

Received March 13, 2020, accepted April 24, 2020, date of publication May 11, 2020, date of current version May 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2992507

Short-Term Traffic Speed Prediction of Urban Road With Multi-Source Data

XUN YANG¹, YU YUAN¹, AND ZHIYUAN LIU¹

Jiangsu Key Laboratory of Urban ITS, Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, School of Transportation, Southeast University, Nanjing 210096, China

Corresponding author: Zhiyuan Liu (zhiyuanl@seu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1600900, in part by the General Projects under Grant 71771050, and in part by the National Natural Science Foundation of China through the Key Project under Grant 51638004.

ABSTRACT In recent years, the convolutional and recurrent neural networks are widely applied in traffic prediction tasks. Traffic speed prediction is an important and challenging topic in intelligent transportation systems. In this case, this paper proposes a hybrid deep learning structure for short-term traffic speed prediction, which combines convolutional neural networks and long short-term memory neural networks together. External factors such as weather condition and air quality can also affect the driving behavior of travelers and cause fluctuation of traffic speed. Based on theories in traffic engineering, we propose a data-fusion method to measure the impact of environmental factors. To enhance the performance of our model, we introduced attention mechanism to our model. With convolutional block attention module, our network could emphasize important channels and pixels of input features and suppress unnecessary ones. Comparing with several deep learning methods and hybrid deep learning structures, an experiment in one region of Suzhou which contains 909 links shows the outperformance of our model. Under different time steps, the prediction error of our model is lower than any other methods in urban expressway, primary-arterial, secondary-arterial, and branch-road. The results indicate that the spatial dependencies, the temporal correlations, and environmental impact should not be ignored in traffic speed prediction tasks.

INDEX TERMS Traffic speed prediction, convolutional neural network, long short-term memory, attention mechanism, environmental factor.

I. INTRODUCTION

Over the past few years, the rapid progress of urbanization and motorization has caused many urban problems, such as traffic congestion in metropolises around China. According to China Urban Transportation Report in 2019 released by Baidu, people in Beijing and Chongqing will cost more than twice the time (2.165 and 2.040) on the same road during rush hour. In order to meet people's traffic demand and mitigate these problems, governments have taken much efforts in traffic planning and urban traffic infrastructures. However, it is impossible to completely satisfy the increasing traffic demand with limited land resources. Therefore, improving the efficiency of the existing roads is important and necessary [1]. Intelligent transportation systems (ITS) has been widely applied in China [2], [3]. As one of the most important topics in ITS, precise short-term traffic speed prediction is

useful in road guidance and speed inducing, which can help drivers to avoid traffic congestion and get fast, safe, and comfortable trip experience [4], [5].

So far, numerous studies have proposed various models for traffic speed prediction. These models can be mainly divided into two divisions: parametric and non-parametric models [6], [7]. Machine learning approaches like neural networks (NNs) have been widely applied in ITS and perform well [8]. However, traditional NNs fail to extract the spatial and temporal correlations among different road links in traffic prediction tasks. In recent years, advanced sensing and computing technologies bring us multi-source heterogeneous data of our city [9]. At the same time, deep learning methods achieved great success in computer vision (CV) and natural language process (NLP). Among these various deep learning methods, convolutional neural networks (CNNs) is widely utilized in traffic speed prediction to extract spatial relationships among different road links. As a special recurrent neural networks (RNNs), long short-term memory networks

The associate editor coordinating the review of this manuscript and approving it for publication was Mamoun Alazab¹.

(LSTMs) is powerful to capture the temporal evolution pattern of time-series data such as traffic state parameters and traffic demand prediction [46]. LSTMs is prevalent in related speed prediction works [4], [10]–[13]. With the combination of CNNs, LSTMs and graph neural networks (GCNs), etc., researchers build remarkable spatiotemporal deep learning structures [10], [14]. But problems such as complex data preprocessing and restriction of input (e.g., adjacent matrix) cause low portability of these models. External factors such as weather condition and air quality can affect the driving behavior of travelers and cause fluctuation of traffic speed. By taking these environmental factors into consideration, our model will be more corresponding to reality and get better results as well [15]. In this paper, we propose a hybrid deep learning framework based on two dimensional CNNs (2D CNNs) and LSTMs for short-term traffic speed prediction. We also introduce attention mechanism to improve our model. The contributions of our work are two-fold:

- 1) We design a hybrid deep learning structure HDL-net for short-term traffic speed prediction. HDL-net combines multi-2DCNN layers and LSTMs together to capture the spatial and temporal correlations of traffic speed among different road links. Comparing with sequence-based and grid-based model, HDL-net is highly-portable. HDL-net focuses on partial road network structure for each road link, and the input data is quite simple. We also introduce convolutional block attention module (CBAM), a widely-used attention block in CNNs to enhance the performance of our model.
- 2) We introduce a convective data-fusion method to measure the impact of external features. This method matches the theory of free flow speed calculating in traffic engineering field.

This paper is organized as follows: Section II gives the literature review of traffic prediction, including parametric and non-parametric approaches. Section III introduces the details of the hybrid deep learning framework (HDL-net). We illustrate the mechanism of CNNs, LSTMs, CBAMs and data preprocessing approaches in our work. In Section IV, an urban road network with 909 road links in Suzhou is employed for the validation of our model. We use mean absolute error (MAE) and mean absolute percentage error (MAPE) to measure the performance of our model with several prevailing deep learning architectures. We present conclusions of our work in section V.

II. LITERATURE REVIEW

The presented methods for traffic prediction can be classified into two categories: parametric models and non-parametric models.

A. PARAMETRIC MODELS

The parametric model presumes that the traffic speed follows a probability distribution with a fixed set of parameters.

In primary studies, researchers focused on techniques of time series data analysis. Auto-Regressive Moving Average model (ARMA), a classical method in series data analysis, was introduced in traffic prediction by Smith *et al.* [16]. Hamed *et al.* proposed autoregressive integrated moving average (ARIMA) with the order (0,1,1), which performs well in volume and occupancy prediction [17]. W. Min and L. Wynter developed multivariate spatial-temporal auto-regressive (MSTAR) to extract spatial dependency of traffic speed [18]. Next, Ding *et al.* introduced a single space-time autoregressive integrated moving average model (STARIMA) which effectively describe the spatiotemporal variation of traffic flow [19]. Another widely used model is Kalman filter (KF). It has been applied in traffic flow prediction on freeway and achieved fantastic result in travel time prediction [20], [21]. Besides, Markov logic network (MLN) and exponential smoothing (ES) method are prevailing methods in traffic flow and congestion state prediction [22], [23].

B. NON-PARAMETRIC MODELS

The non-parametric model is data-driven and it makes no distribution assumptions. The number of parameters in these models is related to the scale of the training data. Recent advances in machine learning field brings researchers new tools for traffic prediction. An example is K-nearest neighbor (KNN) non-parameter regression method [24]. Neural network based (NN-based) model is a prevailing non-parametric approach, which can approximate complex nonlinear function. With NN's excellent capability of learning features from multi-dimensional data, Huang *et al.* considered weather condition as environmental factors in their model [25]. Yin *et al.* applied fuzzy neural networks (FNNs) for traffic forecasting in high-speed network. In related works, Tang *et al.* improved the model to extract periodic features [26], [27].

