Transportation Research Part C 100 (2019) 372–385

Contents lists available at ScienceDirect

Transportation Research Part C

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

Traffic speed prediction for urban transportation network: A path based deep learning approach

Jiawei Wang[a,b], Ruixiang Chen[a], Zhaocheng He[a,b,*]

[a] Guangdong Provincial Key Laboratory of Intelligent Transportation Systems, Research Center of Intelligent Transportation System, Sun Yat-Sen University, Guangzhou, Guangdong 510006, China
[b] Shenzhen Cyberspace Laboratory, China

ARTICLE INFO

Keywords:
Traffic speed prediction
Urban network
Deep learning
Bidirectional long short-term memory neural network
Model interpretability

ABSTRACT

Traffic prediction, as an important part of intelligent transportation systems, plays a critical role in traffic state monitoring. While many studies accomplished traffic forecasting task with deep learning models, there is still an open issue of exploiting spatial-temporal traffic state features for better prediction performance, and the model interpretability has not been taken serious. In this study, we propose a path based deep learning framework which can produce better traffic speed prediction at a city wide scale, furthermore, the model is both rational and interpretable in the context of urban transportation. Specifically, we divide the road network into critical paths, which is helpful to mine the traffic flow mechanism. Then, each critical path is modeled through the bidirectional long short-term memory neural network (Bi-LSTM NN), and multiple Bi-LSTM layers are stacked to incorporate temporal information. At the stage of traffic prediction, the spatial-temporal features captured from these processes are fed into a fully-connected layer. Finally, results for each path are ensembled for network-wise traffic speed prediction. In the empirical studies, we compare the proposed model with multiple benchmark methods. Under a series of prediction scenarios (i.e., different input and prediction horizons), the superior performance of the proposed framework is validated. Moreover, by analyzing feature from hidden-layer output, the study explains the physical meaning of the hidden feature and illustrate model's interpretability.

1. Introduction

Recent years have witnessed increasing traffic congestion problem in cities, which not only ruins residents' experience on route, but also causes large amounts of economic loss. In order to improve the efficiency of urban road network and the people's living standard, it is urgent to develop intelligent transportation systems, where we need to sense traffic state of road network efficiently for a quick response to different scenario. Moreover, the reliable network state prediction plays an important role in path guidance as well as transportation management, and therefore can promote appropriate assignment on transportation resources and help reduce traffic congestion.

Previous studies on traffic state prediction mainly focused on a single road, where statistical techniques were widely used, including auto-regressive integrated moving average (Williams and Hoel, 2003), kalman filter model (Guo et al., 2014), hidden markov model (Qi and Ishak, 2014) and bayesian interference (Wang et al., 2014), etc. But it should be pointed out that these studies handled a less complex traffic condition and transportation datasets were small in size (Cui et al., 2018). Attempting to deal with high

* Corresponding author.

https://doi.org/10.1016/j.trc.2019.02.002
Received 26 July 2018; Received in revised form 15 January 2019
Available online 08 February 2019
0968-090X/

dimension traffic state data and capture non-linear relationship, many machine learning methods have been adopted, such as artificial neural network (Karlaftis and Vlahogianni, 2011;Chan et al., 2012) and support vector machine (Cong et al., 2016). To take a step further, Ma et al. (2015) utilized long short-term memory network (LSTM NN) to predict traffic speed on the express way, and achieved a better performance compared to the classical methods. Du et al. (2018) made traffic speed prediction on highway with LSTM NN and convolutional neural network (CNN), in which multiple traffic flow parameters were taken into account. However, the mentioned studies lacked consideration on the spatial correlations of traffic state in road network.

With the enrichment of traffic data, researchers begin to study network traffic state prediction, where how to exploit spatial-temporal traffic state feature of road network becomes a hot spot (Min and Wynter, 2011; Ryu et al., 2018; Wu et al., 2018). Asif et al. (2014) leveraged spatial-temporal feature of traffic state through principle component analysis, K-means and self-organizing map, then state prediction was made through support vector regression. Cai et al. (2016) proposed an improved k-nearest neighbor (KNN) method for short-term traffic forecasting, the spatial-temporal feature was taken into account by defining network state matrix. Thanks to the strong ability of feature extraction and non-linear fitting, deep learning has been popular in traffic state prediction recently. Several approaches for exploring spatial-temporal characteristics of road network are proposed: Ma et al. (2017) converted network traffic to images and used CNN for prediction, Yu et al. (2017b) mapped network-wise speed to a grid and established a model with CNN and LSTM NN, besides, Yu et al. (2017a) mined spatial-temporal feature through graph convolution. Furthermore, Wu et al. (2018) predicted traffic flow with CNN, LSTM NN, as well as attention model. In particular, this study presented how the model understood traffic flow data. However, the model still regarded traffic state of each segment at each interval as a pixel, and convolutional process on these pixels could not accurately indicate topology information and traffic flow mechanism, which resulted in deficiency on physical meaning to some extent. In a word, although the mentioned studies achieved satisfying results by considering spatial-temporal feature of traffic state, they put more emphasis on making data adapt to model, rather than constructing a model to match the real world, as a result, these models are usually hard to interpret.

From the review above, it is recognized that spatial-temporal feature is hidden in the road network and the evolution of traffic state, how to understand and exploit these information is crucial in state prediction. To the best of our knowledge, the studies for state prediction are divided into two categories, i.e., classical statistical methods and machine learning (Cai et al., 2016). On one hand, being poor at capturing non-linear relation and dealing with high dimension data, classical statistical techniques cannot leverage spatial-temporal feature comprehensively; on the other hand, though machine learning, especially the deep learning model, performs well on extracting feature for forecasting, most current studies directly consider the entire study area, in this case redundant information may be included and interfere with prediction. More importantly, such a scheme makes them hard to interpretate, leaving the model to be a “black box”. In fact, much endeavor has been made to remove this barrier for the widespread application of deep learning, such as visualizing hidden feature in the image classification task (Erhan et al., 2009; Nguyen et al., 2016), and the techniques to explain individual predictions (Simonyan et al., 2013;Zeiler and Fergus, 2014; Montavon et al., 2017). Whereas in the area of transportation, few studies focused on understanding of deep learning model related to traffic domain knowledge. To fill this gap, mainly three questions need to be solved:

- How to organize network-wise high dimension traffic data to explore spatial-temporal feature of traffic state?
- How to establish an interpretable model utilizing the spatial-temporal feature to accomplish urban network speed prediction?
- How to illustrate the feature which the model has extracted with traffic domain knowledge?

To this end, this paper introduces a path-based deep learning framework for network traffic speed forecasting task, which maps the model to the spatial-temporal structure of the network-wise traffic state. First, the network-wise high dimension data is divided and assigned to the selected critical paths, for the reason that the most frequently used paths tend to share more regular and dominating traffic flow, thus the most useful information can be picked up for speed prediction. Then, bidirectional LSTM neural network (Bi-LSTM NN) (Schuster and Paliwal, 1997) is introduced to model each critical path, following which multiple layers are stacked along the temporal dimension. In this way each model can capture spatial-temporal feature of path for better speed prediction while maintain strong interpretability. At last, results produced by each model are ensembled to give final prediction.

The rest of the paper is organized as follows: In Section 2, the path-based deep learning framework is introduced, including critical path selection and model construction. In Section 3, multiple experiments are conducted to demonstrate the effectiveness of the proposed method as well as illustrate the interpretability. Finally, we draw conclusions for the study and come up with future study directions in Section 4.

2. Methodology

In Fig. 1, a path-based deep learning framework is presented. Given paths composed of a sequence of segments, we suggest that the regular traffic flow tends to appear in the most frequently used paths, which are defined as critical paths. Since it is the traffic flow that determines the traffic state on each segment, we can have a clear insight into spatial-temporal feature among segments by analyzing critical paths. The basic idea is to derive critical paths from historical trajectories, and each critical path is modeled through Bi-LSTM neural network layer, named as Path-LSTM. Then, the Path-LSTM layers are stacked to incorporate temporal information, simulating the traffic state evolution. After establishing models for each critical path, the training processes are launched in parallel. At final forecasting task, inspired by ensemble learning (Zhou, 2012), the trained models make prediction independently, the results of shared segments among critical paths are averaged for a better generalization.

