12	110	298	2	6
31	278	2	6	71	471	4	12	111	366	3	9
32	478	4	12	72	305	3	9	112	267	3	9
33	345	5	15	73	261	3	9	113	262	3	9
34	269	6	18	74	331	3	9	114	291	3	9
35	231	3	9	75	292	3	9	115	185	2	6
36	207	3	9	76	265	2	6	116	267	2	6
37	263	3	9	77	194	2	6	117	219	2	6
38	305	3	9	78	238	2	6	118	339	2	6
39	164	2	6	79	239	2	6	119	337	3	9
40	301	2	6	80	339	4	12	120	211	3	9

#	L(m)	Lanes	W (m)	#	L(m)	Lanes	W(m)	#	L(m)	Lanes	W(m)
121	206	5	15	162	282	3	9	203	249	3	9
122	198	2	6	163	223	4	12	204	259	3	9
123	367	2	6	164	251	5	15	205	311	3	9
124	475	2	6	165	314	5	15	206	303	2	6
125	247	3	9	166	271	4	12	207	177	2	6
126	299	3	9	167	229	3	9	208	329	3	9
127	274	3	9	168	274	4	12	209	474	3	9
128	300	4	12	169	232	3	9	210	452	3	9
129	344	6	18	170	297	4	12	211	607	3	9
130	306	5	15	171	280	4	12	212	174	4	12
131	373	5	15	172	259	5	15	213	282	3	9
132	369	3	9	173	227	5	15	214	329	3	9
133	312	3	9	174	251	5	15	215	264	4	12
134	262	3	9	175	142	5	15	216	223	4	12
135	464	2	6	176	374	5	15	217	270	5	15
136	262	6	18	177	261	6	18	218	195	3	9
137	236	3	9	178	280	5	15	219	320	3	9
138	251	3	9	179	265	4	12	220	243	4	12
139	213	5	15	180	260	5	15	221	281	4	12
140	282	4	12	181	261	6	18	222	245	6	18
141	235	4	12	182	258	6	18	223	325	4	12
142	299	4	12	183	235	6	18	224	326	4	12
143	191	5	15	184	268	5	15	225	317	6	18
144	223	3	9	185	219	3	9	226	297	5	15
145	298	3	9	186	223	3	9	227	234	5	15
146	237	4	12	187	360	3	9	228	337	5	15
147	201	4	12	188	145	4	12	229	394	5	15
148	298	4	12	189	325	2	6	230	244	6	18
149	296	4	12	190	258	2	6	231	437	4	12
150	225	5	15	191	158	2	6	232	434	2	6
151	219	4	12	192	170	4	12	233	531	2	6
152	186	5	15	193	210	2	6	234	402	3	9
153	222	4	12	194	272	3	9	235	381	3	9
154	157	5	15	195	273	3	9	236	426	4	12
155	226	3	9	196	206	3	9	237	360	4	12
156	206	4	12	197	248	2	6	238	408	3	9
157	189	3	9	198	250	2	6	239	390	3	9
158	176	5	15	199	387	4	12	240	485	3	9
159	256	3	9	200	415	2	6	241	565	3	9
160	332	4	12	201	469	2	6	242	351	3	9
161	259	3	9	202	498	2	6				

L = length, W = width.

References

M.S. Ahmed, A.R. Cook. Analysis of freeway traffic time-series data by using box-jenkins techniques; 1979.

D. Bahdanau, K. Cho, Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. Eprint arXiv. 2014.

Box, G., Pierce, E.P., David, A., 1970. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. Public. Am. Statist. Assoc. 65 (332), 1509–1526.

Cai, P., Wang, Y., Lu, G., Chen, P., Ding, C., Sun, J., 2016. A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. Transp. Res. Part C 62, 21–34.

Castro-Neto, M., Jeong, Y.S., Jeong, M.K., Han, L.D., 2009. Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. Exp. Syst. Appl.: Int. J. 36 (3), 6164–6173.

