Long-Term Traffic Speed Prediction Based on Multiscale Spatio-Temporal Feature Learning Network

Di Zang, *Member, IEEE*, Jiawei Ling, Zhihua Wei, *Member, IEEE*, Keshuang Tang, and Jiujun Cheng

***Abstract*— Speed plays a significant role in evaluating the evolution of traffic status, and predicting speed is one of the fundamental tasks for the intelligent transportation system. There exists a large number of works on speed forecast; however, the problem of long-term prediction for the next day is still not well addressed. In this paper, we propose a multiscale spatio-temporal feature learning network (MSTFLN) as the model to handle the challenging task of long-term traffic speed prediction for elevated highways. Raw traffic speed data collected from loop detectors every 5 min are transformed into spatial–temporal matrices; each matrix represents the one-day speed information, rows of the matrix indicate the numbers of loop detectors, and time intervals are denoted by columns. To predict the traffic speed of a certain day, nine speed matrices of three historical days with three different time scales are served as the input of MSTFLN. The proposed MSTFLN model consists of convolutional long short-term memories and convolutional neural networks. Experiments are evaluated using the data of three main elevated highways in Shanghai, China. The presented results demonstrate that our approach outperforms the state-of-the-art work and it can effectively predict the long-term speed information.**

***Index Terms*— Traffic speed prediction, multiscale spatio-temporal feature learning network, deep learning, intelligent transportation system.**

I. Introduction

TRAFFIC speed prediction is of significance for the Intelligent Transportation System (ITS), it provides useful information for both the vehicle drivers and traffic management agencies. Elevated highways, as important components of urban road transportation, expand the traffic capacity of road network. The accurate speed prediction of elevated highways can help road users to avoid traffic congestion and establish a good travel. In addition, the transportation management agencies can optimize the distribution of traffic resource. According to the data collected from the road sensors, researchers attempt to estimate traffic information with many approaches including data driven statistical models and machines learning based methods.

Among the various data driven statistical models, the Auto-Regressive Integrated Moving Average (ARIMA) approach and some optimized models based on ARIMA were applied in many researches. It is a time-series prediction model which considers the correlations in successive time sequences of traffic variables. In [1], ARIMA model was first introduced for traffic data prediction. The extensions of ARIMA [2] were also proposed to estimate traffic flows. Other data driven statistical approaches include Karman filters [3] and stochastic Lagrangian traffic flow model [4].

Machine learning based methods such as K-Nearest Neighbors(KNN), Support Vector Machine (SVM) and Support Vector Regression (SVR) have attracted much attention for the estimation of traffic data. Sun *et al.* [5] applied flow-aware WPT KNN to predict traffic parameters. A hybrid prediction method [6] was proposed by combining SVM and double exponential smoothing in a framework. Some models based on SVR, such as Seasonal SVR [7] and Online-SVR [8], optimize the conventional SVR and improve the accuracy of traffic prediction. Due to the flexible implementation and the ability to deal with multi-dimensional traffic data, Artificial Neural Networks (ANNs) have been used in traffic prediction. In [9], the ANN model, which coupled raw traffic parameters and the weather data as inputs, was applied in traffic speed prediction. Asif *et al.* [10] presented a real-time model to estimate speed based on ANN. A new ANN based model combined with conventional Bayes theorem [11] was employed in short-term freeway traffic information prediction. ANNs are also used in road traffic prediction and congestion control [12]. Additionally, RBF neural network [13] and improved fuzzy neural network [14] were also applied in traffic prediction.

Recently, deep learning models have been proved to achieve perfect performance in many fields [15]. Researchers in transportation are interested in applying new deep learning models to solve problems of traffic information prediction. At present, deep learning methods are used in short-term traffic prediction. Deep Belief Networks(DBNs) [16] have been favored by scholars [17]. Koesdwiady *et al.* [18] have proposed a DBN-based model which takes advantages of historical flow, weather data and event-based data to predict

Manuscript received January 23, 2018; revised July 4, 2018 and September 8, 2018; accepted October 21, 2018. This work was supported by the National Natural Science Foundation of China under Grant 61876218, Grant 61872271, and Grant 61573259. The Associate Editor for this paper was X. Ma. *(Corresponding authors: Di Zang; Zhihua Wei.)*

D. Zang, J. Ling, Z. Wei, and J. Cheng are with the Department of Computer Science and Technology and the Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 200092, China (e-mail: zangdi@tongji.edu.cn; zhihua_wei@tongji.edu.cn).

K. Tang is with the Department of Traffic Information Engineering and Control, Tongji University, Shanghai 200092, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2018.2878068

traffic flow, and the DBN was also used to improve the performance with the help of the big data [19]. Ma *et al.* [20] used the Long Short-Term Memory (LSTM) as the model to predict the traffic speed. Jia *et al.* [21] compared the performance of DBN and LSTM. Stacked Auto Encoder(SAE) model has also been used to predict the traffic flow [22]. In a recent research, a new deep learning model which combines Restricted Boltzmann Machine (RBM) and Recurrent Neural Network (RNN) [23] has achieved a good performance in traffic prediction since the model integrates advantages of both RNN and RBM. A Convolutional Neural Network (CNN) based model [24] was proposed to solve the problems of traffic speed prediction, which learned traffic speed data as images, experimental results demonstrate that the presented method outperformed other deep learning models. Research work presented in [25] and [26] also illustrate the good performance of CNN based prediction. In [27], CNN and Recurrent Neural Networks (RNNs) are combined to extract spatial-temporal traffic information for traffic condition forecasting, and Li *et al.* [28] proposed a diffusion convolutional recurrent neural network, however, they all consider only the short-term prediction problem, other related state-of-the-art approaches include FCL-Net [29], DMVST-Net [30], ST-ResNet [31] and some new models [32], [33].

In general, deep learning based methods are proved to have much better performance. Up to now, these approaches focus on the problem of short-term traffic data prediction. However, to support advanced transportation management, it is also required to estimate traffic data in a long-term way, i.e., predicting at least 24 hours in advance. Traffic speed is time series which has periodic characteristics, speed matrices fuse both the spatial and temporal information, and Convolutional Long Short-Term Memory (ConvLSTM) [34] has the advantage of extracting spatio-temporal features, therefore, in this paper, we are inspired to propose a multiscale spatio-temporal feature learning network (MSTFLN) to solve the task of long-term traffic speed prediction for elevated highways by coupling ConvLSTMs and CNNs into a single framework.

We first transform raw data collected from loop detectors as spatio-temporal matrices, then each matrix is resampled to generate 3 corresponding matrices according to different time scales. Speed matrices of 3 historical days form a group, 3 groups of matrices representing various time intervals are combined as inputs for MSTFLN to yield the final prediction. The proposed MSTFLN model contains multi-channel ConvLSTMs and CNNs for learning multiscale spatio-temporal information. Experimental results illustrate that our approach can effectively forecast long-term traffic speed and it performs the best when compared with the state-of-the-art work.

