Traffic Speed Prediction and Congestion Source Exploration: A Deep Learning Method

Jingyuan Wang†, Qian Gu†, Junjie Wu‡, Zhang Xiong†
† ACTA Engineering Research Center,
School of Computer Science and Engineering, Beihang Univeristy, Beijing 100191, China
‡ Information Systems Department,
School of Economics and Management, Beihang University, Beijing 100191, China
Email: {jywang, gqian, wujj, xiongz}@buaa.edu.cn

Abstract—**Traffic speed prediction is a long-standing and critically important topic in the area of Intelligent Transportation Systems (ITS). Recent years have witnessed the encouraging potentials of deep neural networks for real-life applications of various domains. Traffic speed prediction, however, is still in its initial stage without making full use of spatio-temporal traffic information. In light of this, in this paper, we propose a deep learning method with an Error-feedback Recurrent Convolutional Neural Network structure (eRCNN) for continuous traffic speed prediction. By integrating the spatio-temporal traffic speeds of contiguous road segments as an input matrix, eRCNN explicitly leverages the implicit correlations among nearby segments to improve the predictive accuracy. By further introducing separate error feedback neurons to the recurrent layer, eRCNN learns from prediction errors so as to meet predictive challenges rising from abrupt traffic events such as morning peaks and traffic accidents. Extensive experiments on real-life speed data of taxis running on the 2nd and 3rd ring roads of Beijing city demonstrate the strong predictive power of eRCNN in comparison to some state-of-the-art competitors. The necessity of weight pre-training using a transfer learning notion has also been testified. More interestingly, we design a novel influence function based on the deep learning model, and showcase how to leverage it to recognize the congestion sources of the ring roads in Beijing.**

I. Introduction

Traffic speed prediction, as a sub-direction of traffic prediction in the area of Intelligent Transportation Systems (ITS), has long been regarded as a critically important way for decision making in transportation navigation, travel scheduling, and traffic management. Traditional models, including autoregression methods [1] and supervised learning methods such as support vector regression [2] and artificial neural networks [3], all treat traffic speed prediction as a time-series forecasting problem, and thus run into the bottleneck gradually.

In recent years, with the rapid development of deep learning techniques, more and more researchers in ITS began to adopt deep neural networks for high-accuracy traffic prediction. The rich studies along this line, however, are mostly concerned with traffic flow and congestion predictions [4], [5]. Traffic speed prediction, therefore, is still an open problem in the deep-learning era, with two notable challenges as follows:

- How to characterize the latent interactions of road segments in traffic speeds so as to improve the predictive performance of a deep neural network?
- How to model the abrupt changes of traffic speeds in case of emerging events such as morning peaks and traffic accidents?

These indeed motivate our study. Specifically, in this paper, we propose a deep learning method using an Error-feedback Recurrent Convolutional Neural Network (eRCNN) for continuous traffic speed prediction. The novel contributions of our study are summarized as follows.

First, we take the matrix containing the spatio-temporal traffic speeds of contiguous road segments as the input of eRCNN. By this means, the complicated interactions of traffic speeds among nearby road segments can be captured by eRCNN naturally without elaborative characterization, which is crucial to the high-accuracy prediction of eRCNN.

Second, we introduce separate error-feedback neurons to the recurrent layer of eRCNN, for the purpose of capturing the prediction errors from the output layer. This empowers eRCNN the ability to model the abrupt changes in traffic speeds due to some emerging traffic events like the morning peaks and traffic accidents.

Third, we put forward a novel weight pre-training method, which adopts a transfer-learning notion by clustering similar yet contiguous road segments into a group for the generation of a same set of initial weights. This “sharing scheme” not only helps to reduce the learning process of eRCNN for every road segment, but also improves the chance of finding better optimal solutions.

Finally, we design a novel influence function based on the deep learning model, and illustrate how to leverage it to recognize the congestion sources of the ring roads in Beijing. To the best of our knowledge, we are among the earliest to explore how to learn road congestion sources from deep learning models.

Extensive experiments on real-life speed data of taxis running on the 2nd and 3rd ring roads of Beijing city demonstrate the strong predictive power of eRCNN, even with the presence of state-of-the-art competitors. The inclusion of spatio-temporal information of contiguous segments, the introduction of error-feedback neurons to the recurrent layer, and the weight pre-training of similar segments, all give a positive boost to the high accuracy of eRCNN.

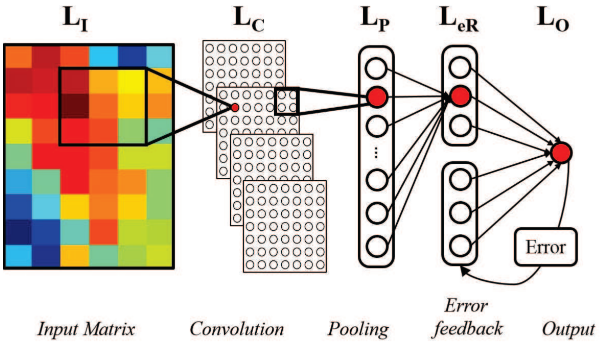


Fig. 1. The framework of the eRCNN model.

II. The eRCNN Framework

Fig. 1 shows the framework of the eRCNN model containing five network layers, including the input layer ($\mathrm{L_I}$), the convolution layer ($\mathrm{L_C}$), the pooling layer ($\mathrm{L_P}$), the error-feedback recurrent layer ($\mathrm{L_{eR}}$), and the output layer ($\mathrm{L_O}$). The function of the input layer is to organize the original traffic speed data as a spatio-temporal input matrix, which can be processed by the CNN layers of eRCNN. The function of the convolution layer and the pooling layer is to extract features from the spatio-temporal input matrix. The function of the error-feedback recurrent layer is to compensate prediction errors using predicting results of previous periods. The output layer uses a modified rectified linear unit to generate the predictions of traffic speeds.

A. The Spatio-Temporal Input Matrix

In order to exploit spatial and temporal correlation information, we construct a spatio-temporal input matrix in the input layer. Given a road segment s, we define the traffic speed of the segment s at the time t as $v_{s,t}$. When we use the proposed model to predict $v_{s,t+1}$, the spatio-temporal matrix for the input layer is defined as

$$\mathbf{V} = \begin{bmatrix} v_{s-m,t} & v_{s-m,t-1} & \cdots & v_{s-m,t-n} \\ \vdots & \vdots & \cdots & \vdots \\ v_{s-1,t} & v_{s-1,t-1} & \cdots & v_{s-1,t-n} \\ v_{s,t} & v_{s,t-1} & \cdots & v_{s,t-n} \\ v_{s+1,t} & v_{s+1,t-1} & \cdots & v_{s+1,t-n} \\ \vdots & \vdots & \cdots & \vdots \\ v_{s+m,t} & v_{s+m,t-1} & \cdots & v_{s+m,t-n} \end{bmatrix}. \quad (1)$$

As shown in Fig. 2, the column vector $\mathbf{v}_{:t}$ contains traffic speed data of all the segments in a range that m segments upstream and downstream of the segment s at the time t, and the row vector $\mathbf{v}_{s,:}$ contains traffic speed data of the segment s from time t to $t-n$. In this way, the input matrix $\mathbf{V}$ contains all the speed information that is spatially and temporally adjacent to the variate to be predicted, i.e., $v_{s,t+1}$.