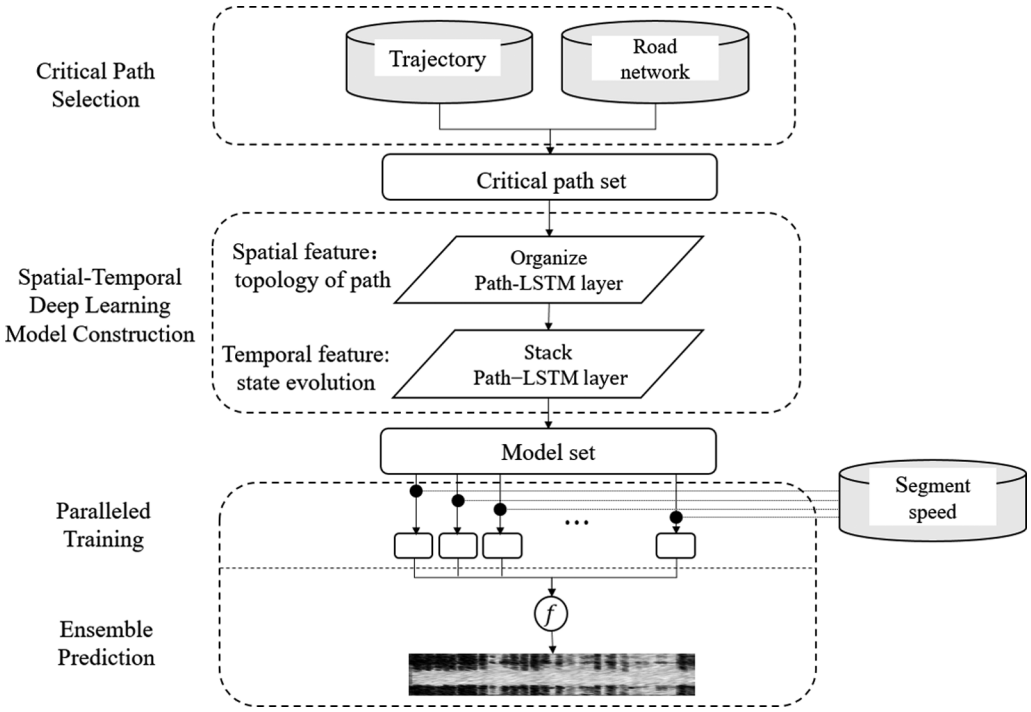


Fig. 1. Path-based deep learning framework for network traffic speed forecasting.

2.1. Critical path selection

In urban road network, the traffic state of a road segment has close relationship with its upstream and downstream segments, and it is quite valuable to be utilized for prediction. (De Fabritiis et al., 2008;Deng and Qu, 2016). For instance, congestion on one segment may be intrigued by its upstream segments, from which there may be massive traffic flow moving downstream. Hence, the future speed can be derived based on such spatial-temporal relationship. Furthermore, by taking a sequence of segments into account, more information can be exploited. As a result, this paper manages to utilize such message by incorporating segments in the view of path, we suggest that the more frequent the path is used, the stronger relation exits among the segments of this path, nevertheless, some segments adjacent to the target segment share less traffic flow, or the pattern of traffic state is irregular, so their information is usually nonsense. Consequently, instead of considering network-wise data directly, traffic data is organized in the view of path. Within the critical path, more precise spatial-temporal feature can be mined for speed prediction while strong interpretability is maintained. To this end, the path occupying major and regular traffic flow is selected, the index r_p is set to facilitate the selection:

$$r_p = \frac{n_p}{|p|} \sum_{s \in p} \frac{n_p}{n_s} \tag{1}$$

where r_p is the measure of regularity and importance for the path p; $|p|$ is the number of segments in p; n_p and n_s stand for the counts of path p and segment s in historical trajectories dataset, respectively; then a larger $\frac{1}{|p|}\sum_{s\in p}\frac{n_p}{n_s}$ implies that the segments of p are less likely to appear on any other paths, so the chosen path is more exclusive to its segments, in other words, the segments of the critical path are more likely to appear at the same time, thus presenting a strong relationship. According to the definition, critical paths are selected in the descending order of r_p, this procedure will last until the segments from the selected paths have covered the whole road network.

2.2. Spatial-temporal deep learning model construction

2.2.1. Bidirectional LSTM neural network

The LSTM NN (Hochreiter and Schmidhuber, 1997) is a kind of recurrent neural network with long short-term memory (LSTM) cells as building blocks for its hidden layers. As shown in Fig. 2, two LSTM NN is connected to construct Bi-LSTM NN (Schuster and Paliwal, 1997), making it capable to deal with sequence data in both forward and backward directions. Notably, LSTM cell is the core of LSTM NN, which helps overcome gradient explosion and vanish problem (Hochreiter, 1991), its structure is shown in Fig. 3.

Considering the scheme of LSTM cell, it can be presented according to the following equations:

$$f_t = \sigma(W_f\,[h_{t-1}, x_t]^T + b_f) \tag{2}$$

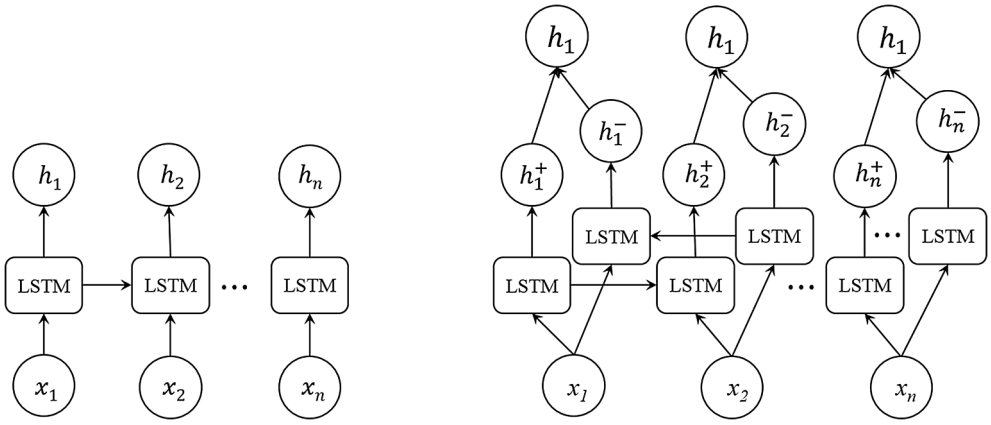


Fig. 2. Structure diagram for LSTM NN (left) and Bi-LSTM NN (right).

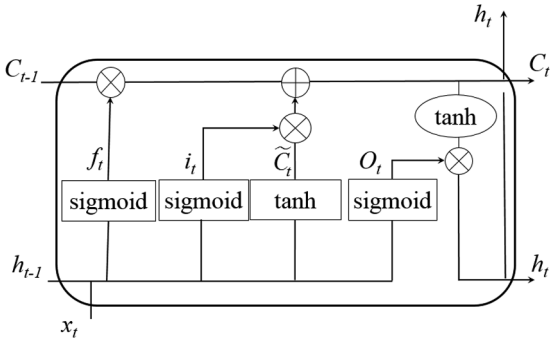


Fig. 3. Structure diagram for LSTM cell.

$$i_t = \sigma(W_i\,[h_{t-1}, x_t]^T + b_i) \tag{3}$$

$$\widetilde{C}_t = \tanh(W_C\,[h_{t-1}, x_t]^T + b_C) \tag{4}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C}_t \tag{5}$$

$$o_t = \sigma(W_o\,[h_{t-1}, x_t]^T + b_o) \tag{6}$$

$$h_t = o_t \odot \tanh(C_t) \tag{7}$$

where $\odot$ is the scalar product between two vectors; $\sigma(\cdot)$ and $\tanh(\cdot)$ represent two kinds of activation function; W_f, W_i, W_C and W_o stand for the weight matrix of input and gates in the cell, respectively; b_f, b_i, b_C and b_o correspond to the bias vectors. h represents the output of the cell, and C is introduced as cell state, which is utilized to store information from previous cell. Specifically, f_t is a factor to determine how much effect the last state C_{t-1} will have on the current cell, likewise, indicates how much we care about the current candidate state $\widetilde{C}_t$, notably, these three elements are all calculated based on the input xt at current time and ht-1, the output from last cell. Moreover, the updated state C_t on the current cell is derived from $f_t \odot C_{t-1}$ and $i_t \odot \widetilde{C}_t$. With C_t and addition adjustment o_t, we can calculate output h_t for the current cell. As for Bi-LSTM NN, similar to LSTM NN, with Eqs. (2)–(7) we can calculate the output of its forward and backward network, respectively. The forward network output is denoted as $[h_1^+, h_2^+, \cdots h_n^+]$ while the backward network output is $[h_1^-, h_2^-, \cdots h_n^-]$, then corresponding elements of these two output are concatenated to produce the final result:

$$\boldsymbol{h_n} = [h_n^-, h_n^+] \tag{8}$$

2.2.2. Model construction

Since the path is a set of segments, we suggest that the traffic state of the path is attributed to the state of each segment. With this regard, different models need to be established to mine spatial-temporal traffic state feature for different paths. According to the previous studies (Ma et al., 2015;Fu et al., 2016;Zhao et al., 2017;Tian and Pan, 2015;Duan et al., 2016;Yu et al., 2017b;Cui et al., 2018), LSTM NN or Bi-LSTM NN is widely used to set up a model along the temporal dimension. However, they may omit the fact that, it is the traffic wave that promotes state variation of road segments. With the motivation to capture such forward and backward information, Bi-LSTM NN is adopted to model the path, as shown in Fig. 4.