II. Proposed Methodology

The proposed methodology is illustrated in figure 1, raw traffic speed data associated with positions are collected from loop detectors on the elevated highways, they are first pre-processed to remove abnormal elements. The main purpose of our paper is to perform a one-day long term speed estimation, to this end, the traffic speed matrix which fuses one-day spatial and temporal information is yielded by encoding the number of the loop detector and the time scale to the y-axis and x-axis, respectively. The number of loop detectors illustrates the position information in the elevated highway. Each element of the matrix denotes the speed value. From the generated speed matrices database, training samples are selected to train the MSTFLN model, the rest speed matrices are considered as test samples to estimate the speed information of the corresponding next day.

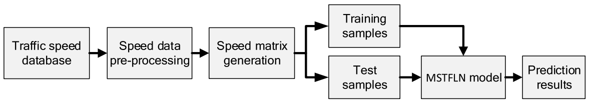


Fig. 1. The system architecture of traffic speed prediction using the MSTFLN model.

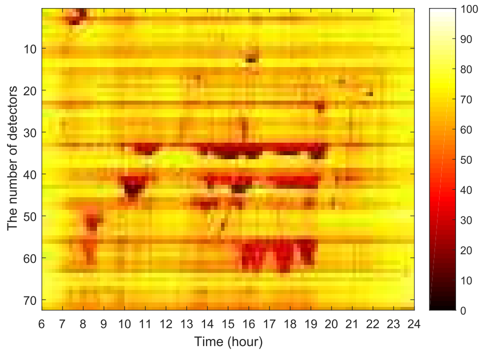


Fig. 2. An example of speed matrix visualized as a heat map.

A. Traffic Speed Matrix Generation

Traffic speed data records the evolution of traffic status of a particular region for a period of time, they can be collected by road sensors such as inductive loop detectors. Generally, traffic speed data coming from the loop detector has a time interval of 5 minutes, loop detectors are installed on highways with some distance in between, the speed information associated with the position represents the average speed. Speed data can be regarded as time-varying signals, to explore the way of speed prediction, spatial positions of loop detectors are also important information for consideration. Hence, speed values and positions are coupled in a framework to generate the spatio-temporal speed matrix for further exploration. In this work, time scales and positions are respectively encoded by the x and y axes, elements of the matrix refer to speed values. Let m and n indicate the number of loop detectors and the number of time intervals, the spatio-temporal speed matrix can be represented as

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \tag{1}$$

where x_{ij} is the average speed of the ith loop detector at the jth time period. As shown in figure 2, a speed matrix can be visualized as a heat map.

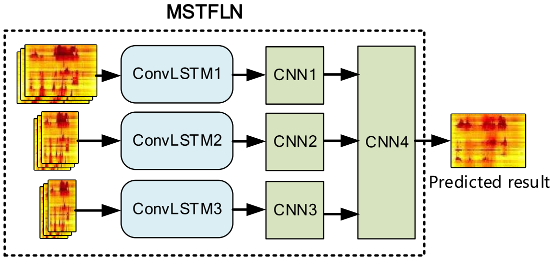


Fig. 3. The structure of the proposed multiscale spatio-temporal feature learning network.

B. Multiscale Spatio-Temporal Feature Learning Network

In order to have more robust speed prediction results, we propose the MSTFLN method. As demonstrated in figure 3, our model includes three channels for learning features of three time scales, for each channel, three matrices corresponding to speed data of three historical days are first fed to the ConvLSTM modules. These modules are employed for extracting both the spatial and temporal features, after that, CNNs are used to learn more high-level abstract features in a hierarchical way. Feature maps generated by CNN1, CNN2 and CNN3 are then directly delivered to the fourth CNN to jointly learn multiscale spatial and temporal features and yield the prediction results. The output of the presented method has the same form as the input which indicates the estimated speed information of the next day.

1) The Input Speed Matrix: The original input of the MSTFLN consists of 9 speed matrices corresponding to three days and three time scales. The raw matrix denotes speed data of every 5 minutes, to learn multiscale features, three groups of matrices, which are formed by resampling the raw matrix with time intervals of 10, 20 and 30 minutes, are generated. Since the traffic data exhibits an evident periodicity of properties, for each time scale, matrices of the current day and two other historical days which have the same weekdays as the predicted one are considered as the inputs. To predict the speed of the dth day, the first matrix illustrates the speed information of the $(d-1)th$ day, the second and third matrices are respectively associated with the $(d-7)th$ and $(d-14)th$ days, which have the same weekdays as the dth day, e.g. they can all represent the speed information of Tuesday. Each input speed matrix can be represented as

$$X(d_i;\sigma_j) = \begin{bmatrix} \boldsymbol{x_{11}(d_i;\sigma_j)} & \dots & \boldsymbol{x_{1n}(d_i;\sigma_j)} \\ \boldsymbol{x_{21}(d_i;\sigma_j)} & \dots & \boldsymbol{x_{2n}(d_i;\sigma_j)} \\ \dots & \dots & \dots \\ \boldsymbol{x_{m1}(d_i;\sigma_j)} & \dots & \boldsymbol{x_{mn}(d_i;\sigma_j)} \end{bmatrix} \tag{2}$$

where d_i with $i \in \{1,2,3\}$ indicates the day of input matrices and σ_j with $j \in \{1,2,3\}$ denotes the time scale, $X(d_i;\sigma_j)$ refers to the speed matrix of the corresponding day and time scale, m and n indicate the number of loop detectors and time intervals, because of the resampling operation, n has different values for each time scale. For a fixed time scale σ_j, the matrices of $X(d_1;\sigma_j)$, $X(d_2;\sigma_j)$ and $X(d_3;\sigma_j)$ illustrate the speed information of the $(d-14)th$, $(d-7)th$ and $(d-1)th$ days.

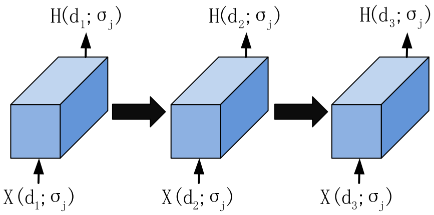


Fig. 4. The structure of the ConvLSTM module.

2) The ConvLSTM Modules: The traffic speed data are time series, speed matrices are constructed by adding the position information of loop detectors to the second dimension, in such a case, they contain significant characteristics both in the spatial and temporal domains. In addition, traffic speed information shows strong periodical properties, therefore, ConvLSTM modules, which have the advantage of simultaneous spatial-temporal feature learning, are applied. Figure 4 demonstrates the structure of the ConvLSTM module for a fixed time scale σ_j, there are three cells corresponding to three time steps in this module, speed matrices of three days are sequentially fed to each cell, the final output of the ConvLSTM contains learned spatio-temporal characteristics.