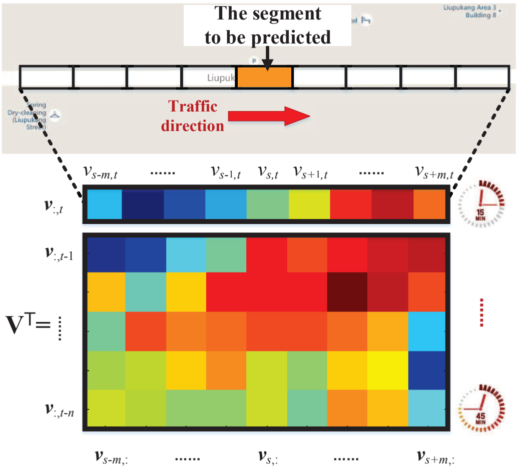


Fig. 2. The structure of the input spatio-temporal matrix.

B. The CNN-based Feature Extracting

In the eRCNN model, we adopt a CNN-based network structure to extract features from the spatio-temporal input matrix. The CNN structure contains a convolution layer and a pooling layer, and then we introduce the two layers in this subsection.

1) The Convolution Layer: The convolution layer is a core part of the CNN model [6]. The convolution layer connects the spatio-temporal input matrix with several trainable filters, with each being a $i \times i$ weight matrix. We define the k-th filter as $\mathbf{W}_k^{(C)}$. The convolution layer uses the $\mathbf{W}_k^{(C)}$ to zigzag scan the input matrix to calculate a convolution neuron matrix. The (p, q) element of the convolution neuron matrix generated by the filter k is calculated by

$$c_k^{p,q} = \sigma\left(b_k + \sum_{x=0}^{i}\sum_{y=0}^{i} w_k^{x,y} m^{p+x,q+y}\right), \quad (2)$$

where b_k is a bias for the filter k, $w_k^{x,y}$ is the (x, y) element of $\mathbf{W}_k^{(C)}$, $m^{p+x,q+y}$ is the $(p+x, q+y)$ element of the spatio-temporal matrix $\mathbf{V}$, and $\sigma(\cdot)$ is a sigmoid activation function defined as:

$$\sigma(x) = \frac{1}{1+e^x}.$$

More details about the implementation of the CNN's convolution layer could be found in [7].

2) The Pooling Layer: The pooling layer is another important component of the CNN model, which is used to reduce the dimension of the convolution neuron matrix through an average down sampling method. In the proposed eRCNN model, the pooling layer divides the convolution neuron matrix into $j \times j$ disjoint regions, and uses the averages of each region to represent the characteristic of the convolution neurons in the region. Through the processing of the pooling layer, the dimension of the spatio-temporal matrix is reduced as about 1/($j \times j$) of its original size. The output of the pooling layer is a feature vector generated through vectoring the down sampled convolution neuron matrix, which is denoted as $\mathbf{p}$.

C. The Error-Feedback Recurrent Layer

An important characteristic of traffic speed data is the abrupt change of speed within a short time period. For example, during the beginning 30 minutes of morning peaks, the traffic speed of the Beijing ring roads could drop from 70km/h to 30km/h; while after a rear-end collision traffic accident, the traffic speed could drop from 50km/h to 20km/h. In general, it is hard to predict the traffic conditions with these abrupt speed changes using traditional neural network structures. In this way, we introduce an error-feedback recurrent layer to improve prediction performance of our model in the above scenarios.

In the error-feedback recurrent layer, a group of neurons are connected with the feature vector $\mathbf{p}$ that is generated by the pooling layer. The k-th neuron r_k is fully connected with all the elements of $\mathbf{p}$ through a sigmoid activation function, i.e.,

$$r_k^{(R)} = \sigma\left(\mathbf{w}_k^{(\mathrm{R})}\mathbf{p} + b_k^{(\mathrm{R})}\right), \tag{3}$$

where $\mathbf{w}_k^{(\mathrm{R})}$ is the connection weight vector for the neuron r_k, and $b_k^{(\mathrm{R})}$ is the bias.

In the traditional RNN model [8], r_k still needs to be connected with the hidden layer neurons of the last prediction steps, i.e.,

$$r_k^{(R)}(t) = \mathrm{sigmoid}\left(\mathbf{w}_k^{(\mathrm{R})}\mathbf{p} + \tilde{\mathbf{w}}_k\mathbf{r}(t-1) + b_k^{(\mathrm{R})}\right), \tag{4}$$

where $\mathbf{r}(t-1)$ is the neuron vector of the $t-1$ step, and $\tilde{\mathbf{w}}_k$ is the corresponding wight vector. However, this network structure does not consider the prediction errors, which is indeed useful in the scenarios of abrupt speed changes. Specifically, if we have the information about the prediction errors at the previous steps, we can design a model to compensate the prediction error at the current step.

In order to overcome the limitations of RNN, we introduce a group of error-feedback neurons in the recurrent layer. The value of the k-th error-feedback neuron $r_k^{(E)}$ at the t prediction step is defined as:

$$r_k^{(E)}(t) = \mathrm{sigmoid}\left(\mathbf{w}_k^{(E)}\mathbf{e}(t-1) + b_k^{(E)}\right), \tag{5}$$

where $b_k^{(E)}$ is a bias, $\mathbf{w}_k^{(E)}$ is a weight needs to train. The vector $\mathbf{e}(t-1)$ in Eq. (5) is a prediction error vector defined as

$$\mathbf{e}(t) = \left[y(t-1) - o(t-1), \ldots, y(t-l) - o(t-l)\right], \tag{6}$$

where $y(t-l)$ is the real traffic speed at the step $t-l$, and $o(t-l)$ is the predicted speed at the step $t-l$.

The output of the error-feedback recurrent layer is a combination of the regular neurons $r^{(R)}$ and the error-feedback neurons $r^{(E)}$, i.e.,

$$\mathbf{r} = [\mathbf{r}^{(R)}; \mathbf{r}^{(E)}]. \tag{7}$$

In the error-feedback recurrent layer, we do not connect the input of the current step and the error-feedback of the previous steps together in the same group of neurons as in traditional RNN. On the contrary, the input is connected into separate neuron groups. This is because the current input and the recurrent input in our model have different characteristics.

D. The Output Layer

Considering the error-feedback recurrent layer, the output neurons $\mathbf{r}$ is then used as an input, and the output layer generates a final prediction value as

$$o = \sigma\left(\mathbf{w}^{(OR)}\mathbf{r}^{(R)} + \mathbf{w}^{(OE)}\mathbf{r}^{(E)} + b^{(O)}\right), \tag{8}$$

where $\mathbf{w}^{(OR)}$, $\mathbf{w}^{(OE)}$, and $b^{(O)}$ are the weights and bias of the output layer. In the output layer, we adopts a modified $ReLU$ function as the activation function, which is defined as

$$\sigma(x) = \begin{cases} 0 & \text{if} \quad x \le 0 \\ x & \text{if} \quad 0 < x < 1 \\ 1 & \text{if} \quad x \ge 1 \end{cases}. \tag{9}$$

The output of Eq. (8) can be regarded as a linear combination of the traffic speed prediction (generated by the input of current step) and the error compensation (generated by previous steps).