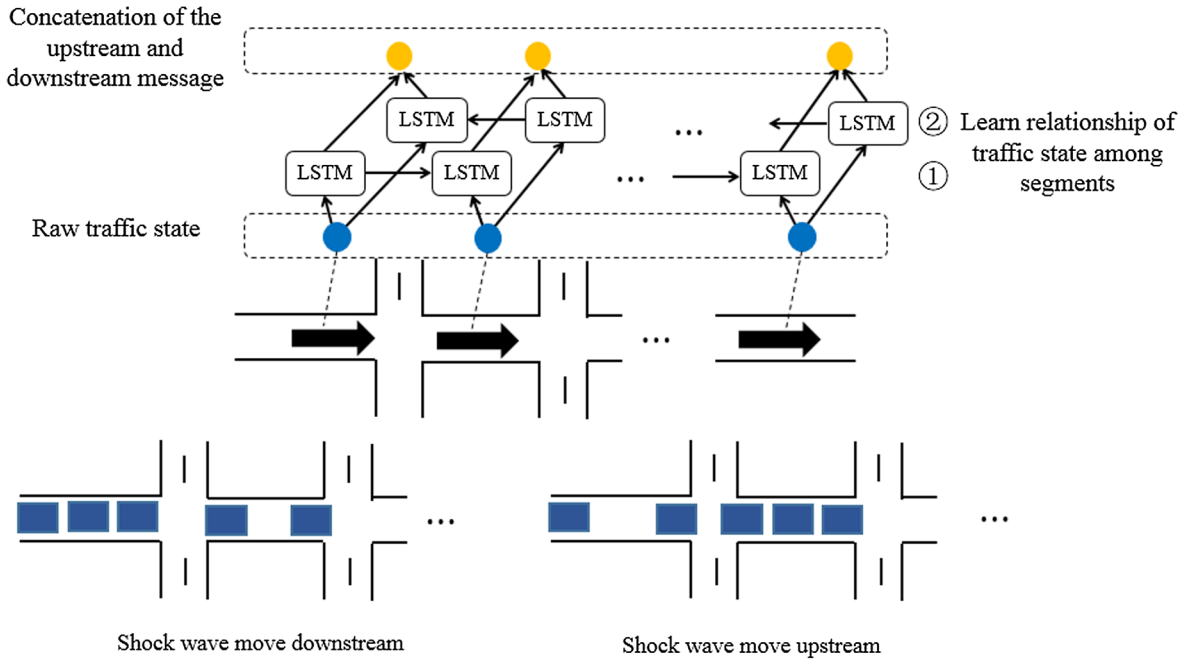


Fig. 4. Model path with Bi-LSTM NN.

The Bi-LSTM NN layer used to model the path is named as Path-LSTM, where LSTM cells of the forward network ① and backward network ② correspond to the segments along the path. For the forward network ①, its message flow is in accordance with the actual direction of the traffic flow, aiming to simulate traffic wave moving downstream. With regard to the backward network ②, its message flow is opposed to the actual direction of traffic flow, indicating that the traffic wave can pass upstream. Based on this architecture, the traffic state feature among segments can be incorporated comprehensively. In addition, since traffic state has strong correlation from time to time, it is necessary to assimilate temporal information for reliable forecasting. To this end, we introduce the framework of forward neural network, with Path-LSTM as the hidden layer, and the number of layers equals the previous available time steps. By stacking multiple Path-LSTM layers, we simulate the state evolution of the path along temporal dimension, as shown in Fig. 5.

Given the speed of segments in a critical path p for last k time steps, $\boldsymbol{V}_{\boldsymbol{T-i}}^{\boldsymbol{p}} = [v_{T-i}^{1}, v_{T-i}^{2}, \cdots v_{T-i}^{|p|}]$, $i = 0, 1, \cdots k-1$, where T is the current interval and $|p|$ indicates the number of segments in p. It should be noted that the order of elements $\boldsymbol{V}_{\boldsymbol{T-i}}^{\boldsymbol{p}}$ in is in accordance with the direction of traffic flow, thus, $\boldsymbol{V}_{\boldsymbol{T-i}}^{\boldsymbol{p}}$ represents speed sequence along spatial dimension. Model shown in Fig. 5 aims to predict

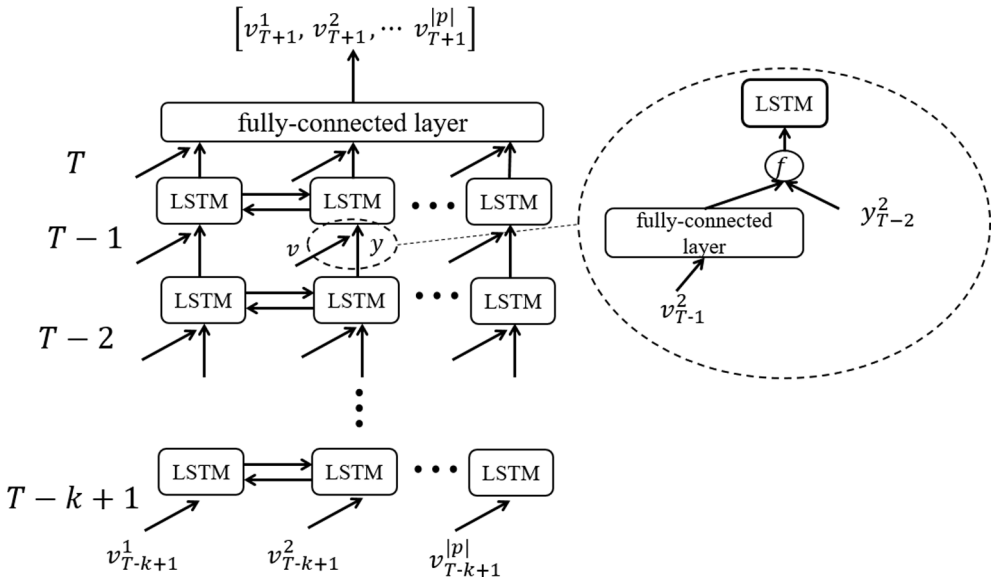


Fig. 5. Overall structure of the path-based deep learning model.

speed $[v_{T+1}^1, v_{T+1}^2, \cdots v_{T+1}^{|p|}]$ at next time step.

