The 2016 IEEE International Conference on Data Mining

Barcelona, Spain. December 12-16, 2016

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

Jingyuan Wang, Qiang Gu, Junjie Wu ,Guannan Liu, Zhang Xioing

Beihang University, Beijing, China

Targets of our work

- Traffic speed prediction
  - Problem: Predicting future traffic speed of a road segment using history speed data
  - **Users: car drivers**

- Congestion source exploration
  - Problem: Discovering segments that may cause traffic congestions
  - **Users: urban planners**

Related works

- Traffic speed prediction
  - Before 2000: ARIMA and its variations
    - ARIMA (1979), Kohonen-ARIMA (1996), ARIMAX (1999), etc.
  - After 2000: ANN and SVR
    - SVR (2004), OLWSVR (2013), ANN (2001), fuzzy neural network (2006), etc.
  - After 2014: Deep Learning
    - RNN-RBM (PlosOne2015), SAE (T-ITS2014, T-ITS2015)
- Shortcomings
  - Consider traffic prediction as a regular time series prediction problem.
  - Fail to model some unique features of the traffic scenarios.

Unique features

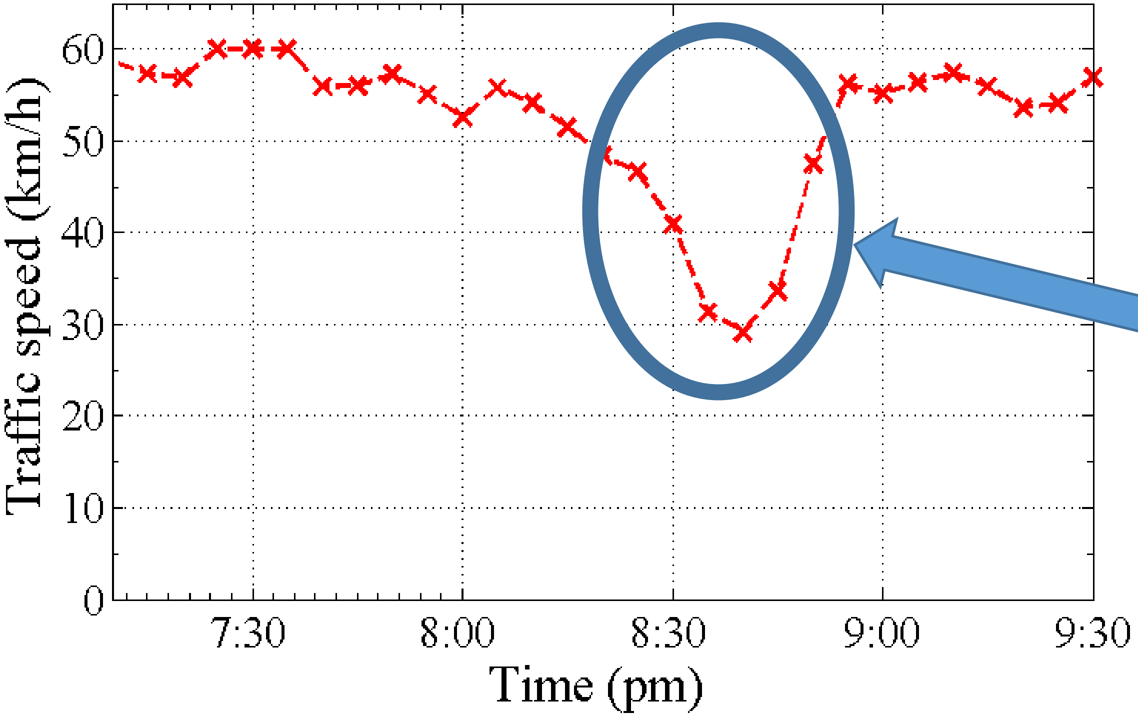
- Spatio-temporal correlations
  - Congestion may be caused by a segment in the downstream of a road.
  - Traffic congestion on a road always last a very long time.
- Spatio-temporal **locality**

No-congestion

Congestion

Unique features

- Unpredictable events

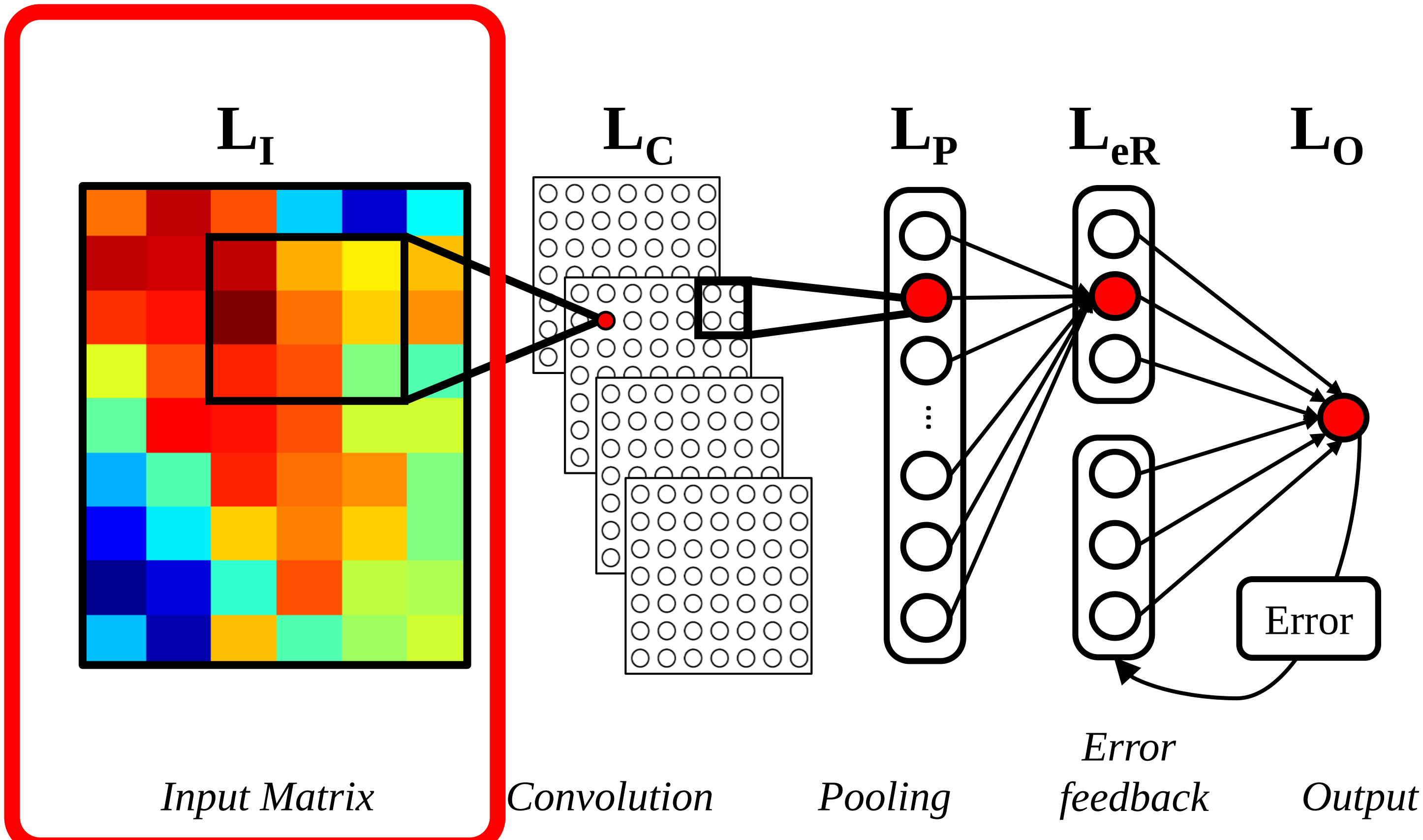


Framework

- Error Feedback Recurrent CNN (eRCNN)