For each cell, assume X_t represents the input speed matrix of the current time step $X(d_i;\sigma_j)$, the corresponding output H_t can be computed according to the following formula

$$\begin{aligned} i_t &= \sigma\left(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i\right), \\ f_t &= \sigma\left(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f\right), \\ C_t &= f_t \circ C_{t-1} + i_t \circ tanh\left(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c\right), \\ o_t &= \sigma\left(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o\right), \\ H_t &= o_t \circ tanh\left(C_t\right). \end{aligned} \tag{3}$$

where i_t, f_t, C_t, o_t refer to the input gate, forget gate, cell status and output gate; σ and $tanh$ indicate active functions; symbols $*$ and $\circ$ are convolution operator and Hadamard product; W and b represent the corresponding weight and bias parameters.

3) The CNN Modules: The proposed MSTFLN includes 4 CNN modules, CNN1, CNN2 and CNN3 are applied to extract more advanced abstract spatio-temporal features for each time scale, feature maps yielded by these CNN modules are sent to CNN4 for multiscale feature joint learning and generating the prediction result. Because the inputs of three scale channels have different sizes, CNN1, CNN2 and CNN3 are designed not only to refine feature extraction but also to yield the same output sizes for CNN4, in such a case, multiscale information can be further fused and learned.

As shown in figure 5, CNN1, CNN2 and CNN3 have three convolution layers followed by a pooling layer, CNN4 contains two convolution layers, one pooling layer and a fully connected layer.

The convolutional layer plays a significant role in the CNN, it enables the extraction of spatio-temporal features by convolving the input with learnable kernels. Although both spatial and temporal information is taken into consideration, the temporal relation is undoubtedly more important, therefore,

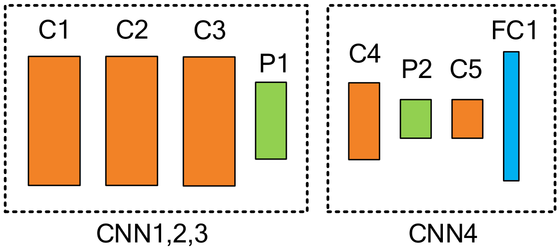


Fig. 5. The structure of the CNN modules, C means the convolution layer, P refers to the pooling layer and FC indicates the fully connected layer.

asymmetric convolution kernels are employed. Assume $\mathbf{k}$ indicates the convolution kernel with a window size of $p \times q$ with $q >= p$, the convolved output can be written as

$$x^l = f\left(\sum_{i=1}^{p}\sum_{j=1}^{q} x_{ij}^{l-1} k_{ij}^{l} + b^l\right) \tag{4}$$

where x^l represents the neuron output of the feature matrix at the lth layer, x_{ij}^{l-1} denotes the neighborhood neurons in the previous layer, k_{ij}^{l} refers to the related kernel weights, b^l means the bias parameter of layer l and f is the rectified liner unit active function which is shown as the following

$$f(x) = \begin{cases} 0, & x <= 0 \\ x, & x > 0 \end{cases} \tag{5}$$

A feature map can be generated by sliding the kernel window and performing the above operation on the input. Different inputs convolved with the same kernel are merged to yield a single feature map. A convolution layer consists of multiple feature maps by convolving with multiple kernels.

The pooling layer aims to reduce the number of neurons by downsampling the convolved result with a given pooling technique. Compared with the images in the domain of computer vision, the pattern of the input in the current context is relatively simply, pooling after each convolution layer can result in dramatic information loss. Therefore, for the first, second and third CNNs, there exist three consecutive convolution layers followed by one single pooling layer. And the fourth CNN contains two convolutional layers interleaved by one pooling layer. After convolution layers, maps with more abstract features are obtained, applying the pooling operation to these feature maps can not only decrease the number of the network parameters but also keep the significant pattern information. In this paper, the max pooling technique is employed. Given a pooling window $P \times Q$, the feature map can be divided into multiple subregions according to the size of the pooling window. Each subregion is then replaced by the maximum point in this region, in such a way, feature maps are downsampled by max pooling.

The full connection layers of CNN 1, 2 and 3 are truncated, generated feature matrices are delivered to CNN4 for the joint-learning of advanced characteristics. Let $\mathbf{X}^{L,j}$ represent the jth matrix with a dimension of $u \times v$ in the last pooling layer L, in the full connection layer, it is first reshaped to form a vector $\mathbf{y}^j$, that is

$$\mathbf{y}^j = \left[x_{11}^{L,j}, \ldots, x_{1v}^{L,j}; x_{21}^{L,j}, \ldots, x_{2v}^{L,j}; \ldots; x_{u1}^{L,j}, \ldots, x_{uv}^{L,j}\right] \tag{6}$$

Assume there are N matrices in the last pooling layer of the fourth CNN, cascading all the reshaped vectors can yield a very long vector which can be represented as

$$\mathbf{Y} = [\mathbf{y}^1; \mathbf{y}^2; \ldots; \mathbf{y}^j; \ldots \mathbf{y}^N] \tag{7}$$

For the purpose of prediction, the output layer contains nodes with the same number of the input. All neurons in the full connection layer and the output layer are globally connected, the jth speed node O_j of the next day can then be estimated as

$$O_j = f(\sum_{i=1}^{u\times v\times N} w_i y_i + b_i) \tag{8}$$

where y_i refers to one node in the full connection layer, w_i and b_i denote the associated weight and bias, respectively. And the employed active function at this phase is a sigmoid function

$$f(x) = \frac{1}{1+e^x} \tag{9}$$

4) Model Optimization: To predict the traffic speed, parameters need to be trained with training samples. The employed loss function to evaluate the difference between the ground truth and the estimated result is mean squared error (MSE), which can be optimized by the stochastic gradient decent method. Given N training samples, the loss function reads

$$E = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{O}' - \mathbf{O})^2 \tag{10}$$

where E means the loss function, $\mathbf{O}'$ and $\mathbf{O}$ indicate the predicted speed vector and the ground truth, respectively. The goal of the optimization is to find proper values of network parameters for minimizing the loss function. Let ϕ be the set of parameters of the MSTFLN model, all the parameters can be learned by optimizing the loss function using the stochastic gradient decent method. The update expression of the parameter set $\triangle\phi$ is

$$\triangle\phi = -\eta\frac{\partial E}{\partial \phi} \tag{11}$$

where η means the learning rate.

III. Experimental Results

A. Experimental Data and Setting

In our experiments, the speed data are collected from loop detectors installed on Yan'an, Nanbei and Neihuan elevated highways of year 2011. As the backbones of urban road network, these three elevated highways have significantly expanded the traffic capacity of Shanghai, China. On the elevated highway, there exists a loop detector every 400 meters. Traffic data are recorded every 5 minutes. The number of detectors for the Yan'an, Nanbei and Neihuan elevated highways are 35, 43 and 72, respectively. Figure 6 shows the map

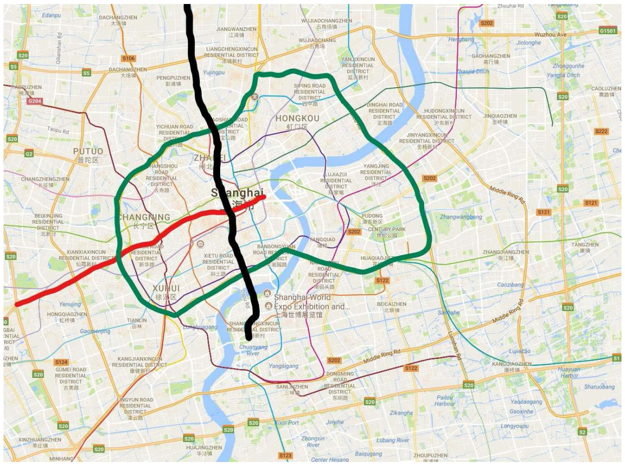


Fig. 6. Marked elevated highways. Red, black and green lines respectively indicate Yan'an, Nanbei and Neihuan elevated highways.