Because the output of Eq. (9) is in the range of [0, 1], we re-scale the traffic speed of road segments into the same range. According to the actual situation of urban traffic, we adopt the following reflect function to re-scale the traffic speed data:

$$\psi(x) = \begin{cases} 1 & \text{if} \quad x \ge 80 \text{ km/h} \\ \frac{80-x}{70} & \text{if} \quad x \in [10, 80] \text{ km/h} \\ 0 & \text{if} \quad x \le 10 \text{ km/h} \end{cases}. \tag{10}$$

In fact, in order to keep the input and output at the same scale, the traffic speeds in the input spatio-temporal matrix $\mathbf{V}$ are also re-scaled by the function in Eq. (10).

III. Network Training

A. Parameters Training

The parameters need to be trained in the eRCNN model include the weight matrix set $\mathbf{W}^{(C)}$ and the bias set $b^{(C)}$ of the convolution layer, the weight vector sets $\mathbf{w}^{(R)}$, $\mathbf{w}^{(E)}$ and the bias sets $b^{(R)}$, $b^{(E)}$ of the error feedback recurrent layer, the weight vector $\mathbf{w}^{(O)} = [\mathbf{w}^{(OR)}; \mathbf{w}^{(OE)}]$, and the bias $b^{(O)}$ of the output layer. For the sake of simplicity, we introduce θ to represent all the parameters.

$$\theta = \left\{\mathbf{W}^{(C)}, \mathbf{w}^{(R)}, \mathbf{w}^{(E)}, \mathbf{w}^{(O)}, b^{(C)}, b^{(R)}, b^{(E)}, b^{(O)}\right\}. \tag{11}$$

The parameter training is achieved by a mini-batch stochastic gradient descent (SGD) method. For a road segment, the objective of parameters training is to minimize the squared error for all the training samples, i.e.,

$$L = \frac{1}{2}\sum_k (y_k - o_k)^2. \tag{12}$$

In the mini-batch SGD, the training samples are divided into several mini-batches. For a mini-batch, we calculate the partial derivatives of L with respect to all the parameters, and then update the parameters using the following equation,

$$\theta \leftarrow \theta - \alpha \frac{\partial L}{\partial \theta}, \tag{13}$$

where α is an adjustable learning rate.

The partial derivatives of L to the parameters are calculated by the error back propagation (BP) algorithm. For a mini-batch with m samples, the partial derivatives of L with respect to the output layer parameters $\mathbf{w}^{(O)}$ and $b^{(O)}$ are

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{w}^{(O)}} &= \frac{1}{m} \sum_{m} d^{(O)}(t) [\mathbf{r}^{(R)}; \mathbf{r}^{(E)}], \\ \frac{\partial L}{\partial b^{(O)}} &= \frac{1}{m} \sum_{m} d^{(O)}(t), \end{aligned} \tag{14}$$

where $d^{(O)}(t)$ is the error propagated from the output layer at the prediction step t. For a given road segment, we define $o(t)$ as the prediction output at the step t, and $y(t)$ is the corresponding real traffic speed, and then $d^{(O)}(t)$ is calculated as

$$d^{(O)}(t) = \delta\left(o(t)\right)\left(y(t) - o(t)\right) - \sum_{k} \tilde{\mathbf{w}}_k^{(E)} \mathbf{d}_k^{(E)}(t), \tag{15}$$

where the function $\delta(x)$ is with a form of

$$\delta(x) = \begin{cases} 1 & \text{if } 0 < x < 1 \\ 0 & \text{if } x = 0 \text{ or } 1 \end{cases}. \tag{16}$$

In the second term of the Eq.(15), $\tilde{\mathbf{w}}_k^{(E)}$ is an inverted form of the weight vector $\mathbf{w}_k^{(E)}$, i.e.,

$$\tilde{\mathbf{w}}_k^{(E)} = [w_k^{(E)}(l), \ldots, w_k^{(E)}(1)], \tag{17}$$

and $\mathbf{d}_k^{(E)}(t)$ is the error propagated from the prediction time $t+1$ to $t+l$, i.e.,

$$\mathbf{d}_k^{(E)}(t) = [d_k^{(E)}(t+1), \ldots, d_k^{(E)}(t+l)]. \tag{18}$$

For a given time t, $d_k^{(E)}(t)$ is calculated as

$$d_k^{(E)}(t) = d_k^{(O)}(t) w_k^{(OE)} r_k^{(E)}(t)(1 - r_k^{(E)}(t)). \tag{19}$$

Moreover, we calculate the partial derivatives of L to the weight parameters of the error-feedback recurrent layer as

$$\frac{\partial L}{\partial \mathbf{w}_k^{(E)}} = \frac{1}{m} \sum_{m} d_k^{(E)}(t) \mathbf{e}(t-1), \tag{20}$$

and

$$\frac{\partial L}{\partial \mathbf{w}_k^{(R)}} = \frac{1}{m} \sum_{m} d_k^{(R)}(t) \mathbf{p}, \tag{21}$$

where $d_k^{(R)}$ at the time t is calculated as

$$d_k^{(R)}(t) = d_k^{(O)}(t) w_k^{(OR)} r_k^{(R)}(t)(1 - r_k^{(R)}(t)). \tag{22}$$

Algorithm 1 The segments clustering algorithm.

```
Require: A segment set $S = \{s_1, s_2, \ldots, s_m\}$ that includes $m$ segments of a road. A Pearson correlation coefficient threshold $P$.
Initialization: The segment cluster $H_i$, and $i = 0$.
while not all segments in the set $S$ are clustered do
    $s_0 \leftarrow$ a segment that is not clustered.
    $H_i \leftarrow \{s_0\}$, $n \leftarrow 1$.
    $s_x \leftarrow$ a segment that is contiguous with the segments in $H_i$.
    while $\frac{1}{n} \sum_{\forall s \in H_i} \text{Pearson}(s_x, s) > P$ do
        $H_i \leftarrow \{H_i, s_x\}$, $n \leftarrow n + 1$.
        $s_x \leftarrow$ a segment that is contiguous with the segments in $H_i$.
    end while
    $i \leftarrow i + 1$.
end while
Output: The segment clusters $H_0, H_1, \ldots, H_i$.
```

The partial derivatives to bias parameters for the error-feedback recurrent layer is calculated as

$$\frac{\partial L}{\partial b_k^{(R)}} = \frac{1}{m} \sum_{m} d_k^{(R)}(t). \tag{23}$$

The partial derivative of weight set $\mathbf{W}^{(C)}$ and bias set $\mathbf{b}^{(C)}$ in the convolution layer is calculated according to the standard CNN BP algorithm [7], which will not be elaborated here.