1) The Spatio-Temporal Input Matrix

Function: Modeling ST relationship



Error Feedback Recurrent CNN

- The Spatio-Temporal Input Matrix

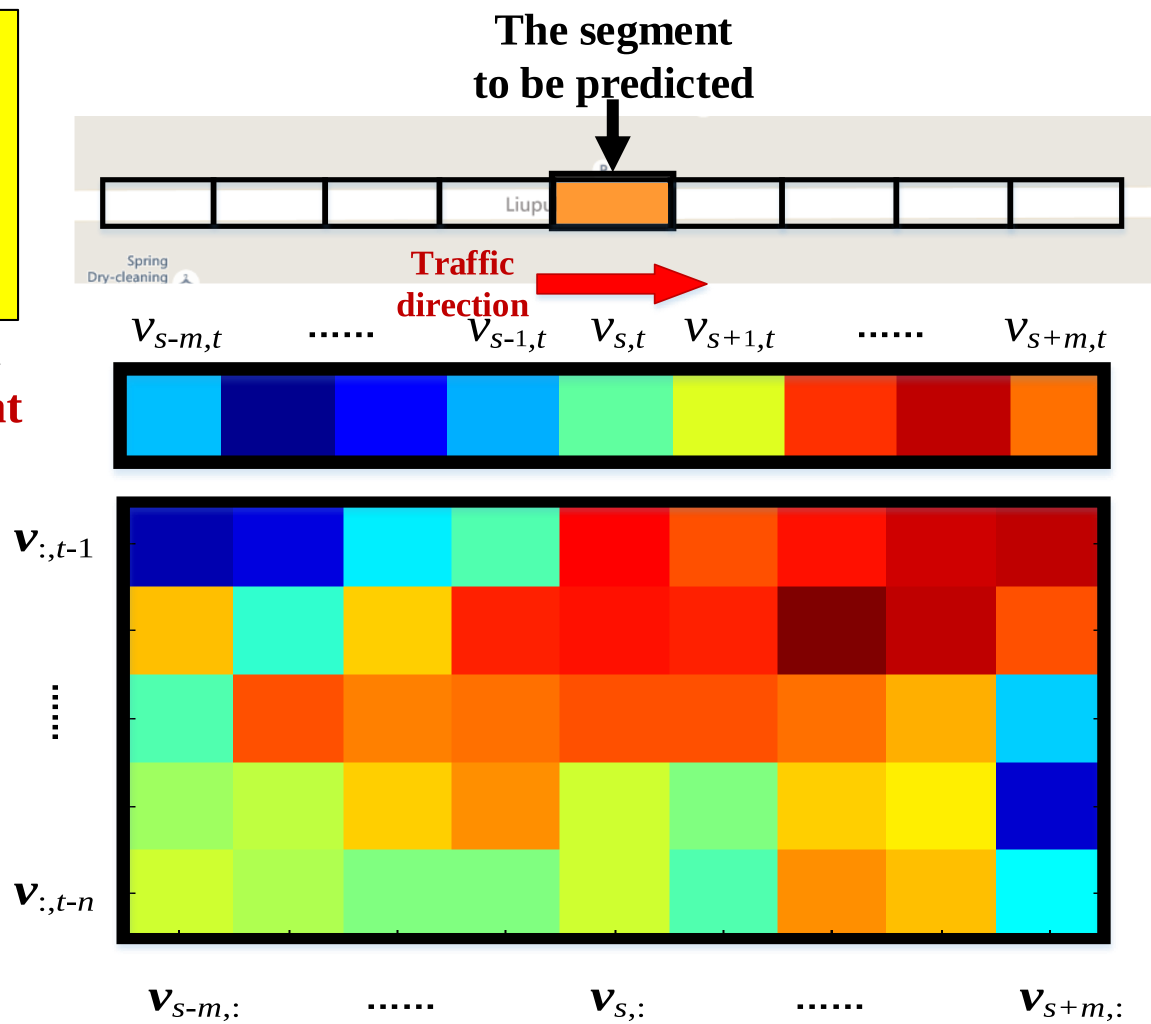
Goal:
Predicting traffic speed of a road segment **at the time *t*+1**