TABLE I

SETTINGS OF MSTFLN FOR YAN'AN ELEVATED HIGHWAY

Convlstm1	**Convlstm2**	**Convlstm3**
32, 3 × 3, 3	32, 3 × 3, 3	32, 3 × 3, 3
CNN1	**CNN2**	**CNN3**
C1: 32, 3 × 3	C1: 32, 3 × 5	C1: 32, 3 × 9
C2: 32, 3 × 3	C2: 32, 3 × 4	C2: 32, 3 × 7
C3: 64, 3 × 3	C3: 64, 3 × 5	C3: 64, 3 × 5
P1: 2 × 2	P1: 2 × 3	P1: 2 × 6
CNN 4		
C4: 256, 3 × 3		
P2: 2 × 2		
C5: 256, 3 × 3		
FC1: 3780		

TABLE II

SETTINGS OF MSTFLN FOR NANBEI ELEVATED HIGHWAY

Convlstm1	**Convlstm2**	**Convlstm3**
32, 3 × 3, 3	32, 3 × 3, 3	32, 3 × 3, 3
CNN1	**CNN2**	**CNN3**
C1: 32, 3 × 3	C1: 32, 3 × 5	C1: 32, 3 × 9
C2: 32, 3 × 3	C2: 32, 3 × 4	C2: 32, 3 × 7
C3: 64, 2 × 3	C3: 64, 2 × 3	C3: 64, 2 × 5
P1: 2 × 2	P1: 2 × 3	P1: 2 × 6
CNN 4		
C4: 256, 3 × 3		
P2: 2 × 2		
C5: 256, 3 × 3		
FC1: 4644		

TABLE III

SETTINGS OF MSTFLN FOR NEIHUAN ELEVATED HIGHWAY

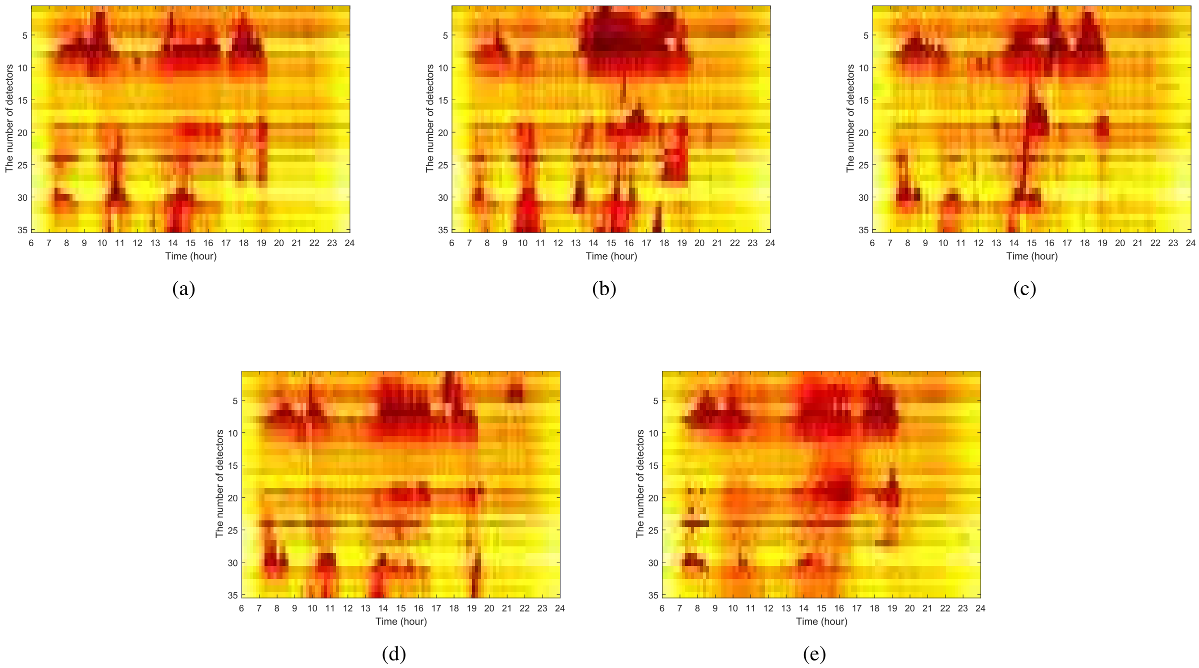
Convlstm1	**Convlstm2**	**Convlstm3**
32, 3 × 3, 3	32, 3 × 3, 3	32, 3 × 3, 3
CNN1	**CNN2**	**CNN3**
C1: 32, 3 × 11	C1: 32, 5 × 11	C1: 32, 9 × 11
C2: 32, 3 × 10	C2: 32, 4 × 10	C2: 32, 7 × 10
C3: 64, 3 × 9	C3: 64, 3 × 9	C3: 64, 5 × 9
P1: 2 × 3	P1: 3 × 3	P1: 3 × 6
CNN 4		
C4: 256, 3 × 3		
P2: 2 × 2		
C5: 256, 3 × 3		
FC1: 7776		

of the above mentioned elevated highways. Thick lines marked by the red, black and green colors respectively represent the Yan'an, Nanbei and Neihuan elevated highways.

As mentioned before, we transform the raw data collected from loop detectors as original spatio-temporal speed matrices for each day. However, due to the malfunction of detectors, the original matrices contain a lot of errors, pre-processing of data is required. Because the speed limit of elevated highway in the city of Shanghai is 80 km/h, we set the maximum of matrix elements as 100 km/h so as to cover the situation of overspeed. In addition, there are some zeros in these matrices, they can be replaced by the average of adjacent values. Considering that some loop detectors are maintained at night, we choose data collected from 6 am to 12 pm. After the pre-processing, the original 5-min traffic speed matrix of each day is resampled to generate three corresponding matrices with time intervals of 10, 20 and 30 minutes. These matrices characterize the speed distribution at different precision scales. All the matrices for the whole year of 2011 form a basic dataset for each elevated highway.