For the first Path-LSTM layer $L_j, j = 0$, its input is the speed vector at $T-k+1+j$, denoted as $\boldsymbol{V}_{\boldsymbol{T-k+1+j}}^{\boldsymbol{p}} = [v_{T-k+1+j}^1, v_{T-k+1+j}^2, \cdots v_{T-k+1+j}^{|p|}]$. Here we take the s^{th} segment in path p as an example to illustrate the feature extraction procedure:

$$x_{T-k+1+j}^s = \text{elu}(W_{T-k+1+j}^x v_{T-k+1+j}^s + b_{T-k+1+j}^x) \tag{9}$$

$$h_s^+ = f_{T-k+1+j}^+(x_{T-k+1+j}^s, c_{s-1}^+) \tag{10}$$

$$h_s^- = f_{T-k+1+j}^-(x_{T-k+1+j}^s, c_{s-1}^-) \tag{11}$$

$$y_{T-k+1+j}^s = \text{elu}(W_{T-k+1+j}^y [h_s^+, h_s^-]^T + b_{T-k+1+j}^y) \tag{12}$$

where elu (Clevert et al., 2015) denotes activate function; $f_{T-k+1+j}^+(\cdot)$, $f_{T-k+1+j}^-(\cdot)$ represents the calculating processes in forward network and backward network for LSTM cells at current layer, respectively, which is shown in detail at Eqs. (2)–(7); $W_{T-k+1+j}^x$ and $W_{T-k+1+j}^y$ are weights for the input and output of the cell, $b_{T-k+1+j}^x$ and $b_{T-k+1+j}^y$ are the corresponding biases. At the beginning, the speed of segments is mapped to high dimension space with Eq. (9), producing the input feature x_s; the upstream information for s is denoted as h_s^+, which is determined by $x_{T-k+1+j}^s$ and the upstream traffic state c_{s-1}^+; downstream information h_s^- for segment s can be obtained similarly. Through non-linear map after concatenating h_s^+ and h_s^-, we get output $y_{T-k+1+j}^s$, which can be regarded as the traffic sate feature of the whole path reflected on segment s.

For the subsequent Path-LSTM layers $L_j, j = 1, 2\cdots k-1$, in order to incorporate information along temporal dimension, the traffic states from last interval and current speed are combined, as show in Eq. (14), thus the input in this layer consists of speed vector at $T-k+1+j$ and the output from last layer:

$$\widetilde{x}_{T-k+1+j}^s = \text{elu}(W_{T-k+1+j}^x v_{T-k+1+j}^s + b_{T-k+1+j}^x) \tag{13}$$

$$y_{T-k+1+j}^s = \text{elu}(W_{T-k+1+j}^x \left[\widetilde{x}_{T-k+1+j}^s, y_{T-k+1+j-1}^s\right]^T + b_{T-k+1+j}^y) \tag{14}$$

$$h_s^+ = f_{T-k+1+j}^+(x_{T-k+1+j}^s, c_{s-1}^+) \tag{15}$$

$$h_s^- = f_{T-k+1+j}^-(x_{T-k+1+j}^s, c_{s-1}^-) \tag{16}$$

$$y_{T-k+1+j}^s = \text{elu}(W_{T-k+1+j}^y [h_s^+, h_s^-]^T + b_{T-k+1+j}^y) \tag{17}$$

Through stacking Path-LSTM layers, the temporal feature is incorporated, in this way we assimilate temporal information and simulate the evolution of traffic state. At the stage of prediction, the captured feature will be input to a fully-connected layer, and speed is predicted as follows:

$$[v_{T+1}^1, v_{T+1}^2, \cdots v_{T+1}^{|p|}] = W_p [x_T^1, x_T^2, \cdots x_T^{|p|}] + b_p \tag{18}$$

where W_p and b_p are weights and biases for the fully connected layer. As shown in Eq. (18), the prediction for each segment is determined by the spatial-temporal feature $[x_T^1, x_T^2, \cdots x_T^{|p|}]$, which comes from layer-wise extraction on the traffic state of the path. Naturally, each feature x_T^s, $s = 1, 2\cdots, |p|$ indicates the contribution of segment s to the forecasting task.

2.2.3. Paralleled training and ensemble prediction

During the speed forecasting task, we train each model to predict speed on segments along the corresponding path. The mean squared error (MSE) between predictions and ground-truth traffic speeds will be minimized during training, parameters of the model is optimized based on Adam optimization (Kingma and Ba, 2014), a kind of stochastic gradient descent method. The objective function is as follows:

$$\theta_p = \underset{\theta_p}{\text{argmin}} \frac{1}{NH|p|} \sum_{i=1}^{N} \sum_{t}^{H} \sum_{s \in p}^{|p|} (v_{T+t}^{(i,s)} - \widetilde{v}_{T+t}^{(i,s)})^2 \tag{19}$$

where θ_p indicates the trained parameters set for model p; and the training goal is the MSE of N training samples, each includes $|p|$ segments and H prediction steps, in other words, we have $H|p|$ predictions to make for each sample. After training for each model, we ensemble them to predict the network-wise speed for future time steps:

$$\bar{v}_{T+t}^s = \frac{1}{|V_{T+t}^s|} \sum_{v_{T+t}^s \in V_{T+t}^s} v_{T+t}^s,\ V_{T+t}^s = \{v_{T+t}^s | s \in p_i,\ i = 1, 2\cdots |P|\} \tag{20}$$

where $\bar{v}_{T+t}^s$ is the final prediction of segments s for the future time step $T+t$ and V_{T+t}^s is the set of all candidate prediction v_{T+t}^s of segment s for $T+t$. $|P|$ stands for the number of critical paths.

Fig. 6. Location of AVI detectors in Xuancheng (left) and road network for experiment (right).

3. Case study

3.1. Data description

The data for case study is collected from automatic vehicle identification (AVI) detectors in the central district of Xuancheng, China. The coverage of AVI detectors exceeds 80%, which is qualified for traffic speed collection and trajectory inference. The study area is a road network consists of 112 road segments, as shown in Fig. 6. The speed data is collected for nearly 3 months, from January 23, 2018 to April 22, 2018, and the time interval is 5 min. In this study, the dataset is mainly divided into two parts. The first part from January 23, 2018 to April 15, 2018, is used to train the models, of which 80% is selected randomly to create the training set, and the remaining is used for validation. The second part from April 16, 2018 to April 22, 2018, is the test set. In addition, we collect trajectories from February 18, 2018 to March 26, 2018 for critical path selection.

3.2. Critical path selection

Before establishment of the model, critical path selection is conducted to divide the road network. Trajectories obtained from AVI detectors in Xuancheng are used to produce the historical path set, which includes 9,535,603 paths. According to the selection procedure in Section 2, 52 critical paths are selected for the study area, in other words, 52 spatial-temporal deep learning models will be set in this study. To illustrate rationality of the selected critical paths, we conduct an analysis on Pearson correlation coefficient (PCC) (Guyon, 2003) between different segments' speed in the training dataset. PCC can measure the strength of linear correlation between two variables, thus we use this index to indicate whether traffic speed in two segment has significant linear relationship.

As shown in Fig. 7, for each segment in the road network, there are 4 possible relationship between this target segment and its strongest correlated segment: "Belong to the same critical path and adjacent", "Only belong to the same critical path", "Only adjacent" and "Others". Fig. 7 is used to present frequency for each relationship. Intuitively, nearly 60% segments' most correlated segment is located adjacently in the same critical path, moreover, nearly 80% of them belong to the same critical path. What should be noted is that we are not aimed to find the most correlated segments, because there may be two strongly correlated segments far apart from each other, and their traffic state has no causality at all. Instead we consider enough correlation among segments in the critical path to be an evidence, with this we draw to a conclusion that the critical paths are qualified for further hidden spatial-temporal feature exploitation with deep learning. Another validation is presented through performance analysis in Section 3.

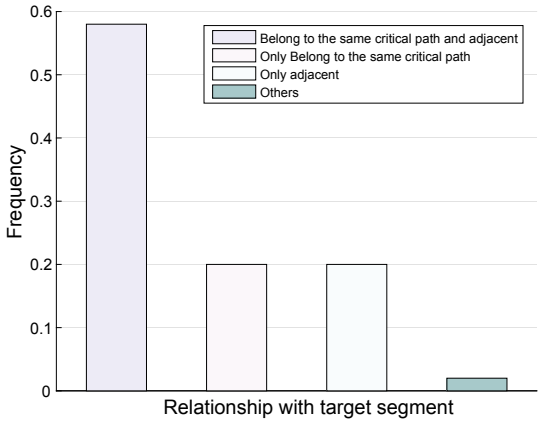


Fig. 7. Frequency of the relationship between target segments and their strongest correlated segment.

Table 1
Experiments settings.

Input time span	Prediction time span	Models for comparison
15-min	5-min	
15-min	15-min	PBDL-CP
15-min	30-min	PBDL-RP
30-min	5-min	PBDL-CP(-)
30-min	15-min	CNN
30-min	30-min	LSTM NN
45-min	5-min	ANN
45-min	15-min	KNN
45-min	30-min	

3.3. Performance analysis

3.3.1. Experiments setting

In this subsection, prediction performance will be evaluated by MSE, the relevant formula is shown as follows:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{21}$$

where n is the number of all predictions, y_i is the ground-truth speed and $\hat{y}_i$ denotes the corresponding prediction.