In recent years, researchers begun to focus on the application of deep learning methods in traffic prediction. For instance, the deep belief networks (DBFs) and stacked auto encoder (SAEs) achieved great improvements over traditional NNs [28], [29]. However, these methods are weak in capturing time-series characteristics. To overcome this problem, recurrent neural networks, such as LSTMs are utilized in traffic speed forecasting. LSTMs could adjust the time lag automatically and capture long-term temporal features of the speed data [13]. Even so, problems still exist in modelling temporal patterns. Great success in CV shows the capability of convolutional neural networks (CNNs) to extract spatial dependencies. Following this, researchers proposed spatiotemporal deep learning models by combining different networks together. Wu *et al.* conducted a short-term traffic flow prediction model [30]. They combined 1D CNNs and LSTMs together to mine the spatiotemporal correlations of traffic speed. They converted the speed values of all the road links on one timestamp into one dimensional vector (sequence), which ignored the topological relationship of road links. Yu *et al.* developed a grid-based network

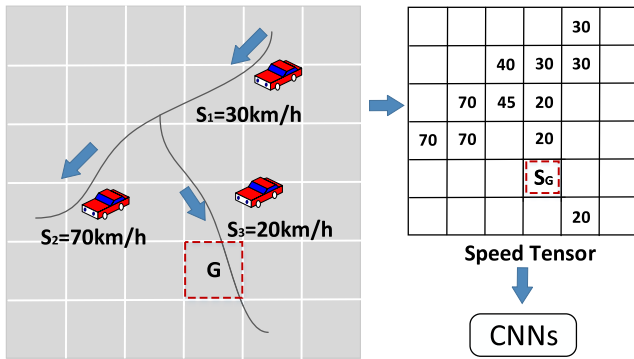


FIGURE 1. Mechanism of grid-based model for traffic speed prediction. Grid G covers one segment of the road network, the pixel value $S_G = S_3 = 20\text{km/h}$. If one grid contains two or more road segments, the speed value of this grid will be the average speed of these road segments.

segmentation method named spatiotemporal recurrent convolutional networks (SRCNs) for traffic speed predicting. This model combines deep 2D CNNs and LSTMs together [14]. Fig. 1 illustrates the mechanism of grid-based method in traffic speed prediction. High-precision geographic data of the entire road network is indispensable in grid-based model. Smaller grid-size indicates better inputs and results, but harder works in data preprocessing at the same time. Graph convolutional neural networks (GCNs) is prevalent in traffic predicting during recent years [10], [31], [32]. GCNs works well in non-Euclidean structure for its excellent ability in spatial correlations processing [33]. However, the adjacent matrix of the entire road network is necessary in GCNs. In addition, traffic speed of one road section is mainly influenced by those links connecting with it for traffic flow merging and diverging. Using multi-layers GCNs to get multi-level spatial relationship makes little sense. To address these drawbacks, this paper represents a novel deep learning structure HDL-net. For each partial road network structure, we can get column vector of traffic speed. We compose all the vectors horizontally together to form a graph and learn the evolution process of road speed as a video with multi-2DCNN layers and LSTMs. Besides, we use CBAM to improve our model. Comparing with several deep learning methods, HDL-net can effectively extract high-level spatiotemporal features of traffic speed data. The inputs of HDL-net are very simple. It has good generalization ability for different road networks.

III. METHODOLOGY

In this section, we illustrate the details of our works. In subsection A, we discuss internal feature: traffic speed. We give definitions of road links and road speed classification. Subsection B introduces 2D CNNs and how we converted speed data of different links at different timestamp into tensors. Subsection C shows the mechanism of CBAM. In subsection D, the structures of LSTM cells and networks are given. Then we discuss the impact of external features and shows the ideas about data-fusion in subsection E. Finally, the data processing approaches and the overview structure of the HDL-net and our work is illustrated in subsection F.

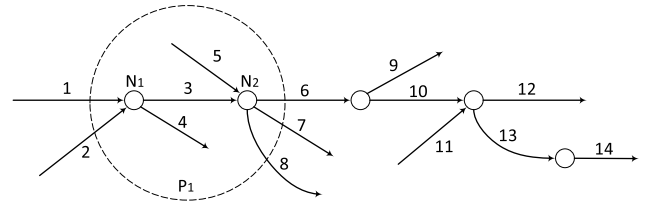


FIGURE 2. A road network with 14 links.

A. INTERNAL FEATURE

In this section, a simple road network with 14 links is built to illustrate the road structure and explain the idea of road speed and road link classification.

In Fig. 2, the solid circles (e.g., N_1 and N_2) refer to connection nodes, which represent the start and end points of road links. Arrows indicate road links with fixed directions (e.g., Link 1 and Link 2). The empty circle with dashed line contains part of the road network structure (e.g., circle P_1). For each road link, there are several links connecting with it. These links can be divided into four categories. Take link 3 as an example:

- 1) *Inflow links (ILs)*: Links pointing to the start point of link k (link 1, 2). *ILs* include on-ramps to the start point of link k.
- 2) *Outflow links (OLs)*: Links pointing out the end of link k (link 6, 7, 8). *OLs* include off-ramps from the end point of link k.
- 3) *Same start links (SLs)*: Links with the same start point as link k (link 4). *SLs* are off-ramps from the start point of link k.
- 4) *Same end links (ELs)*: Links with the same end point as link k (link 5). *ELs* are on-ramps to the end point of link k.

Table. 1 gives some examples about link-classification based on this road network. As we mentioned before, P_1 describes the connection relationship between link 3 and its neighbors. In this paper, we focus on the spatial relationship of one road within this circle.

TABLE 1. Examples of link classification based on the road network in Fig. 2.

Link ID	Link Classification			
	<i>ILs</i>	<i>OLs</i>	<i>SLs</i>	<i>ELs</i>
3	1, 2	6, 7, 8	4	5
6	3, 5	9, 10	7, 8	null
12	10, 11	null	13	null

Traffic speed of these links are named correspondingly as *inflow speed (IS)*, *outflow speed (OS)*, *same start speed (SS)*, and *same end speed (ES)*. Assume that there are a set number of observed speed records from all the links, which contains features like *link id*, *timestamp* and *observed speed value*, etc. According to (1), we can get the average speed of one link

during specific time period.

$$\bar{S}_{kv} = \frac{1}{m} \sum_{i=0}^m S_{kv}^i \quad (1)$$

where \bar{S}_{kv} and S_{kv}^i represent the average speed and i_{th} observed record of link k during period v . m is equal to the number of these records. For instance, if there are 200 observed speed records in link 6 from 12:00 to 12:15, the average speed of link 6 in this time period will be the mean value of these 200 records.

$$\bar{S}_{6,12:00-12:15} = \frac{1}{200} \sum_{i=1}^{200} S_{6,12:00-12:15}^i \quad (2)$$

Due to the variety of road network structure, the link-number in each category is different. In order to build a speed vector with determinate size, we use average speed which is defined in (2) to represent the traffic speed of each link category. The 2D CNNs introduced in Section B is based on these representative speed values.

$$\bar{S}_{kc} = \frac{1}{n} \sum_{i=0}^n S_{kc}^i \quad (3)$$

\bar{S}_{kc} and S_{kc}^i indicate the representative speed value and the i_{th} speed value of category c in link k . m is the number of links in category c . Links 3 and 12 are used to illustrate the calculating process as follows in Table. 2. As Fig. 2 shows, there are not *OLs* or *ELs* connecting with link 12. We use *null* to mark links which do not exist.