B. Pre-Training and Fine-Tuning eRCNN

Since a road is divided into several segments, different segments may have different traffic speed variation patterns. Thus, we need to train special model parameters for each segment. However, in the real situation, the training data for a specific segment is limited in the speed samples. If the training data is not enough, the eRCNN model may suffer from over fitting problem. In order to prevent the over fitting problem and take full advantages of the training data of all the road segments, we develop an approach to cluster road segments as several subsets, and use all the speed data of the segments in the same subset to pre-train an eRCNN model.

The clustering algorithm used here is a Pearson correlation coefficient based algorithm. For a road segment pair s_i and s_j, the Pearson correlation coefficient is calculated as

$$\text{Pearson}(s_i, s_j) = \frac{Cov(\mathbf{v}_{i,:}, \mathbf{v}_{j,:})}{\sqrt{Var(\mathbf{v}_{i,:}) Var(\mathbf{v}_{j,:})}}, \tag{24}$$

where $\mathbf{v}_{i,:}$ and $\mathbf{v}_{j,:}$ are traffic speed series of the road segments s_i and s_j, respectively. Using the Pearson correlation coefficient as a similarity measurement, the clustering algorithm is presented in Algorithm 1. To be more specific, Algorithm 1 clusters road segments that are contiguous and with Pearson correlation coefficients higher than a threshold as a same set. The segments in the same set share their traffic speed as a pre-training data set. Through this approach, we transfer knowledge of other segments into the model of a certain segment.

TABLE I
THE FORMAT OF FLOATING CAR RECORDS.

Field	Definition
ID	The unique ID of a taxi.
TIME	The sample time stamp of this record.
LON	The current longitude of the taxi.
LAT	The current latitude of the taxi.
DIR	The current driving direction of the taxi.
STATUS	Whether the taxi is carrying a passenger.

Furthermore, using parameters of the pre-training model as the initial values of the parameters, we further fine-tune the eRCNN model for each segment by utilizing the local spatio-temporal data. Specifically, we divide the 24 hours of one day into seven time ranges, i.e., [0:00, 6:00], [6:00, 9:00], [9:00, 12:00], [12:00, 15:00], [15:00, 18:00], [18:00, 21:00], and [21:00, 0:00]. For a segment in a certain time range, we fine tune the special parameters by using the speed data of the segment in the given time range based on the pre-trained model.

IV. EXPERIMENTS

A. Data Description

In our experiments, we tested eRCNN over two very important roads in the Beijing city, i.e., the 2nd ring road and the 3rd ring road. These two roads encircle the center of Beijing. According to the Beijing Municipal Commission of Transport, the average traffic flow carried by the two roads goes beyond 200,000 cars every day, which occupies about 10% of the total traffic flow in Beijing downtown area[1].

The length of the 2nd and 3rd ring roads are 32 and 48 km, respectively. We set the average length of each road segment to be 400 meters, which results in 80 and 122 road segments for the two roads. Moreover, the traffic speed of a road segment is collected from the GPS terminals of taxis driving on the segment. In Beijing, about 60,000 taxis are called floating cars, which are installed with GPS terminals and used as floating senors to collect traffic speeds of urban roads. The data records collected from floating cars contain information about the time stamp, location, speed, direction, and status of a driving taxi, as specified in Table I. Such records are collected every minute for a driving taxi, and the total data size generated each day is about 5GB. In the experiment, we exploit a road-map matching algorithm proposed in [9] to match the floating car records into urban roads, and further calculate the average speed of segments in the 2nd and 3rd ring roads. In the data collecting process, the traffic speed of a segment is updated every 5 minutes. Fig. 3 shows an example of one-day traffic speed variations of a road segment in the 2nd ring road. As we can see from the figure, the average traffic speed fluctuates dramatically in a day.

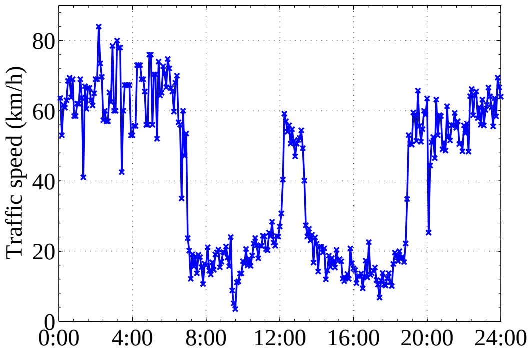


Fig. 3. The traffic speed fluctuations of a road segment in the 2nd ring road.

The data set used in this experiment was collected from the 25 weekdays in November 2013. The data of the first 20 weekdays were used as the training set, and the remaining five days as the test set.

B. Evaluation Metrics

We adopt three widely used metrics to evaluate the performance of prediction models, including Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE), which are defined as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} \left| v_i - \hat{v}_i \right|,$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n} \frac{\left| v_i - \hat{v}_i, \right|}{v_i},$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (v_i - \hat{v}_i)^2},$$

where v_i is the observed speed, and $\hat{v}_i$ is the predicted one.

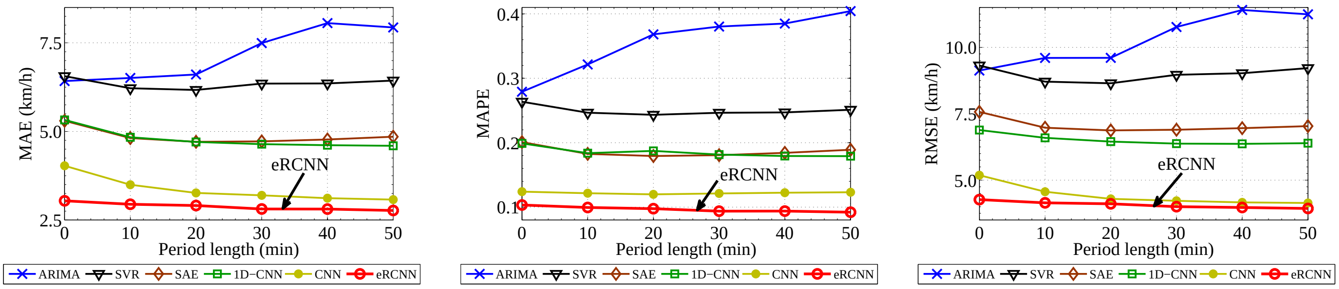
C. Benchmarks

We compared the performance of our model with the following five benchmark methods: 1) Auto Regression Integrated Moving Average (ARIMA) [1]. 2) Support Vector Regression (SVR) [10]. 3) Stacked Auto Encoders (SAE) [5]. 4) 1D Convolution Neural Network (1D-CNN). The network structure of the 1D-CNN is the same as the CNN part of eRCNN, but the input matrix reduces to the time series of the traffic speeds of the segment to be predicted. 5) Convolution Neural Network (CNN). The network structure of the CNN benchmark is the same as eRCNN, except that the CNN removes the error feedback procedure. Note that 1D-CNN is used as benchmark to test the effectiveness of the spatio-temporal input matrix for eRCNN, and CNN is used to test the performance of the error feedback scheme of eRCNN.