Chen, T., Guestrin, C., 2016. XGBoost: A Scalable Tree Boosting System. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016, pp. 785–794.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724–1734.

J. Chung, C. Gulcehre, K.H. Cho, Y. Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. Eprint arXiv; 2014.

Z. Cui, K. Henrickson, R. Ke, Y. Wang. High-Order Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting. Eprint arXiv; 2018.

M. Defferrard, X. Bresson, P. Vandergheynst. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. Eprint arXiv; 2016.

Denoeux, T., 1995. A k-nearest neighbor classification rule based on Dempster-Shafer theory. Syst. Man Cyber. IEEE Transac. 25 (5), 804–813.

Ermagun, A., Levinson, D., 2018b. Spatiotemporal traffic forecasting: review and proposed directions. Transp. Rev. 38 (6), 786–814.

Ermagun, A., Levinson, D.M., 2018c. Development and application of the network weight matrix to predict traffic flow for congested and uncongested conditions. Environ. Plan. B Urban Anal. City Sci 2399808318763368.

Ermagun, A., Levinson, D., 2018a. An introduction to the network weight matrix. Geograph. Anal. 50 (1), 76–96.

Ermagun, A., Chatterjee, S., Levinson, D., 2017. Using temporal detrending to observe the spatial correlation of traffic. PloS One 12 (5), e0176853.

Fang, J., Tu, L., 2018. Prediction based active ramp metering control strategy with mobility and safety assessment. In: AIP Conference Proceedings. AIP Publishing, pp. 040058.