TABLE 2. Example of representative speed value calculating.

Speed Category	Link ID	
	3	12
<i>IS</i>	$S_{3(IS)} = (S_1 + S_2) / 2$	$S_{12(IS)} = (S_{10} + S_{11}) / 2$
<i>OS</i>	$S_{3(OS)} = (S_6 + S_7 + S_8) / 3$	<i>null</i>
<i>SS</i>	$S_{3(SS)} = S_4$	$S_{12(SS)} = S_{13}$
<i>ES</i>	$S_{3(ES)} = S_5$	<i>null</i>

B. SPEED-TENSOR AND ONE-DIMENSIONAL CONVOLUTIONAL NEURAL NETWORKS

Up to now, many studies have proved that CNNs is suitable for training data with strong spatial correlations, such as crowd flow, traffic speed, etc. From the perspective of input size and filter size, CNN can be mainly divided into 3 categories. We list the main idea of these CNN-based models in traffic prediction as follows:

- 1) 1D CNNs (sequence-based model): traffic data to sequence, mining spatial or temporal correlations [34], [35].
- 2) 2D CNNs (grid-based model): traffic data to image, mainly for capturing spatial relationships [11], [14].

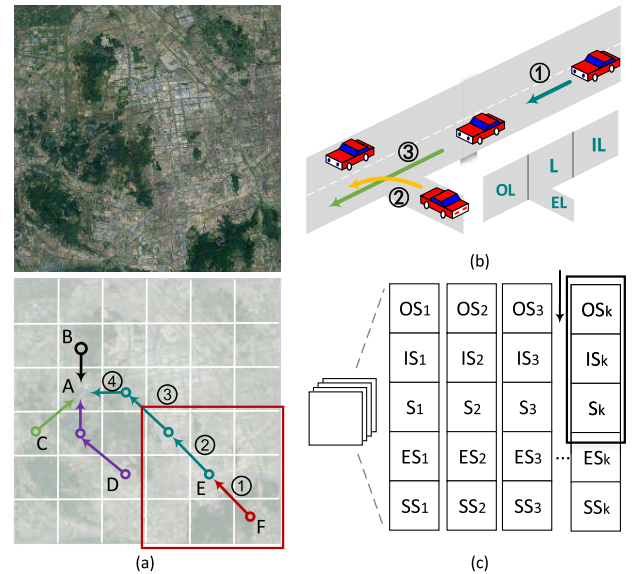


FIGURE 3. (a) Full view of Suzhou, China and overview of grid-based crowd flow prediction method (picture from Google Earth) (b) Schematic diagram of urban road (c) Speed value to speed-tensor and 2D CNNs with rectangular filter.

- 3) 3D CNNs (grid-based model): traffic data to images with different channels, extracting spatial and temporal dependencies simultaneously [36].

In traffic prediction model, deeper CNN structure indicates wider cognitive field [11]. Fig. 3(a) gives an example of grid-based model for crowd flow prediction. It takes time for pedestrians and cars to move from one region to another. In short-term traffic prediction, the crowd flow of one region (e.g., region A) is mainly influenced by regions connecting with it (region B and C, etc.) or regions nearby (e.g., region D). Now we consider the impact of region F. Here we choose a 3×3 filter to extract spatial correlations. When the depth of CNNs is 1, the crowd flow impact of region F will be spread through path ① to region E. Similarly, when the depth of CNN is 4, the crowd flow impact of region F will be spread to region A through path ①–④. In other words, with a 4-layer CNN-based model, we can model the impact of region F to region A. Likewise, in short-term traffic speed prediction, we transform our speed data into tensors. Those links which have greater impact to link k will be arranged connecting with link k in the speed-tensor. After that, we use multi-CNN layers to capture the spatial correlations among these links. With equation (3) introduced in section A, we can get the representative speed value of *IS*, *OS*, *SS*, and *ES* of every link in our road network. For each link in the road network, we combine

- 1) speed value of it (*S*)
- 2) representative speed value of links connecting with it (*IS*, *OS*, *SS*, and *ES*)

vertically together to form a column vector. In this way, we can get k column vectors with a road network which contains k links. Then we assemble these column vectors

horizontally together and the result will be a $H \times W$ speed-tensor. Here $H=5$ and $W=$ link number.

In fact, there are 12 possible sequence arrangement mode of the column vector ($C_4^2 C_2^1 = 12$). As Fig. 3(b) shows, travelers drive from IL to L through path ①, the continuity of traffic flow indicates the close relationship between IS and S. Besides, as path ② and ③ shows, in weaving section of L and SL, travelers tend to slow their speed down to avoid traffic conflict and keep safe. As reasons given above, IL and SL affect the speed of L much more. Therefore, we set IS and ES connecting with S in speed-tensor. The sequence mode we choose is shown in Fig. 3(c). In this paper, we use speed-tensor of time $t - 1 \times TI$, $t - 2 \times TI$, $t - 3 \times TI$ (TI: time interval), and $t - 1day$ to predict the speed of time t. We concatenate these four tensors to form a new $C \times H \times W$ tensor, or a ‘video’ with images as frames. Here C (channel) = 4, H (height) = 5, and W (width) = link number. As we mentioned before, we could choose 3D CNNs for our task. However, the parameters of 3D filters are fixed after training. All the channels of the input will follows same terms. For example, we want focus more on channel $t - 1day$ when timestamp t is on Tuesday, for both Monday and Tuesday are workdays and the traffic demand on urban roads follows an similar evolution pattern. The condition will be on the contrary on Saturday (weekend) and Friday (weekday). Besides, the impact of IS, OS, SS, and ES to S will be different in different road network structures. In this case, we build two 2D CNN layers to mine spatial dependencies and introduce convolutional block attention module (CBAM) to improve our model. We choose rectangular filter (3×1) to detect spatial features. The details are illustrated in Fig. 3(c). The mechanism of CBAM will be introduced in the following subsection.

C. CONVOLUTIONAL BLOCK ATTENTION MODULE

Attention mechanism has been widely utilized in traffic prediction tasks to improve the performance of the models, [37], [38]. With attention block, one model can focus on important features and suppressing unnecessary ones [39]. In order to emphasize meaningful features along channels and spatial pixels and compress unnecessary ones, we applied convolutional block attention module (CBAM) proposed by Woo et al. in our model [40].

$$\begin{aligned}
 A &= \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}, \\
 B &= \begin{pmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{m1} & \cdots & b_{mn} \end{pmatrix} \\
 A \otimes B &= \begin{pmatrix} a_{11} \times b_{11} & \cdots & a_{1n} \times b_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} \times b_{m1} & \cdots & a_{mn} \times b_{mn} \end{pmatrix} \quad (4)
 \end{aligned}$$

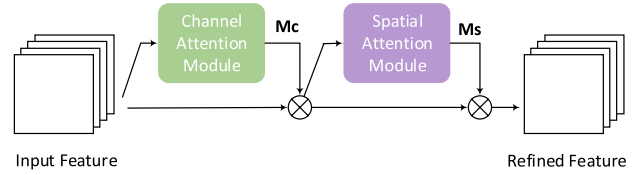


FIGURE 4. The overview of CBAM.