D. Overall Performance

We compared the performance of eRCNN with the benchmark methods in two different experimental scenarios. In the

[1] http://www.bjjtw.gov.cn/

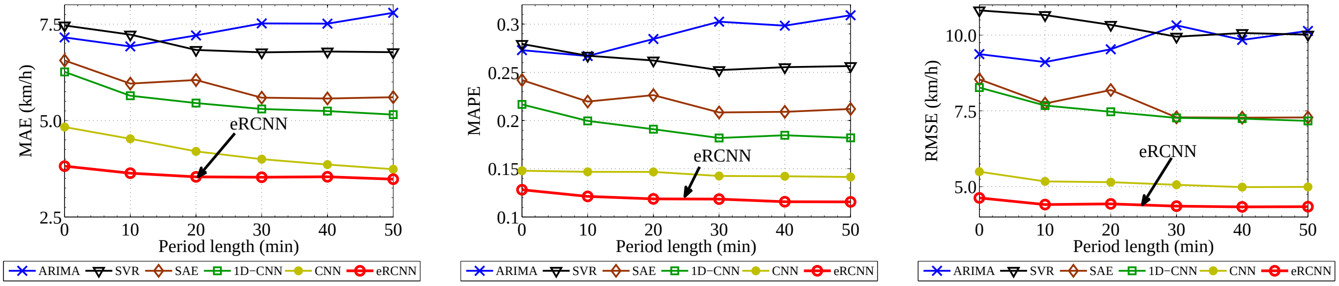


(a) MAE Comparison

(b) MAPE Comparison

(c) RMSE Comparison

Fig. 4. Prediction performance on the 2nd ring road with varying period lengths (Scenario I).



(a) MAE Comparison

(b) MAPE Comparison

(c) RMSE Comparison

Fig. 5. Prediction performance on the 3rd ring road with varying period lengths (Scenario I).

first scenario, we predict the the average traffic speed of a road segment in a following period, with the length varying from 5 to 30 minutes. Fig. 4 and Fig. 5 show the comparative performances of eRCNN and the benchmarks for the 2nd and 3rd ring roads, respectively. As shown in the figures, the prediction error of eRCNN is obviously smaller than ARIMA, SVR, SAE, and 1D-CNN in terms of the three evaluation metrics. Although the performance of CNN is comparable to eRCNN, the prediction error is still greater than eRCNN. Generally we can see that the prediction performance becomes better when the length of the prediction period increases. Intuitively, this may be due to the fact that the traffic speed of a segment becomes smoother when the average period length increases. The results indicate that eRCNN can effectively extract the spatio-temporal features from the traffic speed data of contiguous road segments, and the introduction of the error-feedback recurrent layer is indeed positive for eRCNN.

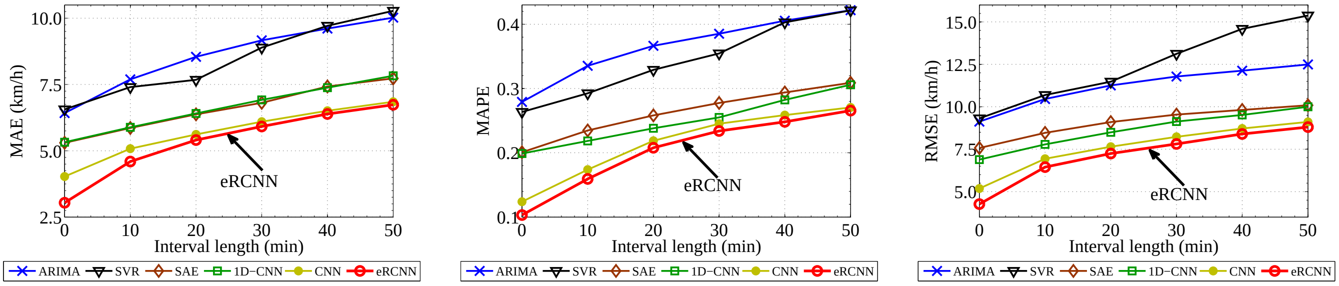
In the second scenario, we aim to predict the traffic speed of a segment after a given time interval, with the interval length varying from 0 to 50 minutes. Fig. 6 and Fig. 7 show the prediction errors of eRCNN in comparison with the benchmarks for the 2nd and 3rd ring roads, respectively. As shown in the figures, eRCNN achieved the best performances compared with other methods. Note that since the correlation between the traffic speed of two adjacent periods decreases as the interval increases, the prediction performances become worse with the increase of the interval length.

In summary, from the above experimental results, we find that eRCNN achieves the best performance compared with the state-of-the-arts. The inclusion of spatio-temporal information of contiguous segments and the error-feedback neurons to the network structure is the key for success.

E. Performance for Individual Road Segment

To further demonstrate the advantages of eRCNN, we took a closer look at the prediction errors of every road segments with CNN, SAE and SVR as benchmarks. Fig. 8(a) shows the comparative performances on each segment of the inner ring (in clockwise direction) of the 2nd ring road. In the experiment, the prediction period is set as 5 minutes and the interval is 0 minutes. As depicted in the figure, the prediction errors for different segments indeed vary greatly. We can see that for all the prediction methods, the predictability of the segment #30-#50 is better than other segments in general. On the contrast, the predictability of segment #7-#15, #20-#25, and #62-#80 is much poorer.

As shown in the figure, the performances of SAE and SVR degrade severely for the low predictability segments, whereas the performances of eRCNN and CNN remain stable across all the segments. Fig. 8(a) shows the similar experimental results for the inner ring of the 3rd ring road, with the same settings to the time period and interval. To sum up, the results in Fig. 8 indicate the robustness of eRCNN empowered by the learning scheme from the spatio-temporal speed matrix of nearby segments.

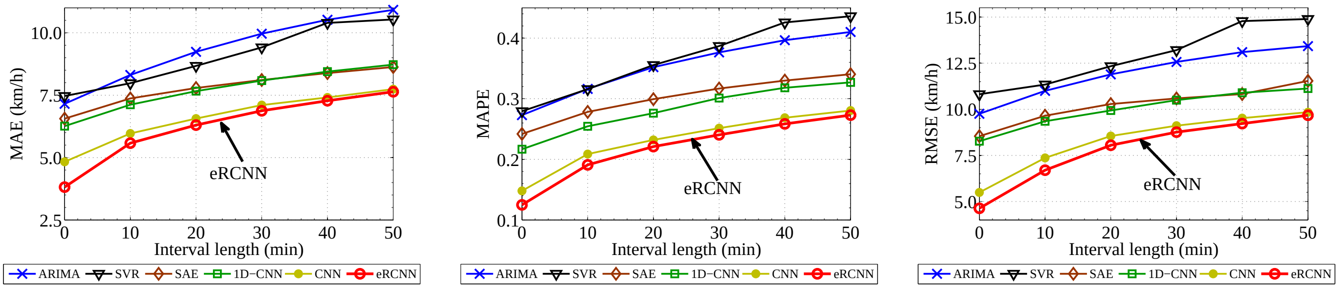


(a) MAE Comparison

(b) MAPE Comparison

(c) RMSE Comparison

Fig. 6. Prediction performance on the 2nd ring road with varying interval lengths (Scenario II).