Speed of up/down stream segments **at the time *t***

at the time *t*-1

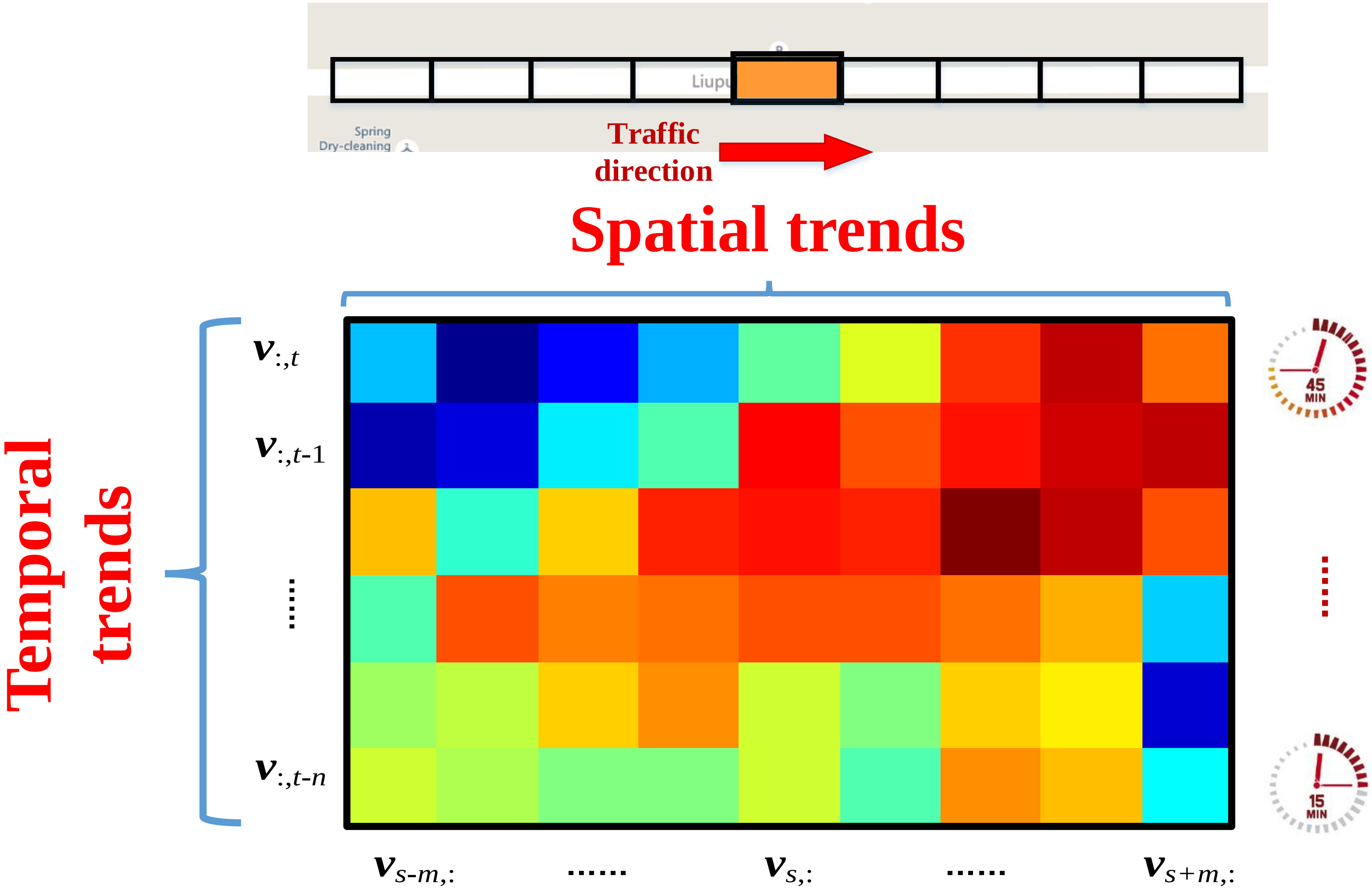
⋮

at the time *t*-*n*



Error Feedback Recurrent CNN

- The Spatio-Temporal Input Matrix

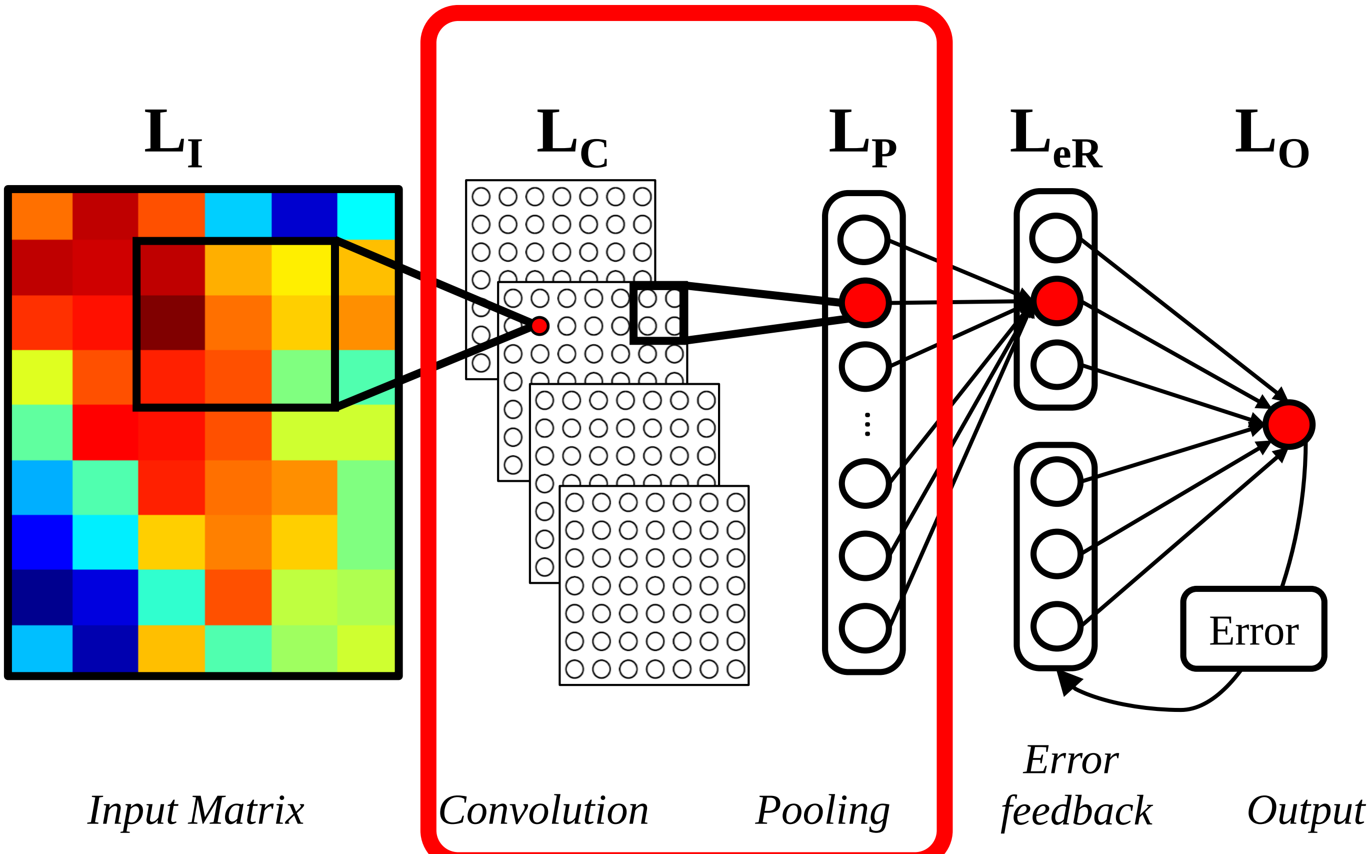


Framework

- Error Feedback Recurrent CNN

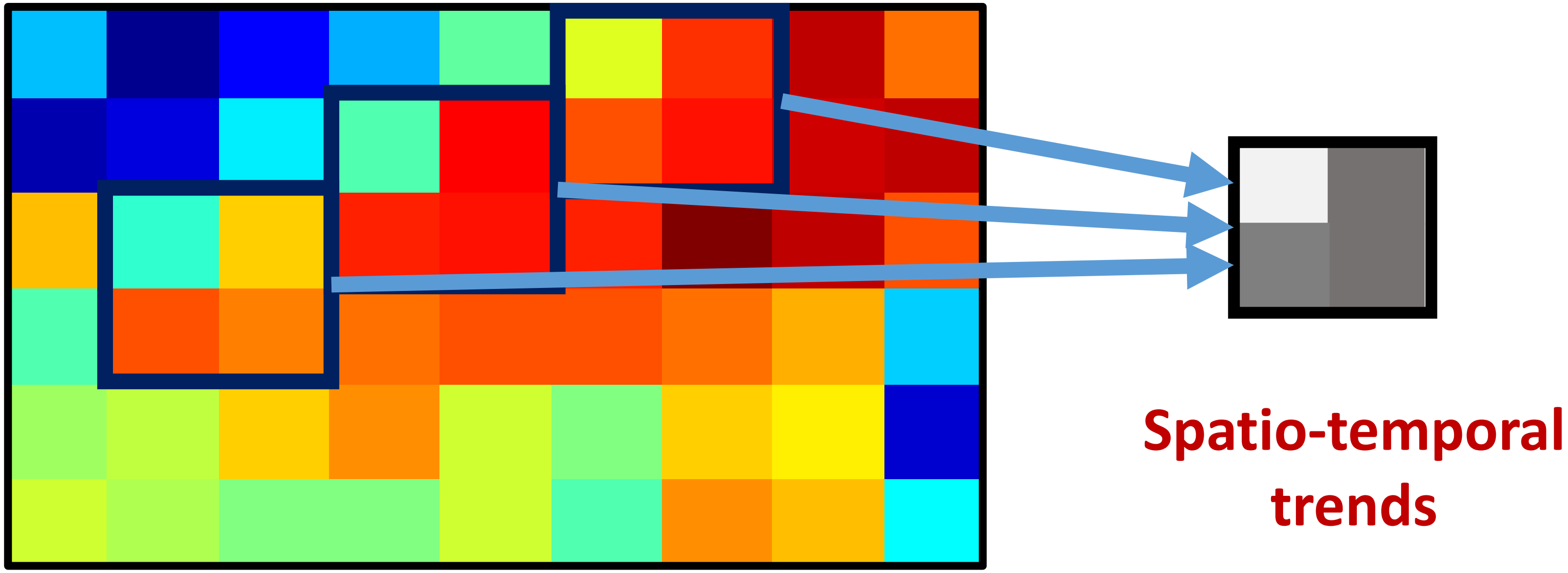
2) The CNN-based Feature Extracting

Function: Extracting ST correlation and locality Feature



Error Feedback Recurrent CNN

- The CNN-based Feature Extracting

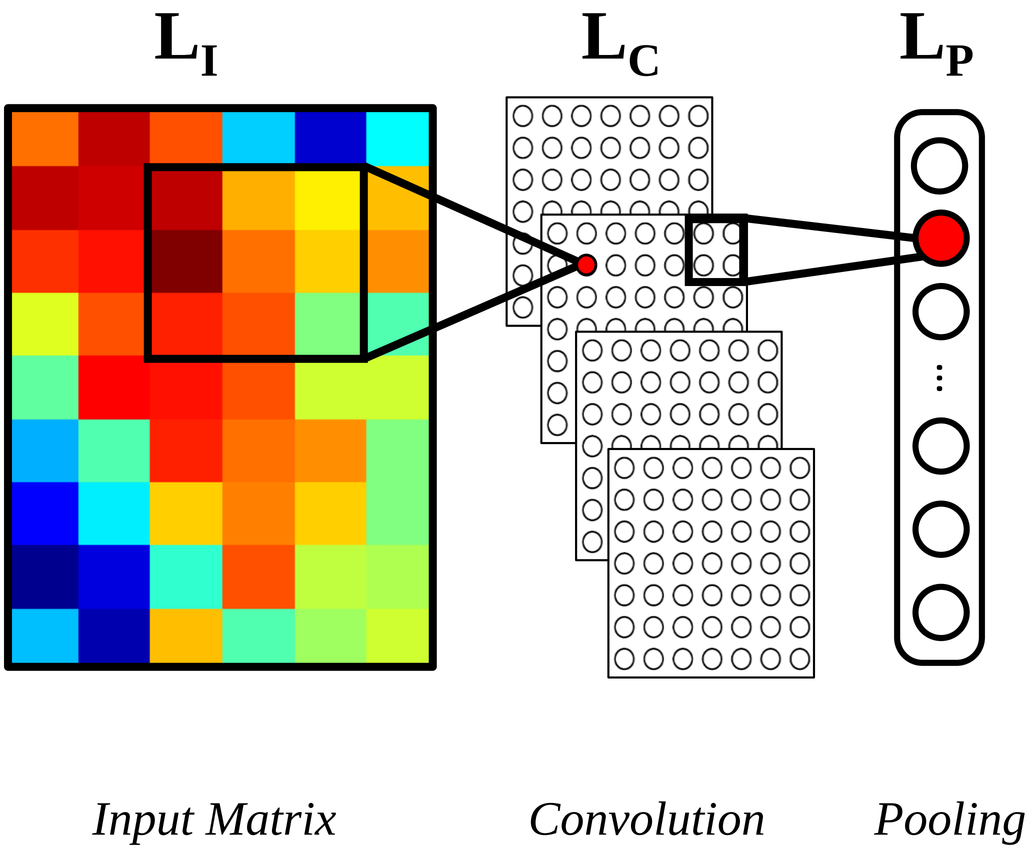