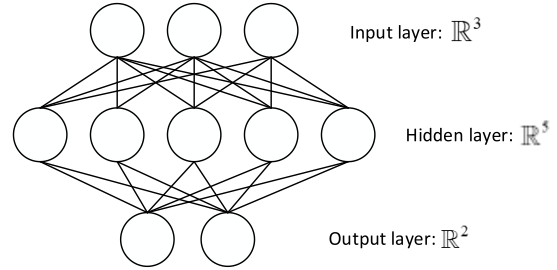


FIGURE 5. Structure of multi-layer perceptron (MLP).

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (6)$$

Firstly, we give the definition of Hadamard Product. Equation (4) shows the mechanism, where A and B are matrices with same shape $m \times n$. \otimes refers to the notation of Hadamard Product. Two prevailing non-linear activation functions are used in this paper, which can be defined as (5) and (6). As the overview of CBAM illustrated in Fig. 4, one CBAM block contains two sub-modules: channel attention module and spatial attention module.

The mechanism of CBAM can be divided into 4 steps:

1) Get Channel Attention Vector

Multi-Layer Perceptron (MLP) are basic neural networks with input layer, hidden layers, and output layer. Fig. 5. illustrates the structure of MLP with input \mathbb{R}^3 and output \mathbb{R}^5 . Pooling is an effective subsample method in CV, which can keep main features of the input, decrease the parameters, and avoid overfitting problem. The mechanism of Pooling is shown in Fig. 6. Fig. 7 demonstrates the structure of channel attention module. This sub-module get max-pooling and average-pooling vectors based on the input tensor $F = \mathbb{R}^{C \times H \times W}$. Then these two Pooling vectors are feed to a (MLP) blocks (parameters shared) and get new $\mathbb{R}^{C \times 1 \times 1}$ vectors. Finally, the module adds these two vectors together and activate the result with sigmoid function $sigmoid(x)$, which is marked as σ in Fig. 7. The output-vector is the channel attention vector $M_c = \mathbb{R}^{C \times 1 \times 1}$.

2) Channel Attention

Multiply the input F with attention vector $M_c(F)$ and get channel-refined tensor F_c as (7).

$$F_c = M_c(F) \otimes F \quad (7)$$

3) Get Spatial Attention Matrix

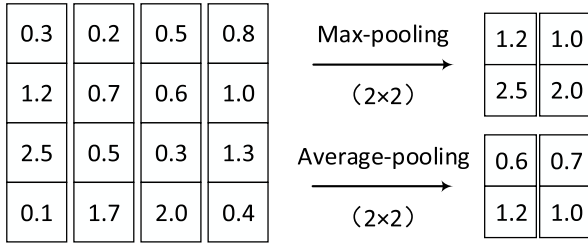


FIGURE 6. Mechanism of max-pooling and average-pooling.

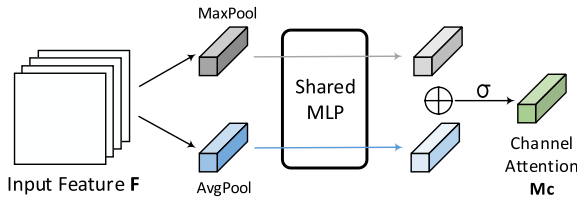


FIGURE 7. The overview of the channel sub-module.

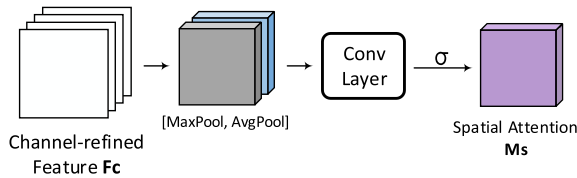


FIGURE 8. The overview of the spatial sub-module.

Fig. 8 gives the structure of spatial attention module. This sub-module gets Max-Pooling and Average-Pooling results based on channel-refined tensor $F_c = \mathbb{R}^{C \times H \times W}$ (axis: C). The results of Pooling are $\mathbb{R}^{1 \times H \times W}$ metrics. Then the module concatenates these two Pooling matrices together and use convolutional layers to blend them. Finally, the module activates the result with sigmoid function $sig(x)$ and the output is the channel attention matrix $M_s = \mathbb{R}^{1 \times W \times H}$. In our model, we choose 1×1 filter to get linear combination of Max-Pooling and Average-Pooling matrices

4) Spatial Attention

Multiply channel-refined tensor F_c and spatial attention matrix M_s together as (8). The output is channel-spatial-refined tensor F_s . In our model, F_s will be the input of multi-2D CNN layers.

$$F_s = M_s(F_c) \otimes F_c \quad (8)$$

In CBAM, M_c and M_s are masks which contain the weight of channels and pixels of the input tensor. For each input of speed-tensor, our model can judge the magnitude of impact among different channels (closeness: $t - 1 \times TI$, $t - 2 \times TI$, $t - 3 \times TI$, period: $t - 1day$) and pixels (IS , OS , S , SS , and ES) automatically.

D. LONG SHORT-TERM MEMORY NEURAL NETWORK

In this paper, the LSTM neural network is used to capture the temporal evolution of traffic speed. Traditional recurrent neural networks (RNNs) has been applied in traffic flow and

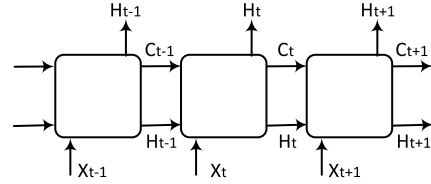


FIGURE 9. The structure of LSTM networks.

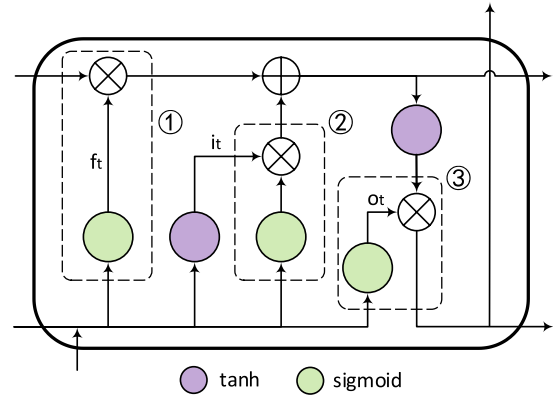


FIGURE 10. The structure of LSTM cells.

traffic speed prediction. In these models, the number of time steps ahead should be determined before training, and they do not work very well on those tasks with long time lags for gradient vanishing or gradient exploding. However, a traffic incident may cause traffic congestion in the following hours, which indicates the strong correlations between two traffic events with long time-interval. As a special recurrent NN, LSTM can determine the optimal time lags automatically. It works well in capturing the long-term temporal dependency of traffic speed [13], [47].

The structure of LSTM network and cell is shown in Fig. (9) and Fig. (10), where X_{t-1} , X_t , and X_{t+1} present speed inputs; H_{t-1} , H_t , and H_{t+1} refer to outputs.