To measure the performance of our approach, for each basic dataset, we randomly select 2 days of each week as test samples. For Yan'an, Nanbei and Neihuan elevated highways, their respective test sets include 95, 97 and 97 samples. Matrices of the remaining days are considered as the initial training set. Due to the lack of training samples, data augmentation is performed. We add random values to all the samples in each of the initial training set, the additive values obey uniform distribution which has a mean value of 0, a maximum of 10 and a minimum of -10. Compared with the initial version, the final training set has been expanded. The numbers of training samples for Yan'an, Nanbei and Neihuan elevated highways are increased as 756, 762 and 762 respectively.

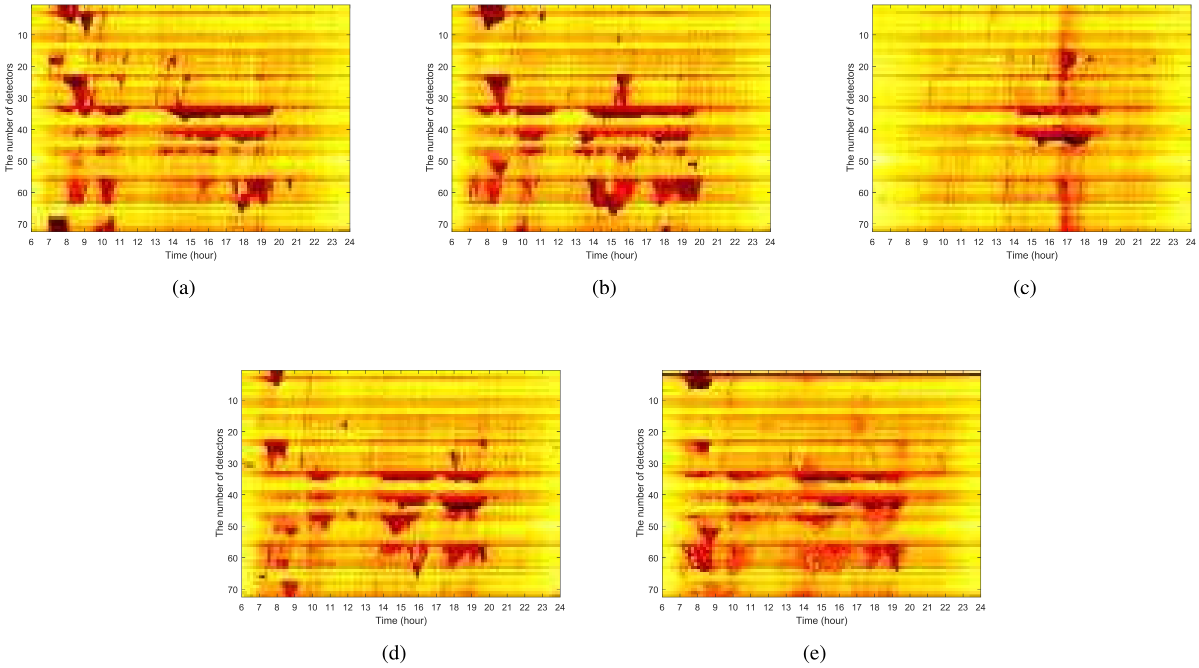
The experiments are conducted on a server with i7-5820K CPU, 48GB memory and NVIDIA GeForce GTX1080 GPU. The presented model is implemented on the TensorFlow framework of deep learning. The parameter settings of our method for 3 elevated highways are demonstrated in tables I - III. There are 3 ConvLSTMs and 4 CNN modules in the proposed MSTFLN model, each of the ConvLSTM contains 3 cells, CNN1, CNN2 and CNN3 for all cases have the same structure, i.e., three convolutional layers followed by one pooling layer. However, since the number of loop detectors are not the same for three elevated highways, corresponding kernel sizes of convolutional and pooling layers are slightly different. All modules of the fourth CNN include two convolutional layers, one pooling layer and one fully connected layer. In addition to the neuron numbers of the full connection layer, parameters for convolutional layers and pooling layers are identical. For the ConvLSTM module, the parameter representation indicates the number of kernels, kernel size and cell numbers. Parameters for CNNs refer to the kernel numbers, kernel sizes and neuron numbers in the fully connected layer. The learning rates of three models corresponding to three elevated highways are all initially set as 0.5. For each model, there exists 300 iterations. Every 100 iterations, the learning



(a) (b) (c)

(d) (e)

Fig. 7. Speed matrices of Yan'an elevated highway visualized as heat maps. (a) Speed matrix of the 128th day visualized as a heat map. (b) Speed matrix of the 135th day visualized as a heat map. (c) Speed matrix of the 141st day visualized as a heat map. (d) Ground truth of the speed matrix of the 142nd day visualized as a heat map. (e) Predicted speed matrix of the 142nd day visualized as a heat map.



(a) (b) (c)

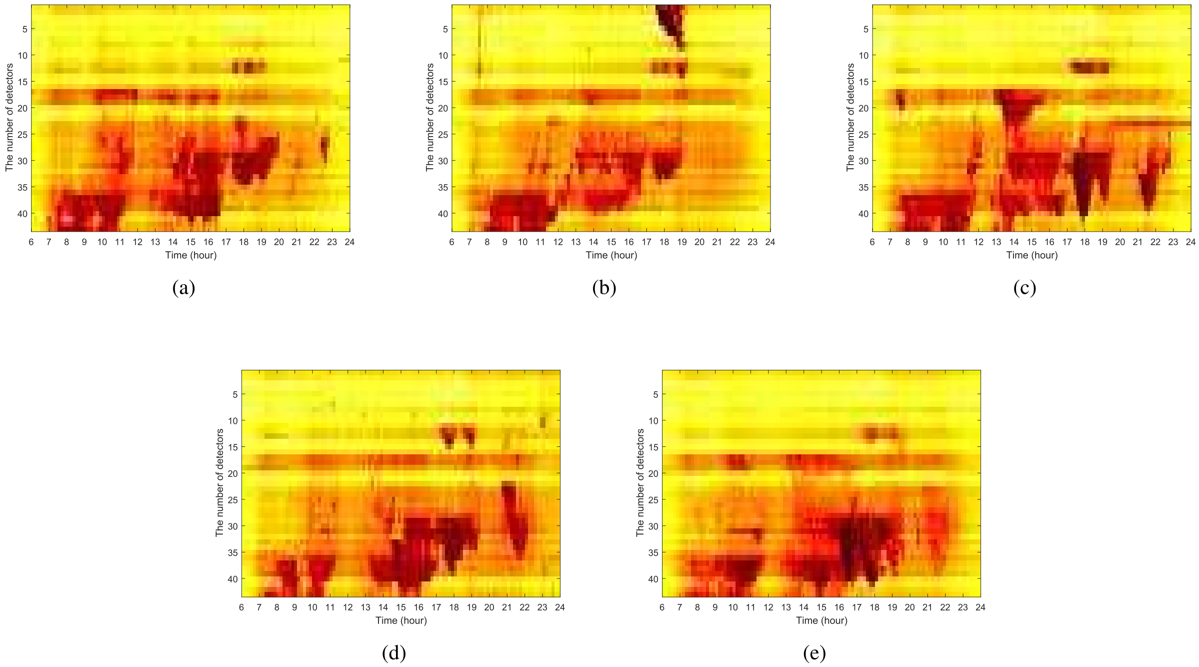
(d) (e)

Fig. 8. Speed matrices of Neihuan elevated highway visualized as heat maps. (a) Speed matrix of the 199th day visualized as a heat map. (b) Speed matrix of the 206th day visualized as a heat map. (c) Predicted speed matrix of the 212nd day visualized as a heat map. (d) Ground truth of the speed matrix of the 213rd day visualized as a heat map. (e) Predicted speed matrix of the 213rd day visualized as a heat map.

rate is updated as the half of the original one. Strides for all the convolution kernels are selected as 1 × 1.