Insight: There exists a number of local patterns (spatio-temporal trends) in the spatio-temporal input matrix.

Idea: We adopt a Convolutional Neural Network based structure to extract spatio-temporal trend features from the spatio-temporal input matrix.

Error Feedback Recurrent CNN

- The CNN-based Feature Extracting



Convolution layer:

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

Using several filters to convolute the input matrix.

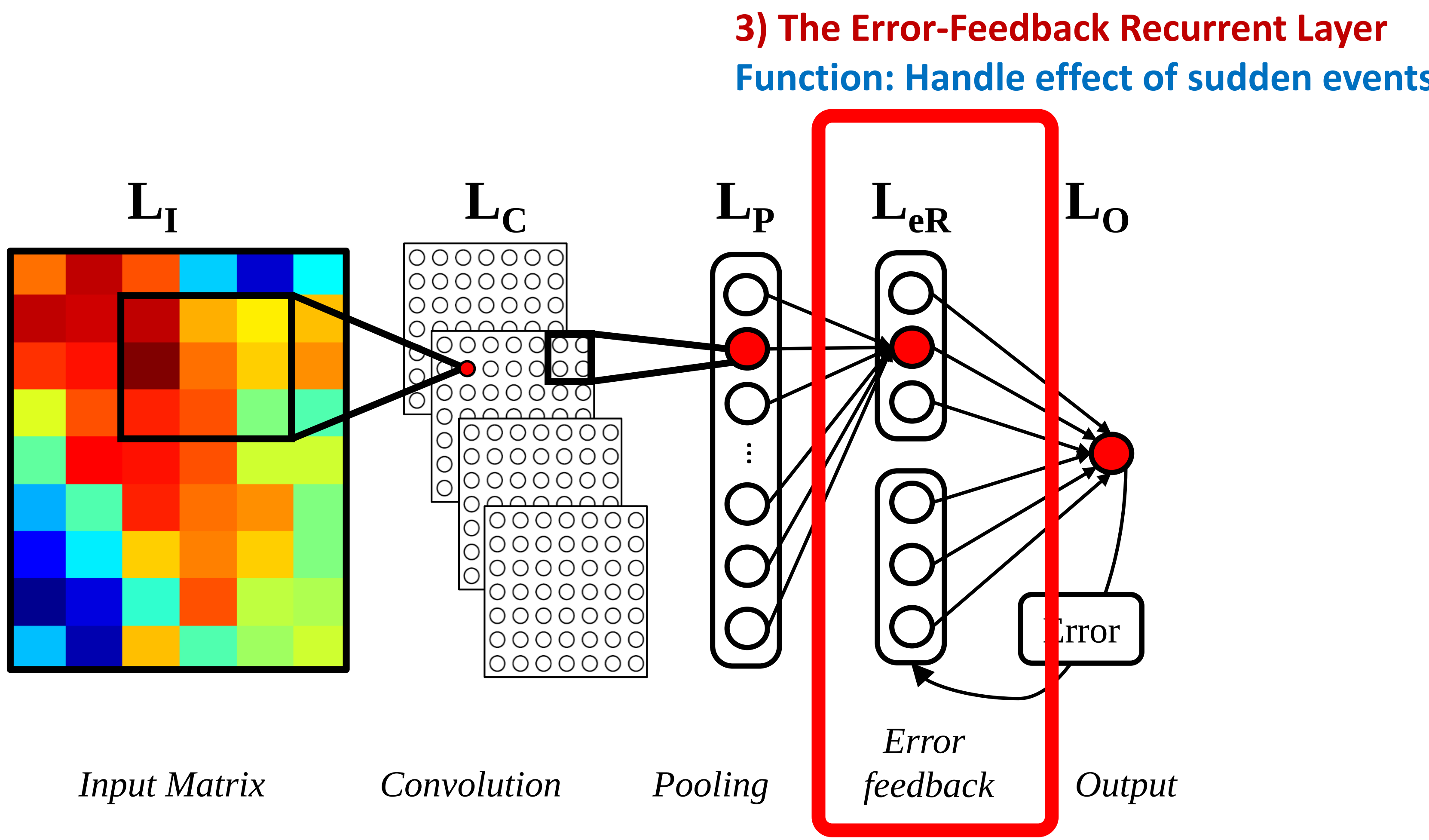
Pooling layer:

$$p_c = \frac{1}{N}\sum_{p}\sum_{q} c_k^{p,q}$$

Using average pooling to down sampling the convolution neuro matrix.

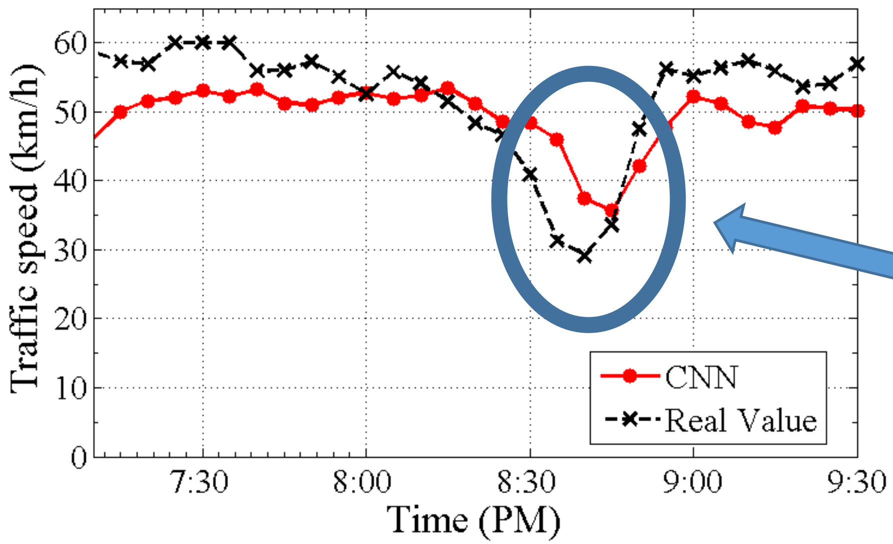
Framework

- Error Feedback Recurrent CNN

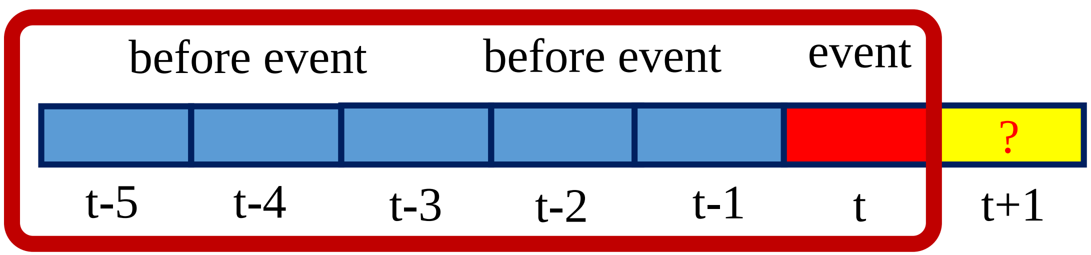


Error Feedback Recurrent CNN

- The Error-Feedback Recurrent Layer



A speed drop off caused by a unpredictable event.

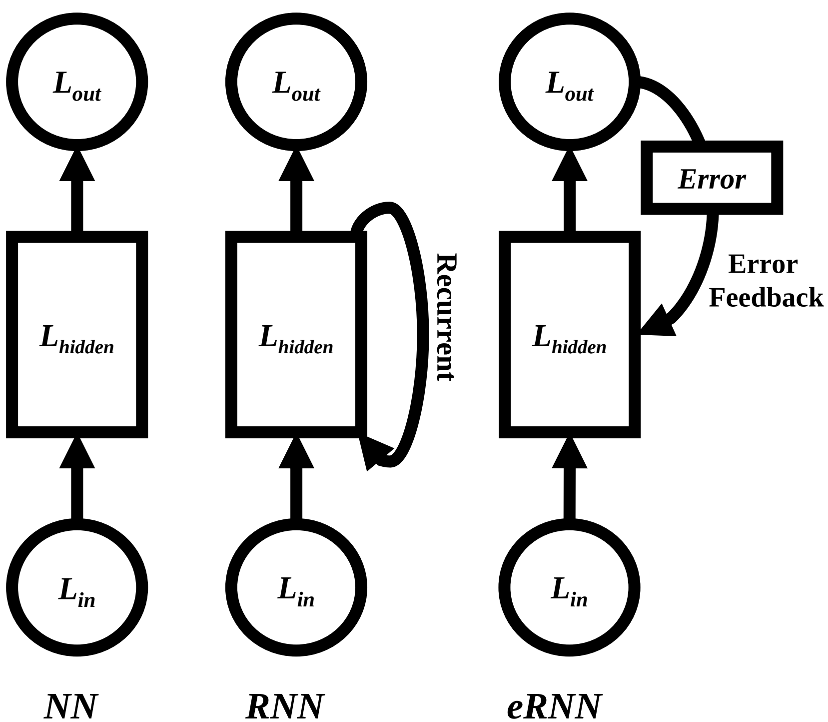


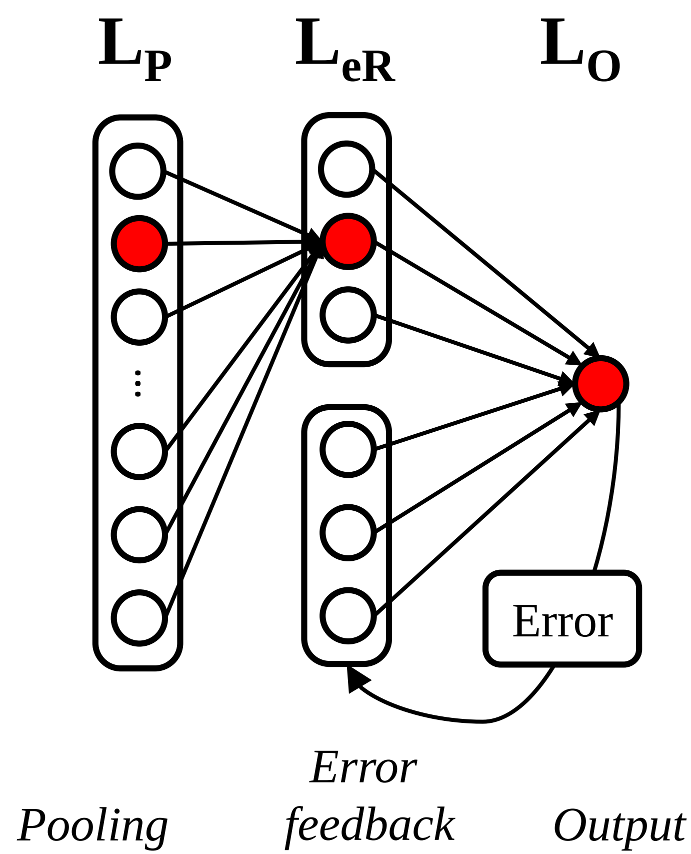
Input: before 5 > after 1

The prediction model input does not contain enough information about the event.