Fusco, G., Colombaroni, C., Isaenko, N., 2016. Short-term speed predictions exploiting big data on large urban road networks. Transp. Res. Part C 73, 183–201.
Gao, Y., Sun, S., Shi, D., 2011. Network-scale traffic modeling and forecasting with graphical lasso. Int. Conf. Adv. Neural Netw. 151–158.
Guo, J., Huang, W., Williams, B.M., 2014. Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. Transp. Res. Part C Emerg. Technol. 43, 50–64.
Habtemichael, F.G., Cetin, M., 2016. Short-term traffic flow rate forecasting based on identifying similar traffic patterns. Transp. Res. Part C 66, 61–78.
Hamner, B., 2011. Predicting travel times with context-dependent random forests by modeling local and aggregate traffic flow. In: IEEE International Conference on Data Mining Workshops, pp. 1357–1359.
Hashemi, H., Abdelghany, K., 2015. Real-time traffic network state prediction for proactive traffic management: simulation experiments and sensitivity analysis. Transp. Res. Record: J. Transp. Res. Board (2491), 22–31.
Hashemi, H., Abdelghany, K.F., 2016. Real-time traffic network state estimation and prediction with decision support capabilities: application to integrated corridor management. Transp. Res. Part C: Emerg. Technol. 73, 128–146.
Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.
Huang, W., Song, G., Hong, H., Xie, K., 2014. Deep architecture for traffic flow prediction: deep belief networks with multitask learning. IEEE Trans. Intell. Transp. Syst. 15 (5), 2191–2201.
Y. Jiang, R. Kang, D. Li, S. Guo, S. Havlin. Spatio-temporal propagation of traffic jams in urban traffic networks. arXiv preprint arXiv:1705.08269; 2017.
Ke, J., Zheng, H., Yang, H., Xiqun, Chen, 2017. Short-term forecasting of passenger demand under on-demand ride services: a spatio-temporal deep learning approach. Transp. Res. Part C Emerg. Technol. 85, 591–608.
T.N. Kipf, M. Welling. Semi-Supervised Classification with Graph Convolutional Networks. Eprint arXiv; 2016.
V. Kuznetsov, Z. Mariet. Foundations of Sequence-to-Sequence Modeling for Time Series. Eprint arXiv; 2018.
Y. Li, R. Yu, C. Shahabi, Y. Liu. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. Eprint arXiv; 2017.
Liebig, T., Piatkowski, N., Bockermann, C., Morik, K., 2017. Dynamic route planning with real-time traffic predictions. Inform. Syst. 64, 258–265.
Lin, Y., Wang, P., Ma, M., 2017. Intelligent Transportation System (ITS): Concept, Challenge and Opportunity. In: IEEE International Conference on Big Data Security on Cloud, pp. 167–172.
M.T. Luong, H. Pham, C.D. Manning. Effective Approaches to Attention-based Neural Machine Translation. Eprint arXiv; 2015.
Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.Y., 2015. Traffic flow prediction with big data: a deep learning approach. IEEE Trans. Intell. Transp. Syst. 16 (2), 865–873.
Ma, X., Tao, Z., Wang, Y., Yu, H., Wang, Y., 2015. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transp. Res. Part C Emerg. Technol.s 54, 187–197.
Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., Wang, Y., 2017. Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. Sensors 17 (4).
Narang, Sunil K., Ortega, Antonio, Vandergheynst, Pierre, 2013. The emerging field of signal processing on graphs. IEEE Signal Process Mag. 30 (3), 83–98.
Okutani, I., Stephanedes, Y.J., 1984. Dynamic prediction of traffic volume through Kalman filtering theory. Transp. Res. Part B 18 (1), 1–11.
Polson, N.G., Sokolov, V.O., 2017. Deep learning for short-term traffic flow prediction. Transp. Res. Part C Emerg. Technol. 79, 1–17.
Qureshi, K.N., Abdullah, A.H., 2013. A survey on intelligent transportation systems. Middle East J. Scient. Res. 15, 629–642.
Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1988. Learning representations by back-propagating errors. MIT Press.
Sohu, 2018. http://www.sohu.com/a/258164241_792633. Accessed on October 23, 2018.
Sutskever, I., Vinyals, O., Le, Q.V., 2014. Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112.
Suykens, J.A.K., Vandewalle, J., 1999. Least squares support vector machine classifiers. Neural Process. Lett. 9 (3), 293–300.
Vlahogianni, E.I., Golias, J.C., Karlaftis, M.G., 2004. Short-term traffic forecasting: overview of objectives and methods. Transp. Rev. 24 (5), 533–557.
Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2005. Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach. Transp. Res. Part C 13 (3), 211–234.
Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2014. Short-term traffic forecasting: where we are and where we're going. Transp. Res. Part C Emerg. Technol. 43, 3–19.
Wang, J., Gu, Q., Wu, J., Liu, G., Xiong, Z., 2017. Traffic speed prediction and congestion source exploration: a deep learning method. IEEE Int. Conf. Data Mining 499–508.
Wang, Y., Zhang, Y., Piao, X., Liu, H., Zhang, K., 2018. Traffic data reconstruction via adaptive spatial-temporal correlations. In: IEEE Transactions on Intelligent Transportation Systems, pp. 1–13.
Williams, B.M., Hoel, L.A., 2003. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results. J. Transp. Eng. 129 (6), 664–672.
Yu, B., Yin, H., Zhu, Z., 2018. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence. AAAI Press, pp. 3634–3640.
Yuan, J., Zheng, Y., Xie, X., Sun, G., 2011. Driving with knowledge from the physical world. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 316–324.
Yuan, J., Zheng, Y., Xie, X., Sun, G., 2013. T-drive: enhancing driving directions with taxi drivers' intelligence. IEEE Trans. Knowl. Data Eng. 25 (1), 220–232.
Zhang, L., Liu, Q., Yang, W., Wei, N., Dong, D., 2013. An improved K -nearest neighbor model for short-term traffic flow prediction. Procedia – Social Behav. Sci. 96, 653–662.
Zhang, Y., Smirnova, M., Bogdanova, A., Zhu, Z., Smirnov, N., 2018. Travel time estimation by urgent-gentle class traffic flow model. Transp. Res. Part B: Methodol. 113, 121–142.