Fig. 10 clearly demonstrates that there are three kinds of gates hidden in the memory cell. These three gates make LSTM capable of modeling the long-term dependencies, which are named correspondingly as ① forget gate, ② input gate, and ③ output gate. They are used to control the information added to or removed from the cell state in the next timestep. The purple and green circles refer to activation functions: $tanh(x)$ and $sigmoid(x)$. Equation (9) shows the calculating process of each step in the memory cell, where X_t represents the input of step t , H_{t-1} and H_t denote the output of step $t - 1$ and t respectively, c_{t-1} and c_t represent the cell state of step $t - 1$ and t , W_x indicates the weight matrix, and b_x refers to the corresponding bias. The mark \otimes refers to Hadamard product. In our model, we set X_{t-1} as input and h_t as predicted result.

$$\begin{aligned} i_t &= \sigma(W_{xi}X_t + W_{hi}H_{t-1} + W_{ci} \otimes c_{t-1} + b_i) \\ f_t &= \sigma(W_{xf}X_t + W_{hf}H_{t-1} + W_{cf} \otimes c_{t-1} + b_f) \\ c_t &= f_t \otimes c_{t-1} + i_t \otimes \tanh(W_{xc}X_t + W_{hc}H_{t-1} + b_c) \end{aligned}$$

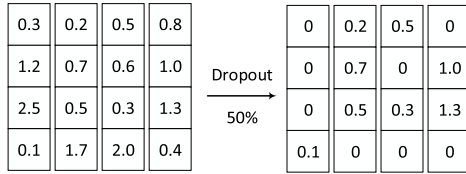


FIGURE 11. Mechanism of Dropout.

$$\begin{aligned} o_t &= \sigma(W_{x_o}X_t + W_{h_o}H_{t-1} + W_{c_o} \otimes c_{t-1} + b_o) \\ H_t &= o_t \otimes \tanh(c_t) \end{aligned} \quad (9)$$

Besides, the number of parameters in LSTMs is approximately in direct proportion to the square of the link number as equation (10) shows. Here $n(p)_{LSTM}$ refers to the parameters of LSTMs, $n(L)$ is the number of road links.

$$n(p)_{LSTM} \propto (2 \times n(L))^2 \quad (10)$$

with LSTMs, the training phase for traffic speed prediction in large scale road network will be difficult. Studies have proved that overfitting is a serious problem in networks which contains massive parameters. In this case, we use Dropout (proposed by Srivastava *et al.*) to address these problems [41]. Dropout refers to ignoring units randomly during the training phase [41]. Fig. 11 illustrate the mechanism of Dropout. In LSTM, Zaremba's work proved that the dropout-operator will corrupt the information carried by the units when applied in *input gate*. On the contrary, it works well on recurrent connections like H_x [42]. We utilize this conclusion in our work.

E. EXTERNAL FACTORS AND FEATURE FUSION

Apart from spatiotemporal correlations, environmental factors such as road properties, weather, and air quality also have a significant impact on traffic speed. More specifically, it is obvious that drivers tend to drive faster on roads with more lanes for less interference of other vehicles. Meanwhile, one will drive slower in rainy days since the roads are more slippery than usual. High concentration of air pollutant (PM 2.5, PM 10, etc.) will lead to low visibility, which will further cause lower driving speed. Besides, traffic speed of one link will be different for the difference in traffic demand between weekdays and holidays.

In this paper, MLP are used to measure and model the impact of external features. Furthermore, we apply theories in traditional traffic engineering for the feature merge layer. When we estimate the free flow speed of one road section, the environmental effects are always considered as modification value. For an example, in multi-lane road, the free flow speed (FFS) can be calculated as follows [43],

$$FFS = BFFS - f_{LW} - f_{LC} - f_M - f_A \quad (11)$$

where BFFS is a constant which represents the basic free flow speed, while f_{LW} , f_{LC} , f_M , and f_A are modification values of road width, lateral clearance, middle strip type and density of entrances. Motivated by this, we consider the impact of

external factors as modification value of the road speed. We use MLP to model it. Equation (12) shows the mechanism of the data-fusion method. Here \hat{y}_t denotes the predicted traffic speed of all the road links at time step t and $h(t)$ illustrates the output of LSTMs. We concatenate features of road information, weather condition (time t), and air quality (time t) together as environmental features, which is marked as X_t^{w+r+a} in (12). The Date-information is treated as date-feature, which is marked as X_t^d in (12).

F. HDL-NET

This sub-section illustrates the data-preprocessing phase of our work and gives the overview of HDL-net.

1) OUTLIERS

Environmental factors such as weather condition and air quality cannot be ignored in traffic speed prediction. However, fluctuation of traffic speed can be caused by abnormal driving behaviors. Here we consider *overspeed*. The observed speed values beyond speed limitation will be treated as outliers. Equation (13) defines speed-transit function $f(x)$, where l is used to indicate the speed limitation. This formula is built under the assumption that if one can drive under such a high speed beyond speed limitation on one road link, the road condition is likely good enough for him to follow the rules.

$$f(x) = \begin{cases} l & x > l \\ x & x \leq l \end{cases} \quad (12)$$

2) MISSING VALUE

Advanced technology of sensors brings abundant traffic data for researchers. Based on traces of GPS positions, floating-car data (FCD) has been collected and widely utilized in traffic state prediction works [44]. Models always need complete data for better analytical results. However, problems in data collection and transmission always cause data missing problem. Researchers have proposed various methods to solve this problem [45], one of the prevalent methods is linear interpolation, which can be defined as (14) and (15). Here (x_0, a) and (x_1, b) represent two known points and (x_k, y_k) denotes missing value y_k with a known index x_k . It is reasonable to assume that the concentration of air quality indices varies continuously with time passing by. However, it is not suitable for speed data. The recurrent nature of speed data is important in data-filling. We choose the non-null average speed value of link k on the same timestamp in different days for our filling work. Equation (16) illustrates the mechanism,

$$(x_0, a), (x_1, b), (x_k, y_k) s. t. x_0 < x_1, x_k \in (x_0, x_1) \quad (13)$$

$$y_k = a + \frac{b-a}{x_1-x_0}(x_k-x_0) \quad (14)$$

$$\bar{S}_{kt} = \frac{1}{m} \sum_{i=0}^m h(S_{ki}) S_{ki}, i = t-1d, t-2d, \dots$$

$$h(x) = \begin{cases} 0 & x = null \\ 1 & x \neq null \end{cases} \quad (15)$$

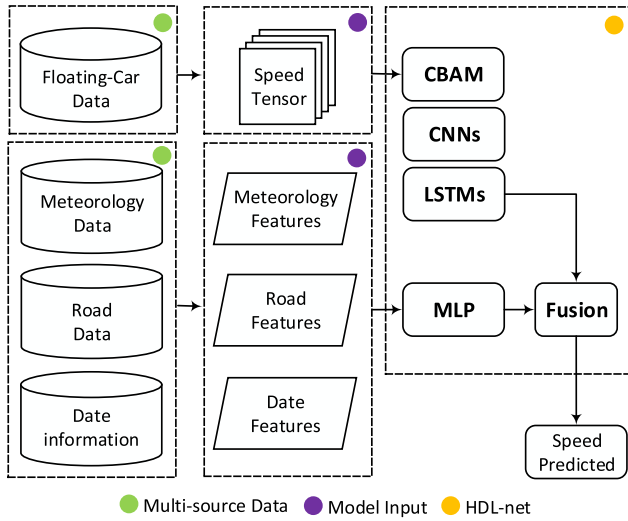


FIGURE 12. Proposed structure.

where \overline{S}_{kt} notes non-null average speed value of link k on time t , $h(x)$ represents judgement function, m refers to the number of non-null speed values, and d means day. There is one specific condition, if for all the days in the existing data, the speed values of link k at one specific timestamp t are missing, we will choose the non-null average speed of the time close to t (e.g., 5, 10, 15 min. . . before) as the inserted value, until we get a complete dataset. This data-filling method is defined as (17). S'_{kt} is the inserted value of link k at time t , d means day and p refers to time interval (5 min, 10 min, etc.)

$$\overline{S'_{kt}} = \frac{1}{n} \sum_{i=0}^n h(S_{ki}) S_{ki}, i = t - 1d - p, t - 2d - p, \dots \quad (16)$$

3) OVERVIEW OF HDL-NET

Fig. 12 illustrates the overall process of our work. The rectangle with yellow mark illustrates the structure of HDL-net. Our work can be divided into 3 steps:

- 1) Data preprocessing. Including removing outliers, filling missing data, turning speed records to speed-tensor, and data normalization. We choose Min-Max normalization to map our input into [0, 1]. Min-Max normalization can be defined as (17). Here x is the initial value, x' is the value after normalization, min and max are maximum and minimum value of the data.

$$x' = \frac{x - min}{max - min} \quad (17)$$

- 2) Train our model with preprocessed data, get training loss based on predicted speed value and the ground truth, then update the parameters of our model.
- 3) Evaluate our model with testing data.