Error Feedback Recurrent CNN

- The Error-Feedback Recurrent Layer
  - Idea: feed the prediction error back to the network



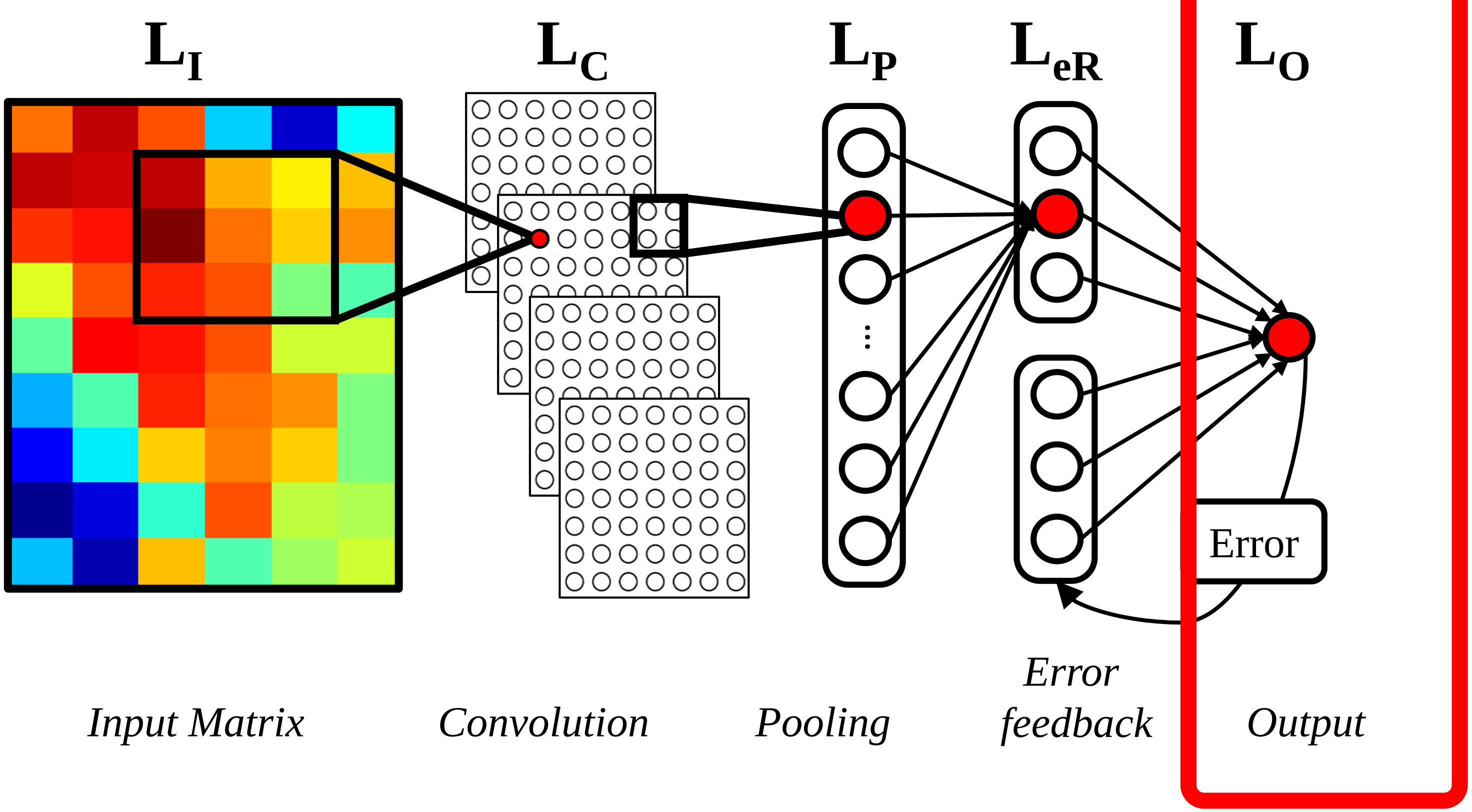


Framework

- Error Feedback Recurrent CNN

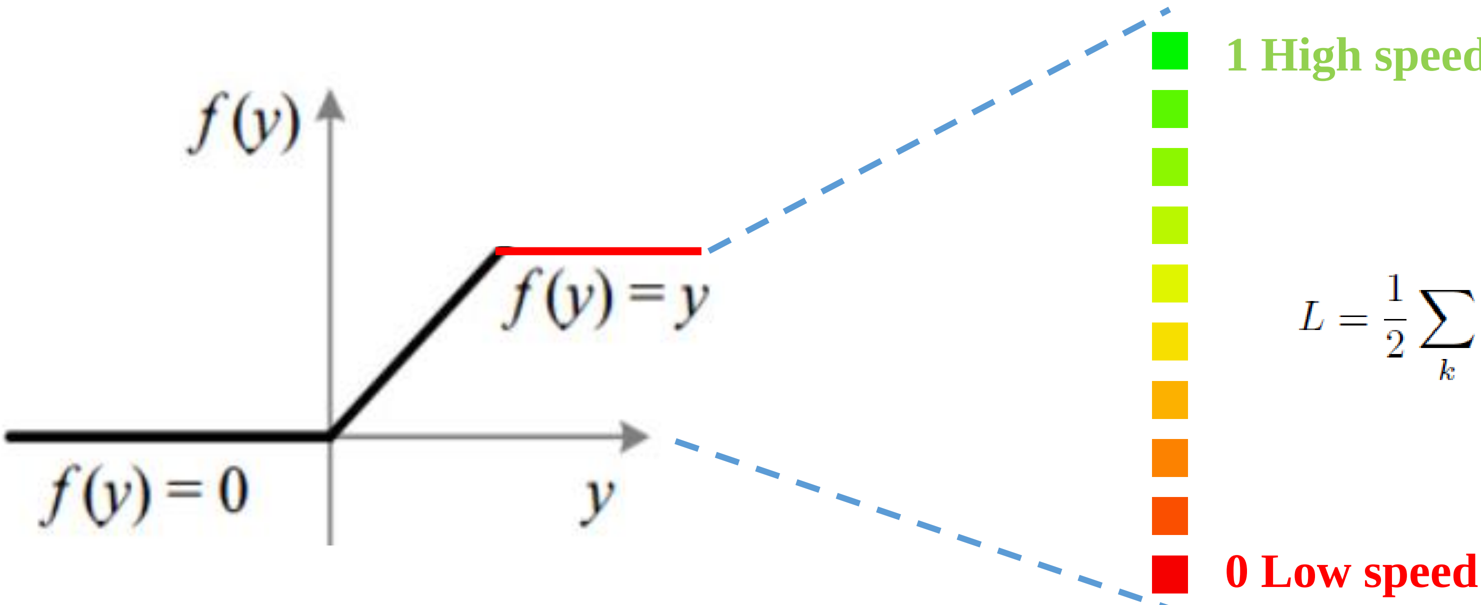
4) The Regression Output Layer

Function: speed regression



Error Feedback Recurrent CNN

- The Output Layer



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

Modified *ReLU*

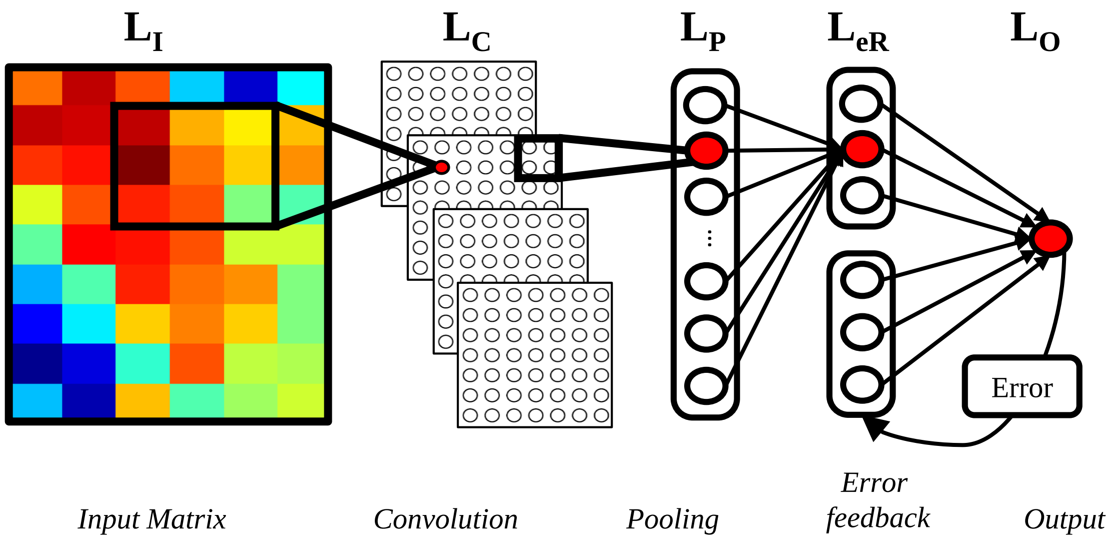