As shown in Table1, with the aim to test the performance of the proposed model, nine prediction scenarios are set and multiple prevailing models are chosen for comparison. The input time spans 15 min with prediction time spans 5 min means that, we will predict network speed in 5 min (i.e., 1 time step) given previous 15 min (i.e., 3 time steps) speed. As for the benchmark models, CNN (Ma et al., 2017) takes spatial-temporal traffic state matrix of the road network as input, which is an instance of considering the whole network directly for state prediction; LSTM NN (Ma et al., 2015) works well in learning temporal feature for sequence data, which is a typical model treating prediction along the temporal dimension; and traditional neural network (ANN) is adopted to evaluate the performance of simple deep learning model. In addition, KNN is selected for a comparison between deep learning and classical machine learning technique. In addition, to identify the effect of critical path selection, another model is constructed based on randomly selected paths, of which the number is kept as the same as critical paths. The path based deep learning framework for critical paths is abbreviated to PBDL-CP, and the framework based on random selected paths is named as PBDL-RP. Besides, we also establish a model with LSTM NN under the proposed framework considering critical paths, named as PBDL-CP(-), thus the significance of Bi-LSTM NN in the model can be validated.

The structure of PBDL-CP and PBDL-RP includes: (Input time spans/5) × Path-LSTM layers, where the size of hidden neurons in LSTM cell is 32; (Input time spans/5) × fully-connected layers mapping speed to high dimension space, with size (1, 32); 1 × fully-connected layer for final prediction, where the size is determined by the input and prediction time spans. Additionally, the learning rate is set as 0.0006 and the batch size is 32. As for PBDL-CP(-), the structure is the similar to PBDL-CP, the size of hidden neurons in LSTM cell is 64.

The structure and hyper-parameters of other models are initially set according to the experts experience (Cai et al., 2016;Ma et al., 2017), and the final setting is determined based on the five- fold cross validation. Specifically, the structure of CNN includes: convolution layer with filter whose dimensions are (3, 3, 64), max-polling layer with downsample window whose size is (2, 2), fully-connected layer with 1200 hidden units. The structure of LSTM NN includes: one input layer, one LSTM layer and one output layer, the LSTM cell is adjusted to contain 512 hidden units. ANN is optimized to consist of 2 hidden layers with 400 hidden units for each layer. KNN is configured to use the 5 nearest points. Additionally, early stopping scheme is adopted for deep learning methods to avoid overfitting.

3.3.2. Prediction performance

First, the overall evaluation is conducted by comparing the MSE of network-wise prediction among the models in different scenarios. According to Tables 2–4, MSE of each model is presented, and the relative error increment to the proposed model is calculated for the benchmark models. We can observe that PBDL-CP shows the best prediction performance. Furthermore, within all prediction scenarios the average MSE is decreased by 35.40%, 21.09%, 9.92%, and 16.90% for KNN, ANN, CNN and LSTM NN, respectively. The main reason is that, for CNN, as it treats input as image to incorporate the spatial-temporal traffic state, this information is beneficial to prediction to some extent, nevertheless, the consideration of whole network may introduce disturbing feature, especially for coarse-grained speed data (i.e., 5 min), in this case the speed predictions for target segment are interfered by other less correlated segments. For LSTM NN, since it has not taken measures to utilize spatial feature among segments, the prediction precision is affected, especially for short input time spans. In addition, PBDL-CP(-) and PBDL-RP can produce comparative result compared to PBDL-CP in some scenarios, thus it indicates that the path-based framework or critical path selection does help improve prediction performance, however, by introducing both two scheme, we can get better result especially for longer prediction horizon.

Besides, in order to figure out how the performances vary over time, the MSE of network-wise prediction for different models is plotted from 6:00–22:00 in April 16, 2018. The Figs. 8and 9 present the performances of short-term prediction (i.e., 5 min) and long-

Table 2
Network-wise MSE of different models with 15-min input time span.

Model	Prediction time spans		
	5-min	15-min	30-min
PBDL-CP	**7.32**	**13.92**	**15.69**
PBDL-RP	7.41(+1.2%)	15.27(+9.7%)	18.06(+15.1%)
PBDL-CP(-)	7.36(+0.5%)	15.60(+12.0%)	17.99(+14.7%)
CNN	7.86(+7.3%)	14.51(+4.2%)	17.58(+12.0%)
LSTM NN	10.51(+43.5%)	15.83(+13.7%)	17.81(+13.5%)
ANN	10.83(+47.9%)	16.20(+16.3%)	17.66(+12.5%)
KNN	16.90(+130.0%)	17.58(+26.3%)	20.20(+28.7%)

The bold font values indicates the lowest prediction errors.

Table 3
Network-wise prediction MSE of different models with 30-min input time span.

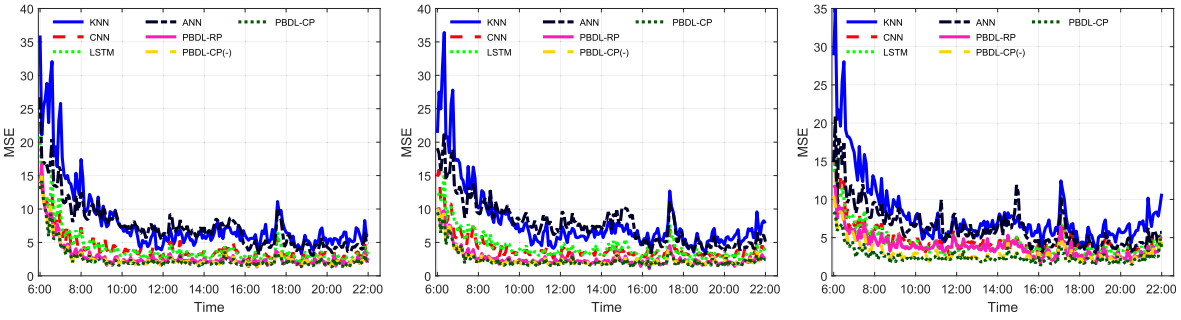
Model	Prediction time spans		
	5-min	15-min	30-min
PBDL-CP	**6.96**	**13.60**	**15.78**
PBDL-RP	8.21(+17.9%)	15.11(+11.1%)	18.50(+17.2%)
PBDL-CP(-)	6.97(+0.01%)	14.55(+6.9%)	16.16(+2.4%)
CNN	8.71(+25.1%)	13.91(+2.2%)	16.04(+1.6%)
LSTM NN	10.14(+45.6%)	16.13(+18.6%)	16.21(+2.7%)
ANN	12.90(+85.3%)	16.31(+19.9%)	17.37(+10.8%)
KNN	15.12(+117.2%)	17.63(+29.6%)	19.16(+21.4%)

The bold font values indicates the lowest prediction errors.

Table 4
Network-wise prediction MSE of different models with 45-min input time span.

Model	Prediction time spans		
	5-min	15-min	30-min
PBDL-CP	**7.03**	**12.81**	**14.45**
PBDL-RP	8.87(+26.1%)	15.51(+21.0%)	19.11(+32.2%)
PBDL-CP(-)	8.64(+22.9%)	14.10(+10.1%)	15.38(+6.4%)
CNN	10.08(+43.4%)	13.73(+7.2%)	15.48(+7.1%)
LSTM NN	10.33(+46.9%)	14.12(+10.2%)	15.65(+8.3%)
ANN	11.99(+70.5%)	14.38(+12.2%)	15.72(+8.7%)
KNN	17.19(+144.5%)	19.90(+55.3%)	20.51(+41.9%)

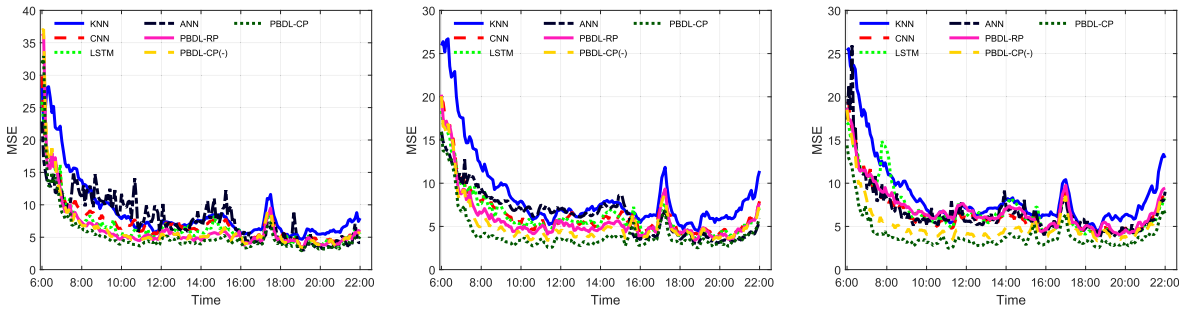
The bold font values indicates the lowest prediction errors.