B. Results and Analysis

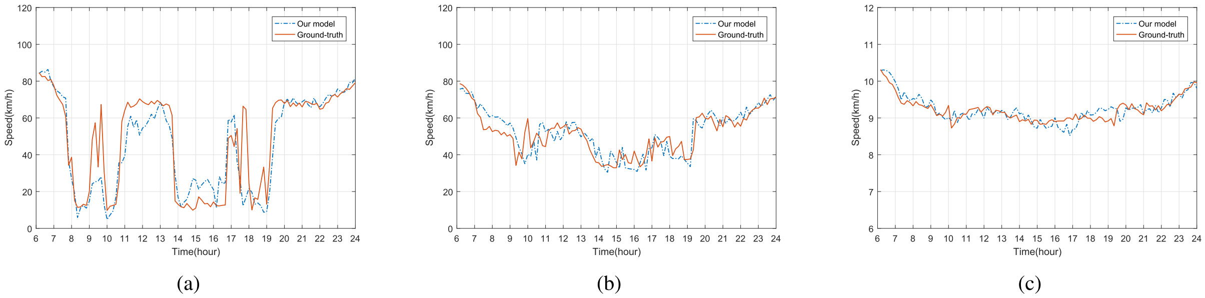
Speed matrix can be visualized as a heat map, the x-axis denotes the time intervals, and the y-axis indicates the number of loop detectors. The color of the heat map is mapped to speed values ranging from 0 to 100. The deeper the color is the lower the speed is. As shown in figure 7, heat maps of Yan'an elevated highway, which represent speed matrices of the 128th, 135th and 141st days are illustrated in the first row. The second row contains heat maps of the ground truth of the 142nd day and the corresponding predicted result. In the



(a) (b) (c)

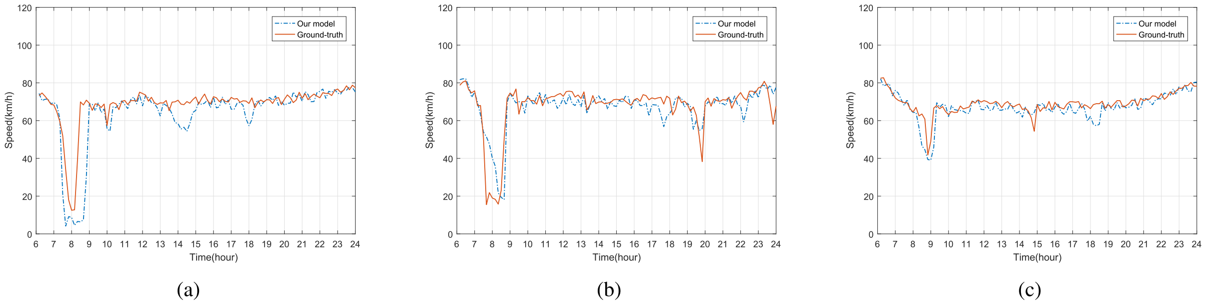
(d) (e)

Fig. 9. Speed matrices of Nanbei elevated highway visualized as heat maps. (a) Speed matrix of the 97th day visualized as a heat map. (b) Speed matrix of the 104th day visualized as a heat map. (c) Speed matrix of the 110th day visualized as a heat map. (d) Ground truth of the speed matrix of the 111st day visualized as a heat map. (e) Predicted speed matrix of the 111st day visualized as a heat map.



(a) (b) (c)

Fig. 10. Predicted speed curves and the corresponding ground truth of three loop detectors of Yan'an elevated highway in the 142nd day. (a) The prediction and the ground truth of the 7th loop detector on the Yan'an elevated highway. (b) The prediction and the ground truth of the 10th loop detector on the Yan'an elevated highway. (c) The prediction and the ground truth of the 12th loop detector on the Yan'an elevated highway.

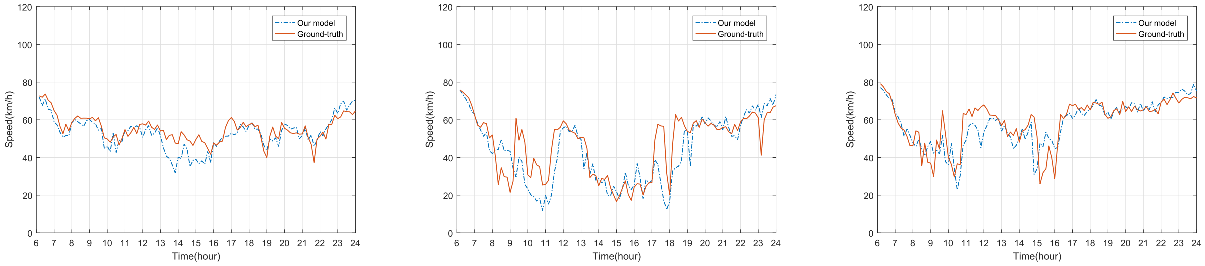


(a) (b) (c)

Fig. 11. Predicted speed curves and the corresponding ground truth of three loop detectors of Neihuan elevated highway in the 213rd day. (a) The prediction and the ground truth of the 4th loop detector on the Neihuan elevated highway. (b) The prediction and the ground truth of the 25th loop detector on the Neihuan elevated highway. (c) The prediction and the ground truth of the 53th loop detector on the Neihuan elevated highway.

current context, speed information of the 142nd day is required for prediction, 135th and 128th days respectively correspond to the same week days in the previous one and two weeks, e.g., they are all Tuesday.

In figure 8, speed matrices of Neihuan elevated highway, corresponding to the 199th, 206th and 212nd days, are demonstrated in the first row, the second row includes the ground truth and the predicted information of the 213rd day.



(a) (b) (c)

Fig. 12. Predicted speed curves and the corresponding ground truth of three loop detectors of Nanbei elevated highway in the 111st day. (a) The prediction and the ground truth of the 17th loop detector on the Nanbei elevated highway. (b) The prediction and the ground truth of the 38th loop detector on the Nanbei elevated highway. (c) The prediction and the ground truth of the 42nd loop detector on the Nanbei elevated highway.

For Nanbei elevated highway, heatmaps of the 97th, 104th and 110th days are shown in the first row of figure 9, the corresponding ground truth and prediction of the 111th day are illustrated in the second row. Estimated heat maps of all elevated highways indicate that our method can yield visually approximate results when compared with the ground truth.