$$\sigma(x) = \begin{cases} 0 & \text{if} \quad x \leq 0 \\ x & \text{if} \quad 0 < x < 1 \\ 1 & \text{if} \quad x \geq 1 \end{cases}$$

Speed Normalization

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

Network Training

- Parameters Training: **B**ack **P**ropagation



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

$$\frac{\partial L}{\partial \mathbf{w}^{(O)}} = \frac{1}{m}\sum_m d^{(O)}(t)[\mathbf{r}^{(R)};\mathbf{r}^{(E)}],$$

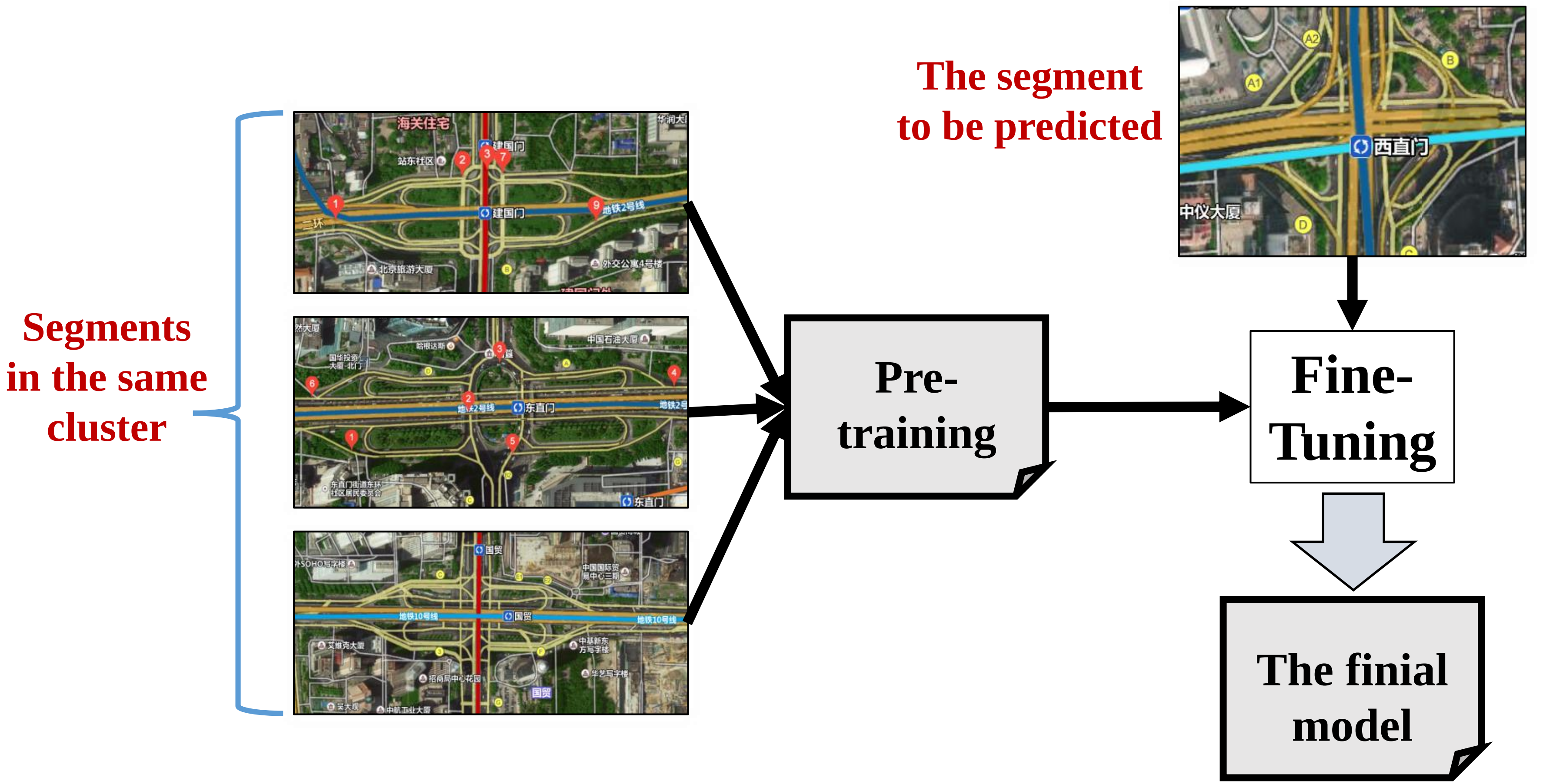
$$\frac{\partial L}{\partial b^{(O)}} = \frac{1}{m}\sum_m d^{(O)}(t),$$