IV. EXPERIMENT AND RESULTS

A. DATA SOURCE AND DESCRIPTION

The floating-car speed data are collected from September. 26, 2019 to October. 7, 2019 in Suzhou. There are about

TABLE 3. Features of road information, weather condition, and air quality.

Road Information	Weather Condition	Air quality
Road length (km)	<i>Weather type</i>	PM _{2.5} (ug/m ³)
Lane number (n/each direction)	<i>Wind direction</i>	PM ₁₀ (ug/m ³)
Road width (m)	Wind level	CO (mg/m ³)
Speed limit (km/h)	Temperature (°C)	NO ₂ (ug/m ³)
<i>Road direction Breadth type</i>		O ₃ (ug/m ³)
<i>Medial strip type</i>		SO ₂ (ug/m ³)
<i>Barrier type</i>		
<i>Road category</i>		

1.4 million observed speed records each day in the road network which encompasses 909 links. These links can be divided into: 489 urban-expressway, 239 primary-arterial, 147 secondary-arterial, and 34 branch-road. There are no intersections in these road links. The weather condition data and air quality data are collected from China National Environmental Monitoring Centre (CNEMC) and National Climate Data Center (NCDC) respectively. The floating car data contains link id, speed value, and timestamp. All the features of road information, weather condition, and air quality data are listed in Table. 3. The time granularity of air quality and weather condition is 1 hour. The categorical features are marked in italic style.

In the following section, two measures of effectiveness are employed to evaluate the performance of the proposed model: mean absolute error (MAE) and absolute percentage error (MAPE), which can be calculated as (18) and (19). Here $y(i, t)$ and $\hat{y}(i, t)$ indicate the ground truth and predicted value of the traffic speed of link i on time t . In these two equations, m is the number of predictions and n refers to the link number of the road network. n_p is equal to $m \times n$.

$$MAE = \frac{1}{n_p} \sum_{i=1}^n \sum_{t=1}^m |y(i, t) - \hat{y}(i, t)| \quad (18)$$

$$MAPE = \frac{1}{n_p} \sum_{i=1}^n \sum_{t=1}^m \frac{|y(i, t) - \hat{y}(i, t)|}{y(i, t)} \quad (19)$$

B. RESULTS AND DISCUSSION

In this section, we will show the results of our HDL-net model and compare them with several deep learning methods including SAEs, LSTMs, 1DCRNs, P-DCRNs, and HDL-n-net.

- **SAEs** [29]: Stack n auto-encoder (AE) together. The output of the k^{th} AE will be the input of the $k + 1^{th}$ AE.
- **LSTMs** [13]: A special RNN which can determine the optimal time lags automatically. The input will be speed-sequence.
- **1DCRN**s [30]: Transform speed value of all the links to 1D vector. Combine 1D CNN and LSTM together to extract spatiotemporal correlations.

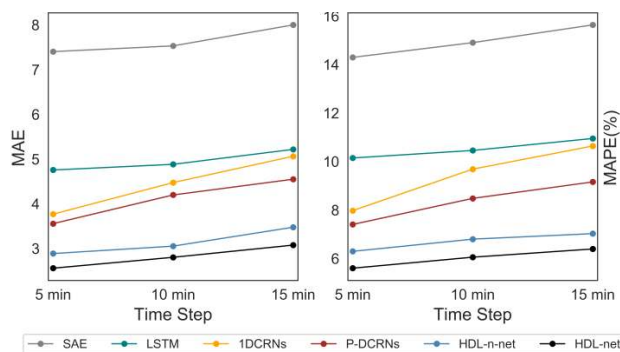


FIGURE 13. Comparison among different methods in terms of short-term speed prediction. MAPE: %.

- **P-DCRNs:** Transform speed value of all the links to 2D matrix. Combine 2D CNN with rectangular filter (3×1) and LSTM together to extract spatiotemporal correlations.
- **HDL-n-net:** Base on P-DCRNs, but take environmental features and date impact into consideration. Utilize MLP to train external features. The mark n indicates that the model does not contain attention mechanism.
- **HDL-net:** HDL-n-net with CBAM. The attention block aims to determine that which channel or pixel of the speed-tensor should be emphasized or suppressed automatically.

The results (MAE and MAPE) of these six different models on testing data are listed in Table. 4 and visually showed in Fig. 13. Fig. 13 illustrates the variation of MAPE and MAE under different time step. For all these 6 deep learning models, the prediction error is smaller with a shorter time interval, which suggests that traffic speed has strongly short-term evolution patterns. As Table 4 shows, as a traditional deep learning method, SAEs fails to extract spatiotemporal correlations. It does not work well in testing data. On average, the relative prediction error (or MAPE) is more than 14% under these 3 Time intervals. With LSTMs we have a great improvement than SAEs, which indicates the remarkable ability of LSTMs in temporal relationship exploring. The 1DCRNs combines 1D CNNs and LSTMs together to extract spatiotemporal correlations of traffic speed and get lower prediction error again.

In order to get a more accurate spatial dependencies, we build P-DCRNs. The result shows that P-DCRNs can effectively mine the spatial relationships based on part of the urban road network structure. The prediction error of is P-DCRNs is lower than models mentioned above. Furthermore, we consider the environmental features and date impact into our model and build a hybrid deep learning structure HDL-n-net. The proposed HDL-n-net outperforms all the marks for different time intervals. Finally, we introduce attention mechanism to enhance the performance of HDL-n-net. With CBAM, the relative prediction error of our model is lower than 6%. Fig. 14 shows the speed observed

TABLE 4. Comparison among different methods in terms of short-term speed prediction. MAPE: %.