Figure 10 shows the predicted speed curves and the corresponding ground truth of Yan'an elevated highway. Subfigures from left to right represent speed values of three different loop detectors in the 142nd day. For Neihuan and Nanbei elevated highways, estimated speed information and the corresponding ground truth of the 213rd day and the 111st day are illustrated in figures 11 and 12. It demonstrates that our prediction results are good approximations of the ground truth. Three metrics, which aim to quantitatively evaluate the prediction performance, are employed, i.e., Mean Relative Error(MRE), Mean Squared Error (MSE) and Mean Absolute Error(MAE). MSE and MAE can evaluate absolute errors between the prediction and real values while MRE is used to measure the relative error. These three metrics can be respectively computed using the following formulas:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{y}' - \mathbf{y})^2 \tag{12}$$

$$MRE = \frac{1}{N}\sum_{i=1}^{N}\frac{|\mathbf{y}' - \mathbf{y}|}{\mathbf{y}} \tag{13}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\mathbf{y}' - \mathbf{y}| \tag{14}$$

where $\mathbf{y}$ denotes the prediction, $\mathbf{y}'$ means the ground truth and N indicates the number of samples in the test set.

To measure the performance of speed estimation, we compare our approach with the methods of conventional CNN, ANN, ARIMA and several state-of-the-art approaches, i.e., FCL-Net [29], DMVST-Net [30] and ST-ResNet [31]. In addition, three variants of our model are also compared. The first one is MSCNN-SC which has multiscale CNNs followed by a single CNN, it simply omits 3 ConvLSTM modules. The second one is MSCNN-FC which includes multiscale CNNs followed by a fully connected layer, this model not only removes 3 ConvLSTM modules but also replaces the CNN4 with a single fully connected layer. The third one is MSTFLN-FC, compared with the proposed MSTFLN model, CNN4 is replaced by a full connection layer, other modules remain the same.

TABLE IV
COMPARISON RESULTS FOR YAN'AN ELEVATED HIGHWAY

Models	MSE	MAE	MRE
MSTFLN	**102.689**	**6.571**	**0.165**
MSTFLN-FC	133.872	7.356	0.170
MSCNN-SC	108.958	6.731	0.175
MSCNN-FC	197.236	8.754	0.205
FCL-Net	142.230	8.123	0.194
DMVST-Net	158.698	8.856	0.190
ST-ResNet	132.674	7.612	0.182
CNN	161.015	8.826	0.201
ANN	166.824	8.619	0.239
ARIMA	286.693	10.372	0.249

In the experiment, the hidden layer of ANN contains 1500 neurons, conventional CNN includes 4 convolutional layers, 2 max pooling layers and 1 fully connected layer. Two variants of our model have the same parameters as our method except the missing parts. For the DMVST-Net, the first two layers are 2 convolutional layers which are followed by one max pooling layer and a fully connected layer. After the full connection layer is a LSTM module containing 3 cells, the last part of the DMVST-Net is a fully connected layer. The FCL-net has 3 concatenated ConvLSTMs, each ConvLSTM has also 3 cells. After the ConvLSTM modules, there exists one convolutional layer followed by a fully connected layer. The ST-ResNet includes 3 branches corresponding to three time fragments, each one is a residual network, results generated by these 3 branches are then fused by a fully connected layer.

The comparison of speed prediction evaluated by the metrics of MSE, MAE and MRE are listed in tables IV, V and VI. For all three elevated highways, our method MSTFLN outperforms the traditional approaches and several state-of-the-art models such as ST-ResNet, DMVST-Net and FCL-Net. As one of the variants of our model, MSCNN-SC performs slightly worse than MSTFLN due to the lack of ConvLSTM modules. Compared with MSCNN-SC, the other variant MSCNN-FC works much worse because multiscale features are not well fused and learned by the deep structure of CNN. As the third variant, MSTFLN-FC substitutes CNN4 with a full connection

TABLE V
COMPARISON OF RESULTS FOR NEIHUAN ELEVATED HIGHWAY

Models	MSE	MAE	MRE
MSTFLN	**98.659**	**5.350**	**0.141**
MSTFLN-FC	117.954	6.869	0.160
MSCNN-SC	104.904	5.654	0.142
MSCNN-FC	136.356	8.058	0.191
FCL-Net	140.877	7.823	0.205
DMVST-Net	147.552	7.210	0.184
ST-ResNet	129.954	7.090	0.173
CNN	142.145	7.963	0.194
ANN	227.136	8.177	0.197
ARIMA	224.863	8.629	0.207

TABLE VI
COMPARISON RESULTS FOR NANBEI ELEVATED HIGHWAY

Models	MSE	MAE	MRE
MSTFLN	**82.487**	**5.238**	**0.138**
MSTFLN-FC	110.376	6.513	0.167
MSCNN-SC	83.3751	5.416	0.141
MSCNN-FC	111.397	7.299	0.187
FCL-Net	110.362	7.355	0.171
DMVST-Net	102.652	6.845	0.161
ST-ResNet	118.322	6.734	0.164
CNN	114.836	7.914	0.176
ANN	119.944	7.583	0.185
ARIMA	225.871	8.783	0.208

layer, although all the ConvLSTMs are kept, multiscale information learning remains insufficient, as a result, it performs even worse than MSCNN-SC. The ARIMA method shows the worst performance with high error values. Conventional machine learning approaches such as ANN works better than the ARIMA model. Because CNN has the ability to learn the features of inputs automatically, it performs slightly better than ANN. For Yan'an elevated highway, MSE, MAE and MRE metrics yielded by the proposed MSTFLN model, are 102.689, 6.571 and 16.5%, respectively. They are the lowest values among all the listed models. For Neihuan elevated highway, our method also obtains the best results with MSE, MAE and MRE being 98.659, 5.350 and 14.1%. Besides, evaluated metrics of MSE, MAE and MRE for Nanbei elevated highway are 82.487, 5.238 and 13.8%.

For the long-term traffic speed prediction, the input matrix represents one-day speed information, it contains significant features at different time scales which are the greatest contribution for predicting, approaches like MSTFLN, MSCNN-SC and ST-ResNet, which enable multiscale feature learning, can capture the complex characteristics of the input and show better performance. In addition, the MSTFLN method is able to extract better spatio-temporal features by coupling the advantages of ConvLSTM and CNN modules. Due to the periodic properties of traffic data, the carefully designed inputs also contribute to the enhancement of the prediction accuracy. The proposed MSTFLN approach takes into account multiscale spatial and temporal features, which can be jointly learned in a single framework. Presented experimental results prove that our method performs the best.

IV. CONCLUSIONS

In this paper, we present a multiscale spatio-temporal feature learning network (MSTFLN) to handle the challenging task of long-term speed forecast. The traffic speed information is formatted as matrices, for each matrix, the x-axis denotes time scales and y-axis represents the position information of loop detectors. The proposed model includes 3 ConvLSTMs and 4 CNNs, 9 speed matrices which correspond to 3 historical days and 3 time scales are taken as the inputs. Complex speed features can be extracted by jointly learning the multiscale inputs in both the spatial and temporal domains. Speed data obtained from loop detectors of elevated highways are used to evaluate the performance of the presented approach, experimental results illustrate that our method can yield good prediction results and outperform the state-of-the-art work.