(a) 15-min input time spans. (b) 30-min input time spans. (c) 45-min input time spans.

Fig. 8. Network-wise MSE comparison from 6:00 to 22:00 in April 16, 2018 for short-term prediction (i.e., 5 min).

term prediction (i.e., 30 min), respectively. For short input time spans (i.e., 15 min), the performances of all models are close, because there is limited feature to exploit, and the short-term prediction is quite easy for these models. With longer input time spans and prediction time spans, the superiority of deep learning model is more clear with the powerful ability to utilize hidden feature and fit



(a) 15-min input time spans. (b) 30-min input time spans. (c) 45-min input time spans.

Fig. 9. Network-wise MSE comparison from 6:00 to 22:00 in April 16, 2018 for long-term prediction (i.e., 30 min).

non-linear relationship. In particular, PBDL-CP yields smaller MSE at most time.

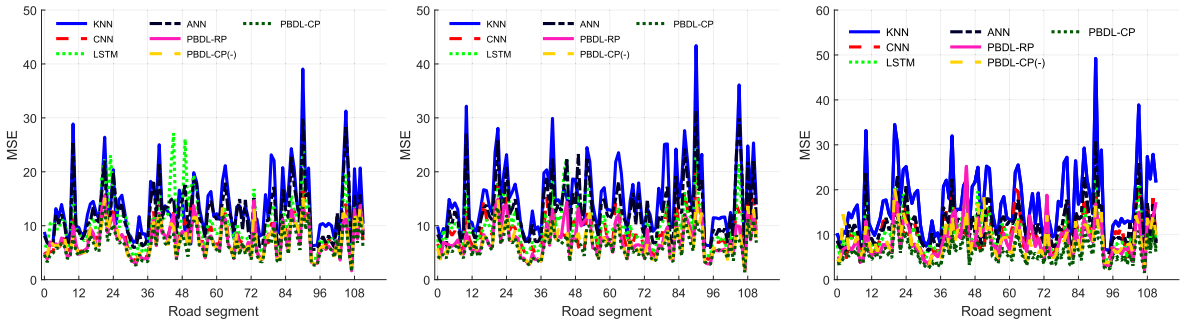
Moreover, for urban traffic speed forecasting task, it is necessary to observe prediction performances of each segment in the road network. The results are also presented under different input time and prediction time spans. MSE for 112 segments on validation dataset is shown in Figs. 10 and 11. For a large portion of segments, the best prediction performances come from the proposed framework. Specifically, for short-term prediction given different input time spans, the proposed framework has a better prediction on average 86% segments, for long-term prediction, the percentage is nearly 70%. These results validate the applicability of the proposed model for urban speed prediction.

3.3.3. Model efficiency

Training deep learning model is a time-consuming work. In most cases, an easily optimized deep learning model is not only beneficial to practical application, but also indicates its better capability to exploit feature through data. As a result, it is important to analyze the training efficiency among different deep learning models. In this study, all the deep learning models are implemented on Tensorflow and experiments are performed by a PC Server (Intel(R) Xeon(R) CPU E5-2630 2.4GHZ, memory 128 GB). As show in Fig. 12, where one epoch indicates one-round training on the whole training dataset. Through comparison, the proposed model yields the best training efficiency, while CNN is the most difficult to train. The reason is that, the proposed models divide the road network into critical paths and are trained in parallel, thus the feature to consider is much leaner. On the contrary, as CNN directly considers the spatial-temporal feature of whole network, such large amount of information plus convolutional process results in more expensive cost.

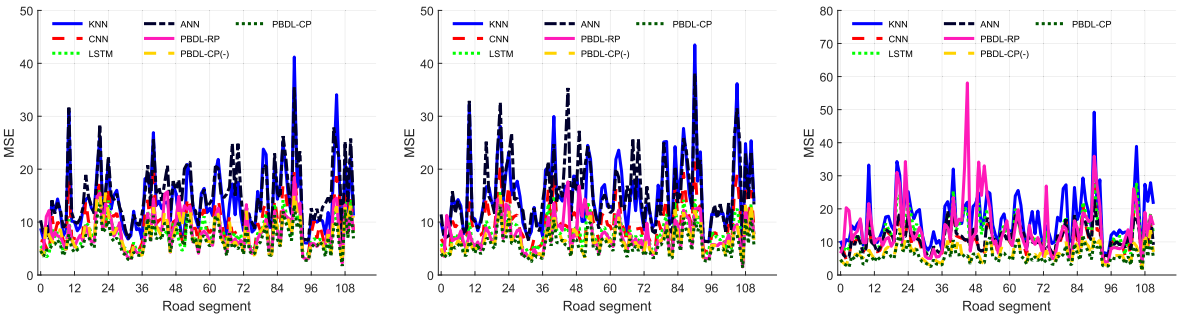
3.4. Model interpretation

Interpretation is the process of giving explanations to human (Kim and Doshi-Velez, 2017). However since the output of hidden layer in the neural network is always hard to understand, interpretability still remains as a critical problem in deep learning (Bau et al., 2017), especially for domain applications. As a result, in the previous studies on forecasting traffic state with deep learning, the models are regarded as a “black box”. To meet this challenge, this subsection will try to demonstrate the spatial-emporal interpretability of the proposed model by analyzing the output of the Path-LSTM layer. For the sake of simplicity, we will focus on the scenario where the input time spans is 45 min and the prediction time spans is 5 min, and the data for illustration is the network speed in April 17, 2018. Additionally, we set the output dimension of Path-LSTM layer to 1 for explicit visualization.



(a) 15-min input time spans. (b) 30-min input time spans. (c) 45-min input time spans.

Fig. 10. MSE comparison among all segments on validation data set for short-term prediction (i.e., 5 min).



(a) 15-min input time spans. (b) 30-min input time spans. (c) 45-min input time spans.

Fig. 11. MSE comparison among all segments on validation data set for long-term prediction (i.e., 30 min).

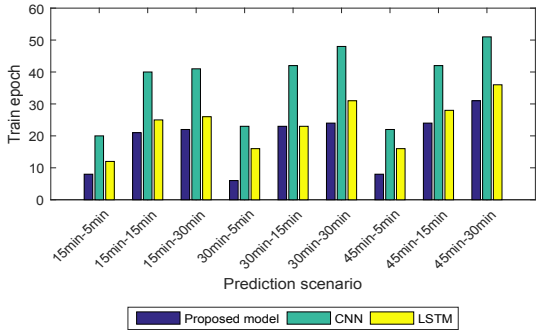


Fig. 12. Average training epoch comparison under different prediction scenarios.

3.4.1. *Temporal feature interpretation*

As mentioned before, the proposed model stacks multiple Path-LSTM layers to simulate the traffic state evolution and assimilate the temporal information. Analyzing what the output of Path-LSTM represents will help demonstrate model's interpretability. Specifically, since the temporal information is abstracted layer by layer, the output of last Path-LSTM layer can be a summation of all the temporal feature, as a result, its visualization is more important and straightforward for illustration.

As shown in Fig. 13(left), one of the critical path is used as an example. To construct input for the model, we divide the speed data for this path according to the interval: 0:00–0:45, 0:05–0:50, ⋯, 23:10–23:55. Then, for each input, the output of last Path-LSTM layer is collected, which can be regarded as a feature vector, as shown in Fig. 13(right). Finally, the feature vectors are stacked to form a feature matrix. Specifically, feature matrix from the model before and after training is denoted as $\boldsymbol{F}$ and $\widetilde{\boldsymbol{F}}$. For comparison, all the speed data of this path in April 17, 2018 is mapped into a spatial-temporal traffic state matrix $\boldsymbol{S}$. The visualizations of matrix $\boldsymbol{S}$, $\boldsymbol{F}$ and $\widetilde{\boldsymbol{F}}$ is shown in Fig. 14.