Models	Time Interval					
	5min		10min		15min	
	MAE	MAPE	MAE	MAPE	MAE	MAPE
SAEs	7.402	14.295	7.530	14.902	7.999	15.633
LSTMs	4.755	10.146	4.882	10.457	5.216	10.950
1DCRNs	3.768	7.975	4.474	9.686	5.062	10.640
P-DCRNs	3.556	7.405	4.197	8.480	4.549	9.159
HDL-n-net	2.887	6.256	3.053	6.799	3.476	7.027
HDL-net	2.560	5.602	2.803	6.055	3.076	6.394

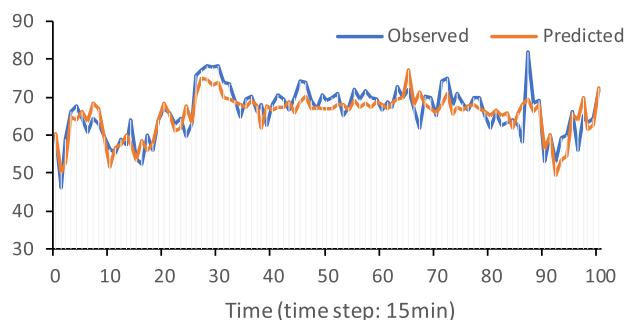


FIGURE 14. Comparison between speed value observed (km/h, without preprocessing) and speed value predicted (km/h) with HDL-net. Time interval: 15min.

and predicted with HDL-net. From LSTM to HDL-net, we improve the model step by step. The error gap from SAEs to LSTMs is greater than any other two model, which proves that temporal correlation is a critical factor in short-term traffic speed prediction. In order to evaluate the performance of our model in different road categories, we choose P-DCRNs (speed data only), HDL-n-net, (with environmental and daily features) and HDL-net (attention mechanism) to show detailing results for each urban road category. The results are shown in Table. 5. The HDL-net which contains attention mechanism CBAM outperforms in every type of urban road under three different time steps. The results of P-DCRNs and HDL-n-net indicate that taking environmental and date features into consideration in traffic speed prediction is important and necessary. Comparing with P-DCRNs, on average, HDL-n-net get 0.718, 1.074, and 1.091 lower on MAE, 1.409, 2.164, 1.091, and 2.271 lower for MAPE. There is less obvious relationship found in the differences of models caused by external factors in different road classes, which indicates that the impact of external factors is fluctuate and uncertain. Besides, a shorter timestep can decrease the prediction error in different road category as well. The average prediction error of different road category is shown in Fig. 15. All the models perform better in urban expressway than other type of roads. In fact, urban expressway is a type of road facility with full access-control, which means there will be little impact

TABLE 5. Comparison of MAPE between P-1DCRNs, HDL-n-net, and HDL-net in different road categories.

Road Category	Model	Time Interval		
		5min	10min	15min
Expressway	P-DCRNs	7.100	8.512	8.827
	HDL-n-net	5.832	6.254	6.609
	HDL-net	5.257	5.772	5.937
Primary Arterial	P-DCRNs	7.754	8.615	9.400
	HDL-n-net	6.055	6.462	7.211
	HDL-net	5.603	5.906	6.405
Secondary Arterial	P-DCRNs	7.757	8.592	9.808
	HDL-n-net	6.422	6.706	7.146
	HDL-net	5.962	6.580	6.654
Branch Road	P-DCRNs	7.818	8.715	9.443
	HDL-n-net	6.104	6.615	7.513
	HDL-net	6.031	6.464	6.495

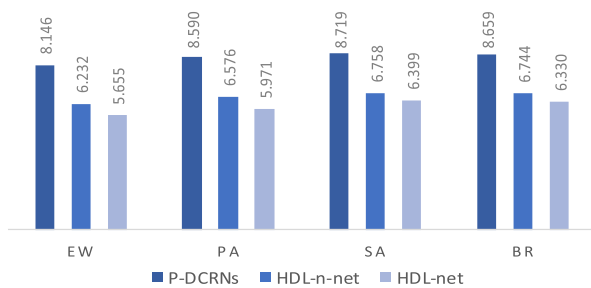


FIGURE 15. Comparison of MAPE among different road categories in terms of short-term speed prediction. EA: Expressway, PA: Primary Arterial, SA: Secondary Arterial, BR: Branch Road.

of pedestrians and non-motor vehicles on urban expressway. On the contrast, pedestrians and non-motor vehicles on the remaining three types road will affect traffic speed a lot.

V. CONCLUSION

In this paper, we proposed a deep learning framework P-DCRNs and improved it step by step. P-DCRNs combines 2D CNNs and LSTMs together to capture the spatiotemporal correlations of traffic speed data. P-DCRNs performed well in short-term speed prediction with time intervals of 5, 10, and 15 minutes. Comparing with traditional SAEs and LSTMs, the prediction error of P-DCRNs on testing data is much lower, which indicates that traffic speed flows a periodical pattern. Furthermore, P-DCRNs works better than 1DCRNs which simply combines 1D CNNs and LSTMs together (0.334 of MAE and 1.09% of MAPE on average). This result emphasizes that the completeness of the road network structure is important in spatial dependency mining. Meteorology condition and air quality are important environmental factors which will affect traveler’s

operation and cause the variation of traffic speed, while date information can affect the traffic demand, to affect traffic speed. We improve P-DCRNs to HDL-n-net, which models the impact of external features as modification values. On testing data, the prediction error of HDL-n-net is lower than P-DCRNs (0.669, 1.144, and 1.073 of MAE; 1.149%, 1.681%, and 2.132% of MAPE). More specifically, among four road categories (urban expressway, primary-arterial, secondary-arterial, and branch-road), HDL-n-net achieves improvements than P-DCRNs (0.864, 1.170, 0.752, and 1.405 in MAE; 1.914, 2.014, 1.961, and 1.915 in MAPE), which proves the importance of external features in speed prediction task. To enhance the performance of our model, we introduced attention mechanism to determine the importance-degree of channels and pixels of our speed-tensor. We build HDL-net with CBAM. The results show that HDL-net outperforms the models mentioned above. In traffic speed prediction, all the models we choose can perform better with a shorter time interval. Besides, P-DCRNs, HDL-n-net, and HDL-net work better on urban expressway, which has full control of all the access. In this case, we conclude that speed prediction will be more accurate under a shorter timestep, as well as roads with less impact of pedestrians and non-motor vehicles.

REFERENCES

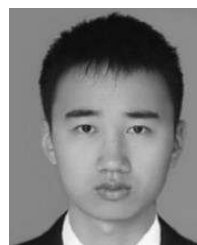
- [1] B. Yao, C. Chen, Q. Cao, L. Jin, M. Zhang, H. Zhu, and B. Yu, “Short-term traffic speed prediction for an urban corridor,” *Comput.-Aided Civil Infrastruct. Eng.*, vol. 32, no. 2, pp. 154–169, 2017.
- [2] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, “Data-driven intelligent transportation systems: A survey,” *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011.
- [3] E.-M. Lee, J.-H. Kim, and W.-S. Yoon, “Traffic speed prediction under weekday, time, and neighboring Links’ speed: Back propagation neural network approach,” in *Proc. Int. Conf. Intell. Comput.* Berlin, Germany: Springer, 2007, pp. 626–635.
- [4] Y. Jia, J. Wu, and Y. Du, “Traffic speed prediction using deep learning method,” in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Rio de Janeiro, Brazil, Nov. 2016, pp. 1217–1222.
- [5] M. T. Asif, J. Dauwels, C. Y. Goh, A. Oran, E. Fathi, M. Xu, M. M. Dhanya, N. Mitrovic, and P. Jaillet, “Spatiotemporal patterns in large scale traffic speed prediction,” *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 794–804, Apr. 2014.
- [6] E. I. Vlahogianni, J. C. Golias, and M. G. Karlaftis, “Short-term traffic forecasting: Overview of objectives and methods,” *Transp. Rev.*, vol. 24, no. 5, pp. 533–557, Sep. 2004.
- [7] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, “Short-term traffic forecasting: Where we are and where we’re going,” *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 3–19, Jun. 2014.
- [8] M. Dougherty, “A review of neural networks applied to transport,” *Transp. Res. C, Emerg. Technol.*, vol. 3, no. 4, pp. 247–260, Aug. 1995.
- [9] Z. Yu, C. Licia, W. Ouri, and Y. Hai, “Urban computing: Concepts, methodologies, and applications,” *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 3, pp. 1–55, Sep. 2014.
- [10] Z. Li, G. Xiong, Y. Chen, Y. Lv, B. Hu, F. Zhu, and F.-Y. Wang, “A hybrid deep learning approach with GCN and LSTM for traffic flow prediction,” in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Auckland, New Zealand, Oct. 2019, pp. 1929–1933.
- [11] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, “DNN-based prediction model for spatio-temporal data,” in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst. (GIS)*, San Francisco, CA, USA, Oct. 2016, pp. 1–4.
- [12] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, “LSTM network: A deep learning approach for short-term traffic forecast,” *IET Intell. Transp. Syst.*, vol. 11, no. 2, pp. 68–75, Mar. 2017.