REFERENCES

[1] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using box-jenkins techniques," *Transp. Res. Rec.*, no. 722, pp. 1–9, 1979.

[2] P. Duan, G. Mao, C. Zhang, and S. Wang, "STARIMA-based traffic prediction with time-varying lags," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 1610–1615.

[3] Z. H. Mir and F. Filali, "An adaptive Kalman filter based traffic prediction algorithm for urban road network," in *Proc. 12th Int. Conf. Innov. Inf. Technol. (IIT)*, Nov. 2016, pp. 1–6.

[4] K.-C. Chu, R. Saigal, and K. Saitou, "Stochastic Lagrangian traffic flow modeling and real-time traffic prediction," in *Proc. IEEE Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2016, pp. 213–218.

[5] B. Sun, W. Cheng, P. Goswami, and G. Bai, "Flow-aware WPT K-nearest neighbours regression for short-term traffic prediction," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2017, pp. 48–53.

[6] J. Tang, H. Wang, Y. Wang, X. Liu, and F. Liu, "Hybrid prediction approach based on weekly similarities of traffic flow for different temporal scales," *Transp. Res. Rec., J. Transp. Res. Board*, no. 2443, pp. 21–31, 2014.

[7] W.-C. Hong, "Traffic flow forecasting by seasonal SVR with chaotic simulated annealing algorithm," *Neurocomputing*, vol. 74, nos. 12–13, pp. 2096–2107, 2011.

[8] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, "Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 6164–6173, 2009.

[9] S. H. Huang and B. Ran, "An application of neural network on traffic speed prediction under adverse weather condition," in *Proc. Transp. Res. Board Annu. Meeting*, 2003.

[10] M. T. Asif *et al.*, "Spatiotemporal patterns in large-scale traffic speed prediction," *IEEE Transp. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 794–804, Feb. 2014.

[11] W. Zheng, D.-H. Lee, and Q. Shi, "Short-term freeway traffic flow prediction: Bayesian combined neural network approach," *J. Transp. Eng.*, vol. 132, pp. 114–121, Sep. 2006.

[12] R. More, A. Mugal, S. Rajgure, R. B. Adhao, and V. K. Pachghare, "Road traffic prediction and congestion control using artificial neural networks," in *Proc. Int. Conf. Comput., Anal. Secur. Trends (CAST)*, Dec. 2016, pp. 52–57.

[13] A. I. J. Tostes, T. H. Silva, R. Assuncao, F. L. P. Duarte-Figueiredo, and A. A. F. Loureiro, "STRIP: A short-term traffic jam prediction based on logistic regression," in *Proc. IEEE 84th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2016, pp. 1–5.

[14] 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.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[16] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[17] Y. Jia, J. Wu, and Y. Du, "Traffic speed prediction using deep learning method," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 1217–1222.

[18] A. Koesdwiady, R. Soua, and F. Karray, "Improving traffic flow prediction with weather information in connected cars: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9508–9517, Dec. 2016.

[19] R. Soua, A. Koesdwiady, and F. Karray, "Big-data-generated traffic flow prediction using deep learning and dempster-shafer theory," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 3195–3202.

[20] 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.

[21] Y. Jia, J. Wu, M. Ben-Akiva, R. Seshadri, and Y. Du, "Rainfall-integrated traffic speed prediction using deep learning method," *IET Intell. Transport Syst.*, vol. 11, no. 9, pp. 531–536, 2017.

[22] 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.

[23] X. Ma, H. Yu, Y. Wang, and Y. Wang, "Large-scale transportation network congestion evolution prediction using deep learning theory," *PLoS ONE*, vol. 10, no. 3, p. e0119044, 2015.

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

[25] C. Song, H. Lee, C. Kang, W. Lee, Y. B. Kim, and S. W. Cha, "Traffic speed prediction under weekday using convolutional neural networks concepts," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2017, pp. 1293–1298.

[26] D. Zang, J. Ling, J. Cheng, K. Tang, and X. Li, "Using convolutional neural network with asymmetrical kernels to predict speed of elevated highway," in *Proc. Int. Conf. Intell. Sci.*, in IFIP Advances in Information and Communication Technology. Cham, Switzerland: Springer, 2017, pp. 212–221, doi 10.1007/978-3-319-68121-4_22.

[27] X. Cheng, R. Zhang, J. Zhou, and W. Xu. (2017). "DeepTransport: Learning spatial-temporal dependency for traffic condition forecasting." [Online]. Available: https://arxiv.org/abs/1709.09585

[28] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[29] J. Ke, H. Zheng, H. Yang, and X. Chen, "Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach," *Transp. Res. C, Emerg. Technol.*, vol. 85, pp. 591–608, Dec. 2017.

[30] H. Yao *et al.*, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2018.

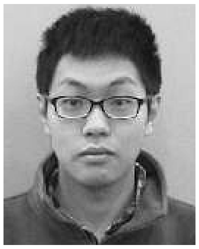
[31] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1655–1661.

[32] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018.

[33] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D.-Y. Yeung. (2018). "GaAN: Gated attention networks for learning on large and spatiotemporal graphs." [Online]. Available: https://arxiv.org/abs/1803.07294

[34] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 802–810.

Di Zang received the Ph.D. degree from the Kiel University, Germany, in 2007. She was a Post-Doctoral Researcher with the University of Minnesota, USA, in 2008. She is currently an Associate Professor with Tongji University, Shanghai, China. Her current research interests include deep learning, intelligent transportation system, and computer vision.



Jiawei Ling received the bachelor's degree from the Nanjing University of Post and Telecommunication of China. He is currently pursuing the master's degree with Tongji University, China. His current research interests include deep learning and intelligent transportation system.

Zhihua Wei received the B.S. and M.S. degrees from Tongji University in 2005 and 2000, respectively, and the dual Ph.D. degrees from Tongji University and Lyon2 University in 2010. She is currently an Associate Professor with Tongji University. Her research interests include machine learning, image processing, and data mining.

Keshuang Tang received the Ph.D. degree in transportation engineering from Nagoya University in 2008. He was a Post-Doctoral Research Fellow with The University of Tokyo. He was a Project Assistant Professor with Tohoku University. He is currently a Professor with the Department of Transportation Information and Control Engineering, Tongji University, China. His main research interests include driver behavior, signal control, and intelligent transportation systems.

Jiujun Cheng received the Ph.D. degree from the Beijing University of Posts and Telecommunications in 2006. In 2009, he was a Visiting Professor with Aalto University, Espoo, Finland. He is currently a Professor with Tongji University, Shanghai, China. His research interests span the areas of mobile computing and complex networks with a focus on mobile/Internet interworking and Internet of Vehicles.