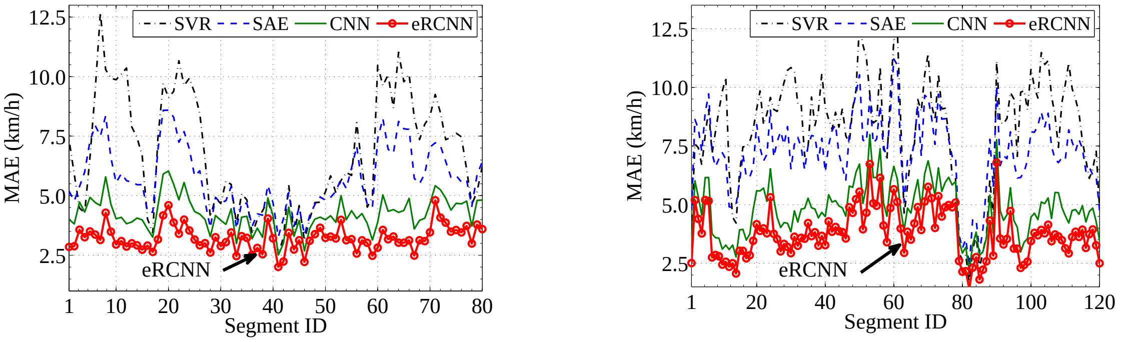


(a) MAE Comparison

(b) MAPE Comparison

(c) RMSE Comparison

Fig. 7. Prediction performance on the 3rd ring road with varying interval lengths (Scenario II).



(a) The 2nd Ring Road

(b) The 3rd Ring Road

Fig. 8. Prediction performances across all road segments.

F. Performance with Time Variation

Here, we study the prediction performances of eRCNN in different time intervals of a day with substantially different traffic conditions. We randomly select a road segment located in the inner loop of the 2nd ring road as the targeted sample, and predict its traffic speeds during the time interval from 18:30 to 21:30 on November 24, 2013. The prediction period is set as 5 minutes and the interval is set as 0.

Fig. 9 demonstrates the real traffic speed and the prediction results from eRCNN and CNN. As can be seen from Fig. 9, from 19:00 to 19:30, the traffic recovers from the last traffic jam of the night peak. While around 20:20, the traffic speed decreases again due to a small accident, and the traffic recovers to normal before 21:00. In general, the traffic speed changes abruptly during both of the two periods. As can be seen from the figure, our proposed eRCNN successfully captures the abrupt changes in speeds and the curve of predictions exactly matches the real values of traffic speeds, while the CNN model does not effectively follow the abrupt changes of traffic speeds. This well demonstrates the necessity of introducing the error feedback scheme to the recurrent layer of eRCNN.

To further demonstrate the statistical property of the error-feedback scheme, Fig. 10 plots the cumulative distribution functions of the absolute prediction error in the 19:00-19:30 time period of all the testing days and all the road segments for eRCNN and CNN, respectively. The prediction period is

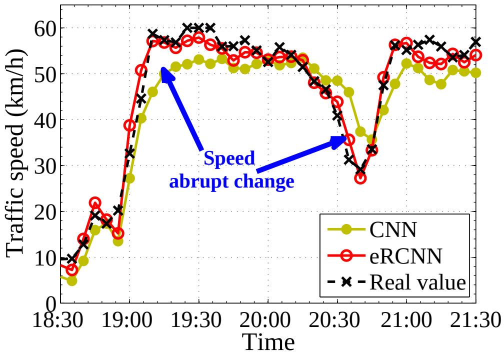


Fig. 9. Traffic speed prediction with time variation.

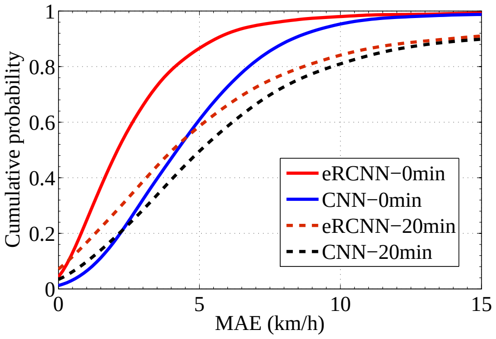


Fig. 10. CDF of predictive error during 7:00-7:30.

5-min and the intervals are 0-min and 20-min, respectively. As shown in Fig. 10, the prediction error of eRCNN is obviously smaller than CNN, which indicates the great improvements from the error-feedback scheme in eRCNN, especially when facing traffic fluctuations.

In summary, the experimental results testified the effectiveness of introducing separate error-feedback neurons to eRCNN when predicting traffic speeds with abrupt changes.

G. Performance with Weight Pre-Training

As discussed in Section III-B, we develop a pre-training method by clustering the road segments. To evaluate the effectiveness of this method, we compared the prediction results of eRCNN under three conditions, i.e., prediction with pre-training, prediction without pre-training, and prediction with only pre-training, in the time period between 6:00 to 21:00. We set the prediction period to 5-min and the intervals to 0-min and 20-min, respectively.

As shown in Fig. 11, the prediction results without pre-training and with only pre-training fluctuate drastically. Particularly, during the morning peak (7:00-9:00) and evening peak (17:00-19:00), the prediction errors for the two cases are much higher than that of other time periods. Nevertheless, the prediction results with pre-training remain stable across all the time ranges, and the prediction errors are significantly lower than the other two cases. This well demonstrates that eRCNN is greatly enhanced by the pre-training scheme even facing the drastic speed changes during the morning and evening peaks.

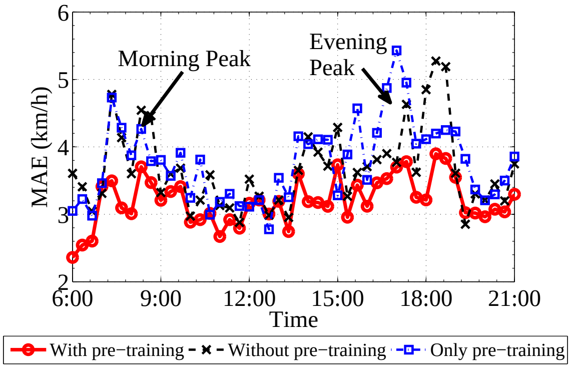


Fig. 11. Illustration of the effectiveness of pre-training.