Where the index of horizontal axis indicates the tag of segments along the selected path, and the vertical axis shows time spot for

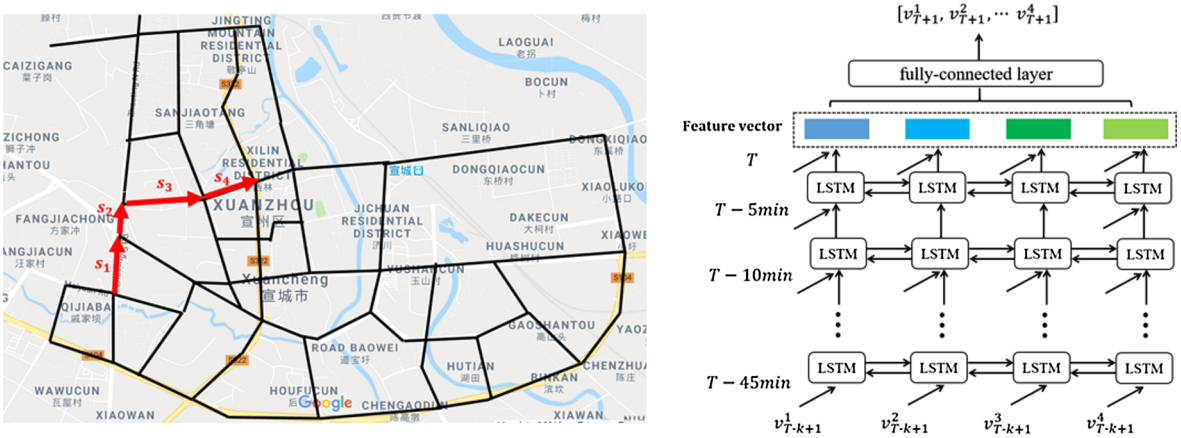
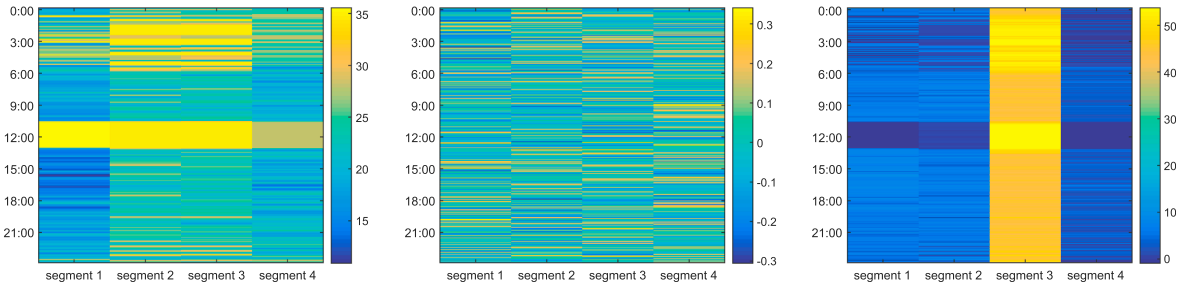


Fig. 13. Example of critical path (left) and corresponding feature vector to visualize (right).



(a) Ground-truth traffic state matrix. (b) Feature matrix before training. (c) Feature matrix after training.

Fig. 14. Visualization of the traffic state matrix and corresponding feature matrices.

April 17, 2018. As shown in Fig. 14, the visualization of traffic state matrix implies that the traffic state of each segment varies along temporal dimension, however, the feature matrix before training makes no sense with its noise-like visualization. By contrast, feature matrix after training exhibits a strong temporal discrimination similar to the state matrix. Through this comparison, it is rational to state that the model has learned to accumulate and abstracted the temporal feature for prediction.

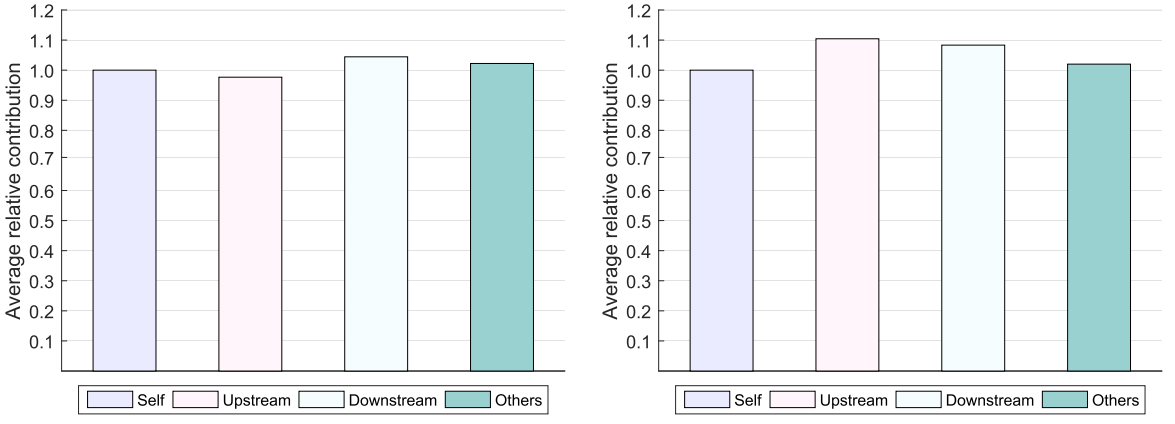
3.4.2. Spatial feature interpretation

With regard to the spatial feature, since we have mapped the Path-LSTM layer to critical path, the relation among its segments can be reflected through the connection among LSTM cells, thus the explanation can be given mainly by analyzing the influence from each LSTM cell on the prediction. From Eq. (18) in Section 2, for the s^{th} segment in critical path p, its prediction at $T + t$ can be treated as the linear combination of the output from the last Path-LSTM layer:

$$v_{T+t}^{s} = \sum_{j=0}^{|p|} W_{fc}(j, s + (t-1)|p|) \cdot x_j = \sum_{j=0}^{|p|} q_{t,j}^{s} \tag{22}$$

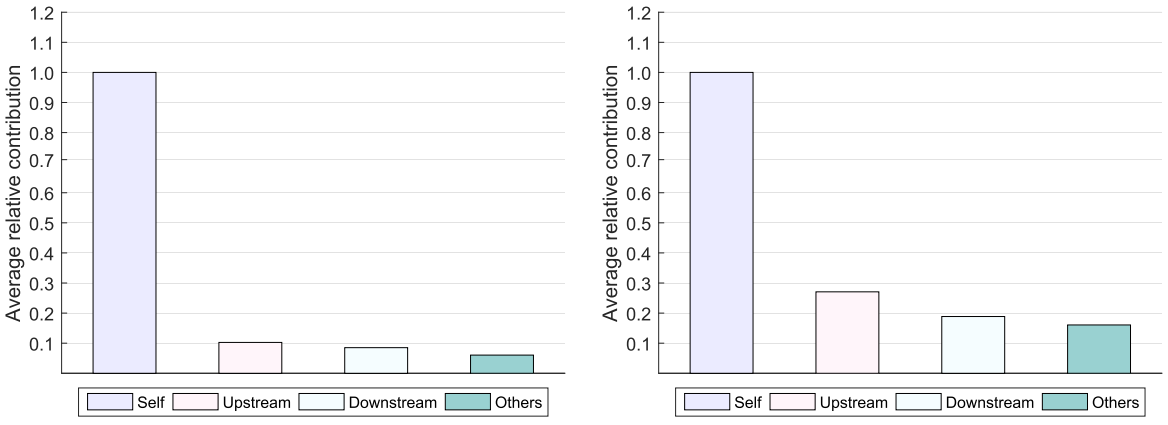
where $W_{fc}(j, s + (t-1)|p|)$ indicates the element with row index j and column index $s + (t-1)|p|$ of the weight matrix for fully-connected layer. x_j, $j = 1, 2 \cdots |p|$ represents the output from the las Path-LSTM layer. According to the proposed architecture, x_j is the feature provided by the j^{th} segment for prediction. Thus $q_{t,j}^{s}$ is regarded as the contribution of j^{th} segment to the prediction. Obviously, the larger the $qt;j$ s is, the more significant impact the j^{th} segment will have on prediction for s segment at $T + t$. Given the speed data in April 17, 2018, prediction test of 52 models (i.e., 52 critical paths) is conducted, from which overall 83,687 predictions are made. For each prediction in a critical path, the spatial feature is interpreted by analyzing the contribution from each segment.