- [13] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 187–197, May 2015.
- [14] 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, 2017.
- [15] L. Lin, J. Li, F. Chen, J. Ye, and J. Huai, "Road traffic speed prediction: A probabilistic model fusing multi-source data," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1310–1323, Jul. 2018.
- [16] B. L. Smith, B. M. Williams, and R. Keith Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 10, no. 4, pp. 303–321, Aug. 2002.
- [17] M. M. Hamed, H. R. Al-Masaied, and Z. M. B. Said, "Short-term prediction of traffic volume in urban arterials," *J. Transp. Eng.*, vol. 121, no. 3, pp. 249–254, 1995.
- [18] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transp. Res. C, Emerg. Technol.*, vol. 19, no. 4, pp. 606–616, Aug. 2011.
- [19] Q. Y. Ding, X. F. Wang, X. Y. Zhang, and Z. Q. Sun, "Forecasting traffic volume with space-time ARIMA model," *Adv. Mater. Res.*, vols. 156–157, pp. 979–983, Oct. 2010.
- [20] I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through Kalman filtering theory," *Transp. Res. B, Methodol.*, vol. 18, no. 1, pp. 1–11, Feb. 1984.
- [21] M. Chen and S. I. J. Chien, "Dynamic freeway travel-time prediction with probe vehicle data: Link based versus path based," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1768, no. 1, pp. 157–161, Jan. 2001.
- [22] M. Lippi, M. Bertini, and P. Frasconi, "Collective traffic forecasting," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Berlin, Germany: Springer, Sep. 2010, pp. 259–273.
- [23] B. M. Williams, P. K. Durvasula, and D. E. Brown, "Urban freeway traffic flow prediction: Application of seasonal autoregressive integrated moving average and exponential smoothing models," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1644, no. 1, pp. 132–141, Jan. 1998.
- [24] L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dong, "An improved K-nearest neighbor model for short-term traffic flow prediction," *Procedia-Social Behav. Sci.*, vol. 96, pp. 653–662, Nov. 2013.
- [25] S. H. Huang and B. Ran, "An application of neural network on traffic speed prediction under adverse weather condition," Ph.D. dissertation, Dept. Civil Environ. Eng., UW-Madison, Madison, WI, USA, 2003.
- [26] H. Yin, S. C. Wong, J. Xu, and C. K. Wong, "Urban traffic flow prediction using a fuzzy-neural approach," *Transp. Res. C, Emerg. Technol.*, vol. 10, no. 2, pp. 85–98, Apr. 2002.
- [27] J. Tang, F. Liu, Y. Zou, W. Zhang, and Y. Wang, "An improved fuzzy neural network for traffic speed prediction considering periodic characteristic," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2340–2350, Sep. 2017.
- [28] 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.
- [29] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.
- [30] Y. Wu and H. Tan, "Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework," 2016, *arXiv:1612.01022*. [Online]. Available: <http://arxiv.org/abs/1612.01022>
- [31] 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>
- [32] Y. Han, S. Wang, Y. Ren, C. Wang, P. Gao, and G. Chen, "Predicting station-level short-term passenger flow in a citywide metro network using spatiotemporal graph convolutional neural networks," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 6, p. 243, 2019.
- [33] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," 2018, *arXiv:1812.08434*. [Online]. Available: <http://arxiv.org/abs/1812.08434>
- [34] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in *Proc. IEEE Int. Conf. Intell. Secur. Informat. (ISI)*, Jul. 2017, pp. 43–48.
- [35] A. Graves, "Generating sequences with recurrent neural networks," 2013, *arXiv:1308.0850*. [Online]. Available: <http://arxiv.org/abs/1308.0850>
- [36] F. Yu, D. Wei, S. Zhang, and Y. Shao, "3D CNN-based accurate prediction for large-scale traffic flow," in *Proc. 4th Int. Conf. Intell. Transp. Eng. (ICITE)*, Harbin, China, Sep. 2019, pp. 99–103.
- [37] Q. Liu, B. Wang, and Y. Zhu, "Short-term traffic speed forecasting based on attention convolutional neural network for arterials," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 11, pp. 999–1016, 2011.
- [38] 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.*, Atlanta, GA, USA, vol. 33, Jul. 2019, pp. 922–929.
- [39] A. Vaswani, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 5998–6008.
- [40] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, 2018, pp. 3–19.
- [41] 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.
- [42] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," 2014, *arXiv:1409.2329*. [Online]. Available: <http://arxiv.org/abs/1409.2329>
- [43] K. Chen and B. Yan, "Multilane road," in *Road Capacity Analysis*, 1st ed. Beijing, China: CCP, 2003.
- [44] C. de Fabritiis, R. Ragona, and G. Valenti, "Traffic estimation and prediction based on real time floating car data," in *Proc. 11th Int. IEEE Conf. Intell. Transp. Syst.*, Beijing, China, Oct. 2008, pp. 197–203.
- [45] I. Ezzine and L. Benhlila, "A study of handling missing data methods for big data," in *Proc. IEEE 5th Int. Congr. Inf. Sci. Technol. (CIST)*, Marrakesh, Morocco, Oct. 2018, pp. 498–501.
- [46] C. Xu, J. Ji, and P. Liu, "The station-free sharing bike demand forecasting with a deep learning approach and large-scale datasets," *Transp. Res. C, Emerg. Technol.*, vol. 95, pp. 47–60, Oct. 2018.
- [47] Y. Liu, Z. Liu, and R. Jia, "DeepPF: A deep learning based architecture for metro passenger flow prediction," *Transp. Res. C, Emerg. Technol.*, vol. 101, pp. 18–34, Apr. 2019.



XUN YANG is currently pursuing the B.S. degree in transportation engineering with the School of Transportation, Southeast University, Nanjing, China.

His research interests include multi-source heterogeneous data implication, data mining, and deep learning application in traffic prediction.



YU YUAN received the B.S. degree in transportation engineering from the School of Transportation, Southeast University, Nanjing, China. He is currently pursuing the M.S. degree with Southeast University.

His research interests include transportation big data analysis and modeling, and data mining.



ZHIYUAN LIU received the Ph.D. degree in transportation engineering from the National University of Singapore, in 2011.

He is currently a Youth Chief Professor with Southeast University. His research interests include transport network modeling, transport data analytics, public transport, and intelligent transport systems. He has published over 50 journal articles in these areas. He was the awardee of the 1000 Talent Program (Youth Program).