V. Importance Analysis for Road Segments

In this section, we introduce a very useful application of our traffic speed prediction model: the importance analysis for road segments. In order to explain the concept of segment importance, we first give a formalized definition of the *influence* between segments. For a segment j whose traffic speed is influenced by the traffic speed of the segment i, we define

$$v_j = f(v_i), \tag{25}$$

where v_i and v_j are traffic speeds of segment i and j, respectively. Based on the relations described by Eq. (25), we define the *influence* of segment i to segment j as the derivative of v_j to v_i, i.e., $\forall\ \varepsilon > 0$

$$I_i(j) = \frac{\mathbf{d}f(v_i)}{\mathbf{d}v_i} = \lim_{\varepsilon \to 0} \frac{f(v_i) - f(v_i - \varepsilon)}{\varepsilon}. \tag{26}$$

For the eRCNN model, the network structure is a function $o = f(\mathbf{V})$, which models the relations between predicted speed o of a segment and real traffic speeds $\mathbf{V}$ of its contiguous segments. Because eRCNN achieved very accurate prediction performance, we can use $\frac{\partial o}{\partial \mathbf{V}}$ to approximate the influence of the contiguous segments to the predicted segment.

The calculation of $\frac{\partial o}{\partial \mathbf{V}}$ is given as follows. According to Eq. (2), in the convolution layer, the partial derivative of the element (p, q) in the neuron matrix for the k-th filter to the input matrix $\mathbf{V}$ is

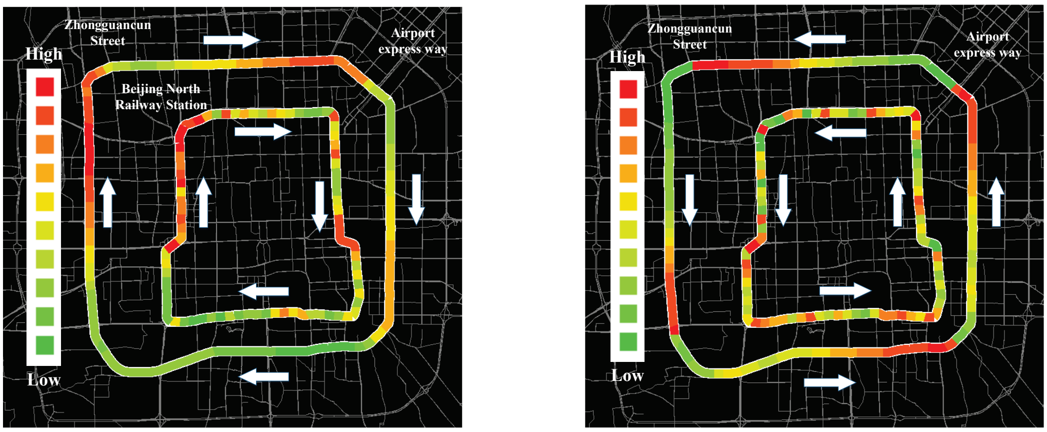
$$\frac{\partial c_k^{p,q}}{\partial \mathbf{V}} = c_k^{p,q}(1 - c_k^{p,q})\mathbf{W}_k^{(C)}. \tag{27}$$

In the pooling layer, the partial derivative of the pool output $p_k^{i,j}$ to the matrix $\mathbf{V}$ is an average of $\partial c_k^{p,q}/\partial \mathbf{V}$, i.e.,

$$\frac{\partial p_k^{i,j}}{\partial \mathbf{V}} = \frac{1}{4} \sum_{m=2i-1}^{2i} \sum_{n=2j-1}^{2j} \frac{\partial c_k^{m,n}}{\partial \mathbf{V}}. \tag{28}$$

The error-feedback recurrent layer contains two kinds of neurons: the regular neuron and the error-feedback neuron. For the sake of reducing complexity, we ignore the influence of error feedback neurons and only consider the regular neurons. We define an intermediate variable as

$$\frac{\partial \mathbf{p}_k}{\partial \mathbf{V}} = \sum_i \sum_j w_{i,j,k}^{(R)} \frac{\partial p_k^{i,j}}{\partial \mathbf{V}}, \tag{29}$$



(a) The Inner Loop (clockwise)

(b) The Outer Loop (anti-clockwise)

Fig. 12. The importance of segments in the 2nd and 3rd ring roads.

where $w_{i,j,k}^{(R)}$ is the element of $\mathbf{w}^{(R)}$ that corresponds to $p_k^{i,j}$. According to Eq. (3), the complete form of the partial derivative for a regular recurrent neuron is given by

$$\frac{\partial r^{(R)}}{\partial \mathbf{V}} = r^{(R)}(1 - r^{(R)}) \sum_k \frac{\partial \mathbf{p}_k}{\partial \mathbf{V}}. \tag{30}$$

We define the partial derivative vector with the N regular neurons in the recurrent layer as

$$\frac{\partial \mathbf{r}^{(R)}}{\partial \mathbf{V}} = \left[\frac{\partial r_1^{(R)}}{\partial \mathbf{V}}, \ldots, \frac{\partial r_N^{(R)}}{\partial \mathbf{V}} \right]^\top. \tag{31}$$

According the Eq. (8), we obtain the partial derivative of the output variable o to the input matrix $\mathbf{V}$ as

$$\frac{\partial o}{\partial \mathbf{V}} = \delta(o)\mathbf{w}^{(OR)} \frac{\partial \mathbf{r}^{(R)}}{\partial \mathbf{V}}. \tag{32}$$

According to definition of the input matrix $\mathbf{V}$ in Eq. (1), the element (1,1) of $\frac{\partial o}{\partial \mathbf{V}}$ is $\frac{\partial o_{s,t+1}}{\partial v_{s-m,t}}$, which denotes the derivative of predicted speed of segment s at time $t+1$ to the speed of the segment $s-m$ at time t. According to the definition of segment influence, we approximately calculate the influence of the segment $s-m$ to s at time t as

$$I_{s-m,t}(s) = \sum_{k=t}^{t-n} \left| \frac{\partial o_{s,t+1}}{\partial v_{s-m,k}} \right|. \tag{33}$$

We define the *importance* of the segment k as its influence to all segments in the same road with it, i.e.,

$$\text{Importance}_k = \sum_t \sum_{s \neq k} \mathrm{I}_{k,t}(s). \tag{34}$$

According to this definition of segment importance, the segments with high importance have high influence to the traffic speeds of other segments. These high important segments could be considered as sources of traffic congestion.

In order to verify the effectiveness of the above-mentioned importance analysis method, we again adopt the 2nd and 3rd ring roads as the demonstrative examples. We use 5-min period and 0-min interval prediction experiment results to calculate the importance of the inner and outer loop segments of the two ring roads. The importance of the segments is demonstrated in the city map of Beijing as shown in Fig. 12. Obviously, the high important road segments are mostly located near the corners of the ring roads, for both of the outer and inner loops. This is possibly because the entrances and exits of the ring roads are concentrated near the corners, which also connect other important roads. For example, the northeastern corner of the two ring roads connects one of the most important highways to leave Beijing and also the expressway of Beijing airport, the northwest corner of the 3rd ring road connects with the Zhongguanchun Street (The silicon valley of China), and the northwest corner of the 2nd ring road connects with the North Railway Station. In a nutshell, the results explicitly detect the key congestion source in the ring roads, which can help the municipal administrators to make better decision in urban planning and resolving the traffic congestion.