For each segment, there are 4 kinds of contribution to traffic prediction: the contribution from itself, its upstream segment, its downstream segment and other segments in the same critical path, hence the average contribution corresponding to each kind can be calculated based on the predictions, then all the contribution is normalized at the contribution from itself for more clear presentation, defined as average relative contribution (ARC). Furthermore, we focus on two characteristic traffic scenarios to illustrate the intepretability. The first scenario is free flow traffic condition, where we defined free speed as the 85th percentile speed value of each segment (Cai et al., 2016); and the second scenario is congested traffic condition with speed lower than 20 km/h (Ma et al., 2017). As shown in Fig. 15, before training there is no significant difference between 4 kinds of contribution, which indicates that the model has



(a) Free flow traffic condition. (b) Congested traffic condition.

Fig. 15. Average relative contribution before training.



(a) Free flow traffic condition. (b) Congested traffic condition.

Fig. 16. Average relative contribution after training.

not learned the spatial feature. By contrast, once the model is trained, contribution from each kind of segment is distinguished. As shown in Fig. 16, the self-contribution plays the most important part in prediction under either free flow or congested condition, it makes sense for the reason that traffic flow is frequently interrupted by the traffic signal in urban network. Under this circumstance, the relationship between segments may be weaken, thus the prediction should heavily depend on the segment itself (i.e., history speed for this segment). In addition, for congested traffic conditions, the self-contribution still plays a dominating role. However, in this case the other three kinds of contribution increase, especially the upstream and downstream. This is because that as the traffic becomes congested, traffic flow among segments is more likely to affect each other and the traffic state in the critical path tends to have a closer relationship.

4. Conclusion and future work

In this study, we suggest a new angle of view to build an interpretable deep learning model for better traffic speed forecasting. The major contributions include three parts:

- Critical paths are defined to divide the study area, thus the network-wise high dimension traffic data is properly separated and can be addressed efficiently
- To the best of our knowledge, it is the first time that Bi-LSTM NN has been used to model path in road network for traffic forecasting task. And as we can exclusively exploit the spatial-temporal feature along each selected path, the useless information can be sifted and it is shown that the proposed framework achieves better prediction performance;
- In contrast to the previous studies, since we map deep learning network to physical network, the proposed method can naturally seize domain knowledge of transportation and present plausible physical meaning, specifically, the output of hidden layer is proved to be interpretable through visualization and qualitative analysis

In future works, as the critical path selection procedure is an important step in model construction, there is still space to analyze more selection criterion. Besides, there is still an open issue of elevating the interpretability of deep learning model for transportation application and we will continue to do more research on this area. And how to make use of the interpretability will also be an interesting research direction.

Acknowledgment

This research is supported by the National Natural Science Foundation of China (No. U1611461). Guangzhou Major Scientific and Technological Project (No. 20183900042050272).

Appendix A. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.trc.2019.02.002.

References

Asif, M.T., Dauwels, J., Goh, C.Y., Oran, A., Fathi, E., Xu, M., Dhanya, M.M., Mitrovic, N., Jaillet, P., 2014. Spatiotemporal patterns in large-scale traffic speed prediction. IEEE Trans. Intell. Transp. Syst. 15 (2), 794–804.

Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A., 2017. Network dissection: Quantifying interpretability of deep visual representations. arXiv preprint arXiv:1704.05796.
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. Transport. Res. Part C: Emerg. Technol. 62, 21–34.
Chan, K.Y., Dillon, T.S., Fellow, L., Singh, J., Chang, E., Member, S., 2012. Neural-Network-Based Models for Short-Term Traffic Flow Forecasting Using a Hybrid Exponential Smoothing and Levenberg Marquardt Algorithm. 13, pp. 644–654.
Clevert, D.-A., Unterthiner, T., Hochreiter, S., 2015. Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289.
Cong, Y., Wang, J., Li, X., 2016. Traffic flow forecasting by a least squares support vector machine with a fruit fly optimization algorithm. Procedia Eng. 137, 59–68.
Cui, Z., Ke, R., Wang, Y., 2018. Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction. arXiv preprint arXiv:1801.02143.
De Fabritiis, C., Ragona, R., Valenti, G., 2008. Traffic estimation and prediction based on real time floating car data. In: International IEEE Conference on Intelligent Transportation Systems, pp. 197–203.
Deng, M., Qu, S., 2016. Road short-term travel time prediction method based on flow spatial distribution and the relations. Mathematical Problems in Engineering 2016.
Du, S., Li, T., Gong, X., Yu, Z., Horng, S.-J., 2018. A hybrid method for traffic flow forecasting using multimodal deep learning. arXiv preprint arXiv:1803.02099.
Duan, Y., Lv, Y., Wang, F.-Y., November 2016. Travel time prediction with lstm neural network. In: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), pp. 1053–1058.
Erhan, D., Bengio, Y., Courville, A., Vincent, P., 2009. Visualizing higher-layer features of a deep network. Univ. Montreal 1341 (3), 1.
Fu, R., Zhang, Z., Li, L., 2016. Using lstm and gru neural network methods for traffic flow prediction. In: Chinese Association of Automation (YAC), Youth Academic Annual Conference of. IEEE, pp. 324–328.
Guo, J., Huang, W., Williams, B.M., 2014. Adaptive kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. Transport. Res. Part C: Emerg. Technol. 43, 50–64.
Guyon, I., 2003. An introduction to variable and feature selection. JMLR.org.
Hochreiter, S., 1991. Untersuchungen zu dynamischen neuronalen netzen. Master's Thesis, Institut Fur Informatik. Technische Universitat, Munchen.
Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.
Karlaftis, M.G., Vlahogianni, E.I., 2011. Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. Transport. Res. Part C: Emerg. Technol. 19, 387–399. https://doi.org/10.1016/j.trc.2010.10.004.
Kim, B., Doshi-Velez, F., 2017. Interpretable machine learning: the fuss, the concrete and the questions. ICML Tutorial on interpretable machine learning.
Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
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), 818.
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. Transport. Res. Part C: Emerg. Technol. 54, 187–197.
Min, W., Wynter, L., 2011. Real-time road traffic prediction with spatio-temporal correlations. Transport. Res. Part C: Emerg. Technol. 19 (4), 606–616.
Montavon, G., Lapuschkin, S., Binder, A., Samek, W., Müller, K.-R., 2017. Explaining nonlinear classification decisions with deep taylor decomposition. Pattern Recogn. 65, 211–222.
Nguyen, A., Yosinski, J., Clune, J., 2016. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. arXiv preprint arXiv:1602.03616.
Qi, Y., Ishak, S., 2014. A hidden markov model for short term prediction of traffic conditions on freeways. Transport. Res. Part C: Emerg. Technol. 43, 95–111.
Ryu, U., Wang, J., Kim, T., Kwak, S., Juhyok, U., 2018. Construction of traffic state vector using mutual information for short-term traffic flow prediction. Transport. Res. Part C: Emerg. Technol. 96, 55–71.
Schuster, M., Paliwal, K.K., 1997. Bidirectional recurrent neural networks. IEEE Trans. Signal Process. 45 (11), 2673–2681.
Simonyan, K., Vedaldi, A., Zisserman, A., 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034.
Tian, Y., Pan, L., 2015. Predicting short-term traffic flow by long short-term memory recurrent neural network. In: IEEE International Conference on Smart City/socialcom/sustaincom. pp. 153–158.
Wang, J., Deng, W., Guo, Y., 2014. New bayesian combination method for short-term traffic flow forecasting. Transport. Res. Part C: Emerg. Technol. 43, 79–94.
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.
Wu, Y., Tan, H., Qin, L., Ran, B., Jiang, Z., 2018. A hybrid deep learning based traffic flow prediction method and its understanding. Transport. Res. Part C: Emerg. Technol. 90, 166–180.
Yu, B., Yin, H., Zhu, Z., 2017a. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting.
Yu, H., Wu, Z., Wang, S., Wang, Y., Ma, X., 2017b. Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. Sensors 17 (7), 1501.
Zeiler, M. D., Fergus, R., 2014. Visualizing and understanding convolutional networks. In: European conference on computer vision. Springer, pp. 818–833.
Zhao, Z., Chen, W., Wu, X., Chen, P.C., Liu, J., 2017. Lstm network: a deep learning approach for short-term traffic forecast. IET Intel. Transport Syst. 11 (2), 68–75.
Zhou, Z.-H., 2012. Ensemble Methods: Foundations and Algorithms. Chapman and Hall/CRC.