VI. Related Work

A widely used method for traffic speed prediction is the autoregressive integrated moving average (ARIMA) model [1]. After the birth of the BoxCJenkins time-series analyzing method [11], many ARIMA based variants were proposed to improve the traffic predicting power, including Kohonen-ARIMA [12], ARIMA with explanatory variables (ARIMAX) [13], and seasonal ARIMA (SARIMA) [14].

In recent years, great attention is being paid to supervised learning methods for traffic prediction. Support vector regression (SVR) and artificial neural networks (ANN) are the two kinds of particular interests. For instance, [2] proposed a SVR based method to predict traffic speed. [15] proposed an online learning weighted support-vector regression (OLWSVR) to predict short-term traffic flow. As to ANN, Ref. [3] applied artificial neural networks to predict the speeds on two-lane rural highways. [16] proposed a fuzzy neural network to analyze road traffic. In [17] and [18], a genetic approach is proposed to optimize neural networks for short-term traffic flow prediction. Other learning based methods include the distribution enhanced linear regression [19], the hidden Markov model based prediction method [20], and the Gaussian process-based method [21]. The predictability of road traffic and congestion in urban areas is studied in [22].

With the booming of deep learning techniques [23], [24], some ITS researches begin to adopt deep neural network models as an effective traffic prediction tool. Ma et. al. [25] adopted a RNN-RBM model to predict congestion evolution in a large-scale transportation network. [4] proposed a deep belief network model with shared representation for traffic flow prediction, while [5] adopted a SAE model for this purpose.

To further enhance predictive performance, involving historical and spatio-temporal information becomes a promising trend in traffic prediction. For instance, Ref. [26] claimed that traditional prediction approaches that treat traffic data streams as generic time series might fail to forecast traffic during peak hours and in case of events, and proposed the H-ARIMA+ method to incorporate historical traffic data for traffic prediction. In [10], spatio-temporal trends were introduced to the SVR model to facilitate large-scale traffic speed prediction. In [27], a non-negative matrix factorization based latent space model was introduced to predict time-varying traffic in networked roads in a large spatial area. [28] proposed a tensor based model to predict travel-time through exploiting spatio-temporal information.

Summary: Despite of the abundant research in traffic prediction, to our best knowledge, our work is among the earliest to allow integrating both spatio-temporal and prediction-error information into deep neural networks for traffic speed prediction of high accuracy. Moreover, our study sheds light on how to learn road segment importance from deep learning models.

VII. Conclusion

In this paper, we proposed a novel deep learning method called eRCNN for traffic speed prediction of high accuracy. An error-feedback recurrent covolutional neural network is carefully designed so as to incorporate the spatio-temporal speed information of contiguous road segments as well as to perceive the prediction errors stemming from the abrupt fluctuations of traffic speeds. Experiments on real-world traffic speed data of the ring roads of Beijing city demonstrate the advantages of eRCNN to the excellent competitors. In particular, we illustrate how to explore the congestion sources from eRCNN.

References

[1] M. S. Ahmed and A. R. Cook, *Analysis of freeway traffic time-series data by using Box-Jenkins techniques*, 1979, no. 722.

[2] C.-H. Wu, J.-M. Ho, and D.-T. Lee, “Travel-time prediction with support vector regression,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 5, no. 4, pp. 276–281, 2004.

[3] J. McFadden, W.-T. Yang, and S. Durrans, “Application of artificial neural networks to predict speeds on two-lane rural highways,” *Transportation Research Record: Journal of the Transportation Research Board*, no. 1751, pp. 9–17, 2001.

[4] W. Huang, G. Song, H. Hong, and K. Xie, “Deep architecture for traffic flow prediction: deep belief networks with multitask learning,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 15, no. 5, pp. 2191–2201, 2014.

[5] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, “Traffic flow prediction with big data: A deep learning approach,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 16, no. 2, pp. 865–873, 2015.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[8] L. Medsker and L. Jain, “Recurrent neural networks,” *Design and Applications*, 2001.

[9] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang, “Map-matching for low-sampling-rate gps trajectories,” in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2009, pp. 352–361.

[10] M. T. Asif, J. Dauwels, C. Y. Goh, A. Oran, E. Fathi, M. Xu, M. M. Dhanya, N. Mitrovic, and P. Jaillet, “Spatiotemporal patterns in large-scale traffic speed prediction,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 15, no. 2, pp. 794–804, 2014.

[11] M. Levin and Y.-D. Tsao, “On forecasting freeway occupancies and volumes (abridgment),” *Transportation Research Record*, no. 773, 1980.

[12] M. Van Der Voort, M. Dougherty, and S. Watson, “Combining kohonen maps with arima time series models to forecast traffic flow,” *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307–318, 1996.

[13] S. Lee and D. Fambro, “Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting,” *Transportation Research Record: Journal of the Transportation Research Board*, no. 1678, pp. 179–188, 1999.

[14] B. M. Williams and L. A. Hoel, “Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results,” *Journal of transportation engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[15] Y.-S. Jeong, Y.-J. Byon, M. Mendonca Castro-Neto, and S. M. Easa, “Supervised weighting-online learning algorithm for short-term traffic flow prediction,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 4, pp. 1700–1707, 2013.

[16] C. Quek, M. Pasquier, and B. B. S. Lim, “Pop-traffic: a novel fuzzy neural approach to road traffic analysis and prediction,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 7, no. 2, pp. 133–146, 2006.

[17] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, “Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach,” *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 3, pp. 211–234, 2005.

[18] P. Lingras and P. Mountford, “Time delay neural networks designed using genetic algorithms for short term inter-city traffic forecasting,” in *Engineering of Intelligent Systems*. Springer, 2001, pp. 290–299.

[19] G. Ristanoski, W. Liu, and J. Bailey, “Time series forecasting using distribution enhanced linear regression,” in *Advances in Knowledge Discovery and Data Mining*. Springer, 2013, pp. 484–495.

[20] J. Xu, D. Deng, U. Demiryurek, C. Shahabi, and M. van der Schaar, “Mining the situation: Spatiotemporal traffic prediction with big data,” *Selected Topics in Signal Processing, IEEE Journal of*, vol. 9, no. 4, pp. 702–715, 2015.

[21] J. Zhou and A. K. Tung, “Smiler: A semi-lazy time series prediction system for sensors,” in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 2015, pp. 1871–1886.

[22] J. Wang, Y. Mao, J. Li, Z. Xiong, and W.-X. Wang, “Predictability of road traffic and congestion in urban areas,” *PloS one*, vol. 10, no. 4, 2015.

[23] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[24] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1798–1828, 2013.

[25] 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.

[26] B. Pan, U. Demiryurek, and C. Shahabi, “Utilizing real-world transportation data for accurate traffic prediction,” in *2012 IEEE 12th International Conference on Data Mining (ICDM)*. IEEE, 2012, pp. 595–604.

[27] D. Deng, C. Shahabi, U. Demiryurek, L. Zhu, R. Yu, and Y. Liu, “Latent space model for road networks to predict time-varying traffic,” in *KDD 2016*, 2016.

[28] Y. Wang, Y. Zheng, and Y. Xue, “Travel time estimation of a path using sparse trajectories,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 25–34.