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

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

Network Training

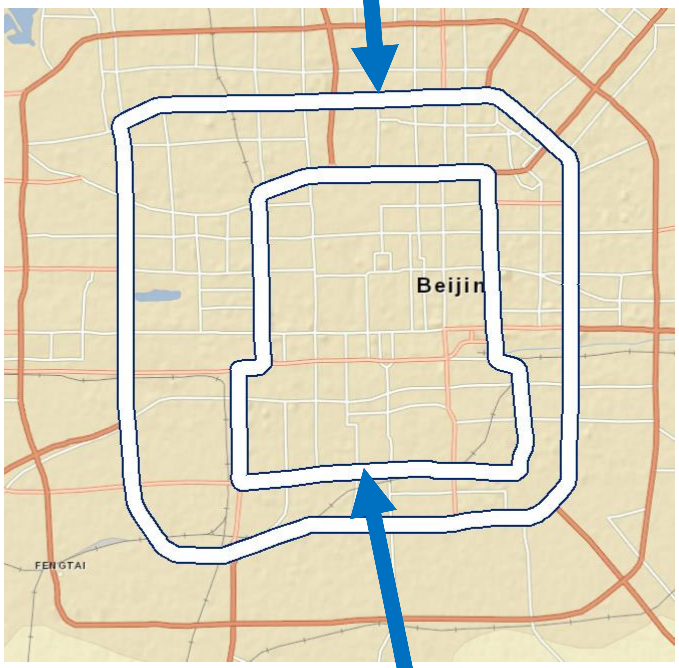
- Pre-Training and Fine-Tuning eRCNN



Experiments: Data Description

- The **2nd ring road** and the **3rd ring road**
  - About **10% of the total traffic flow** in Beijing downtown area.
  - The average length of each road segment is **400 meters**.
  - The traffic speed of a segment is updated every **5 minutes**.
- The data set was collected from the **25 weekdays** in Nov. 2013.
  - The data of the first **20 weekdays** were used as the training set.
  - The remaining **five days** is the test set.

The 3rd ring road



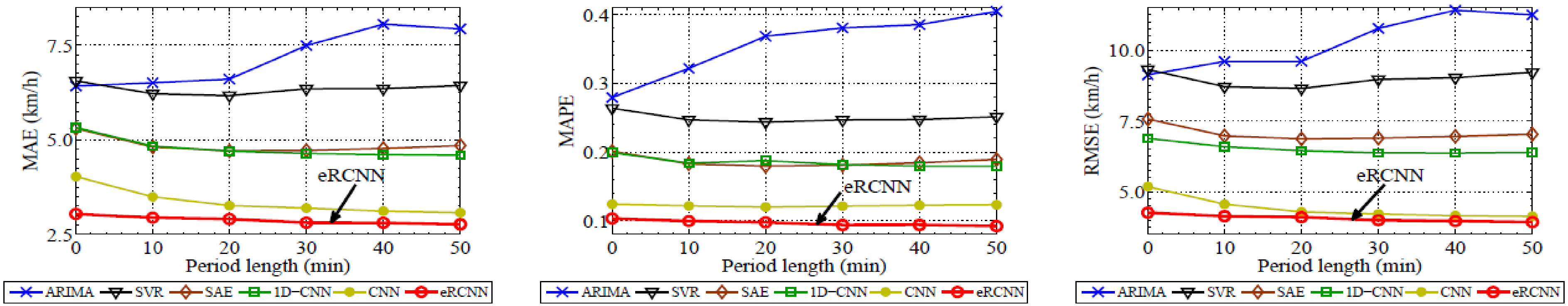
The 2nd ring road

Experiments

- Benchmarks
  - Auto Regression Integrated Moving Average **(ARIMA)** [1]
  - Support Vector Regression **(SVR)** [11]
  - Stacked Auto Encoders **(SAE)** [5]
  - 1D Convolutional Neural Network **(1D-CNN)**
    - 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.
    - benchmark to test the spatio-temporal input matrix.
  - Convolutional Neural Network **(CNN)**
    - the same as eRCNN, except the error feedback procedure is removed.
    - benchmark to test the performance of the error feedback scheme.

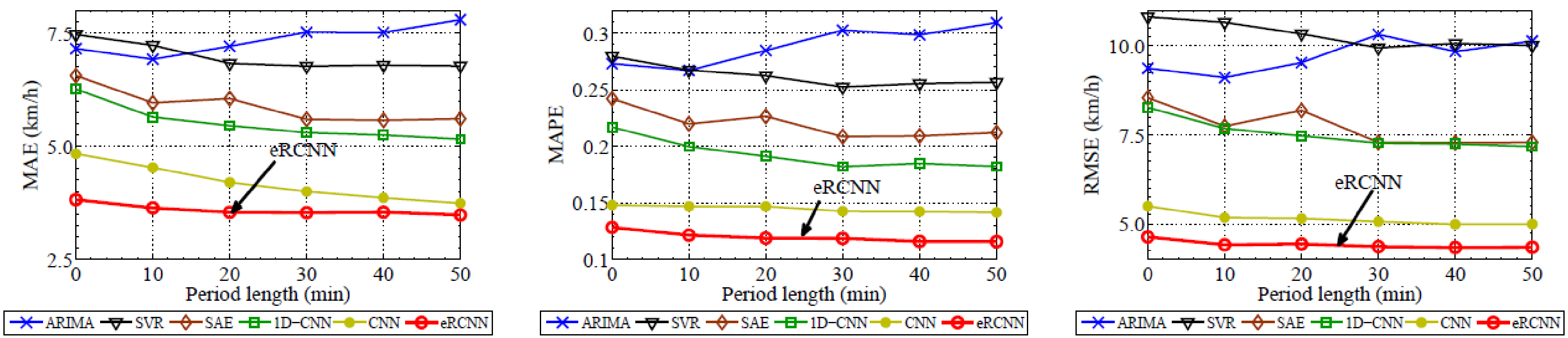
Experiments

- Overall performance: scenario I



(a) MAE Comparison (b) MAPE Comparison (c) RMSE Comparison

The 2nd ring road

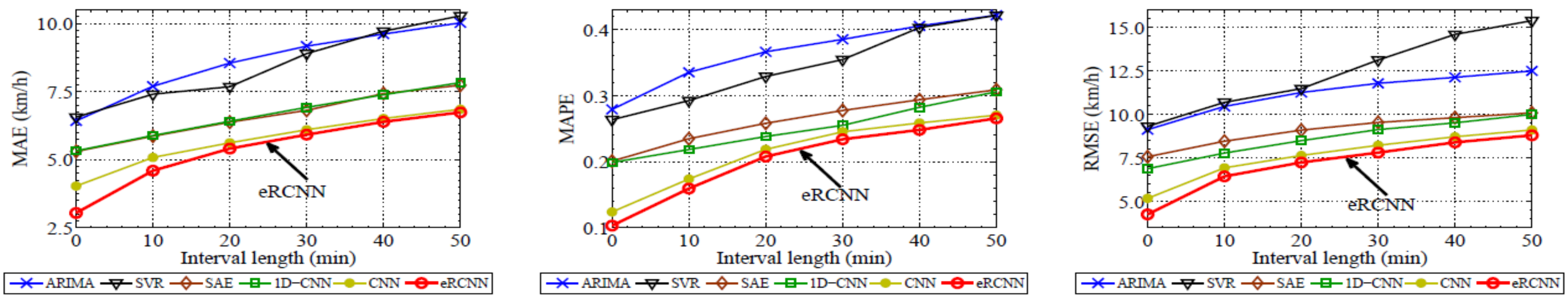


(a) MAE Comparison (b) MAPE Comparison (c) RMSE Comparison

The 3rd ring road

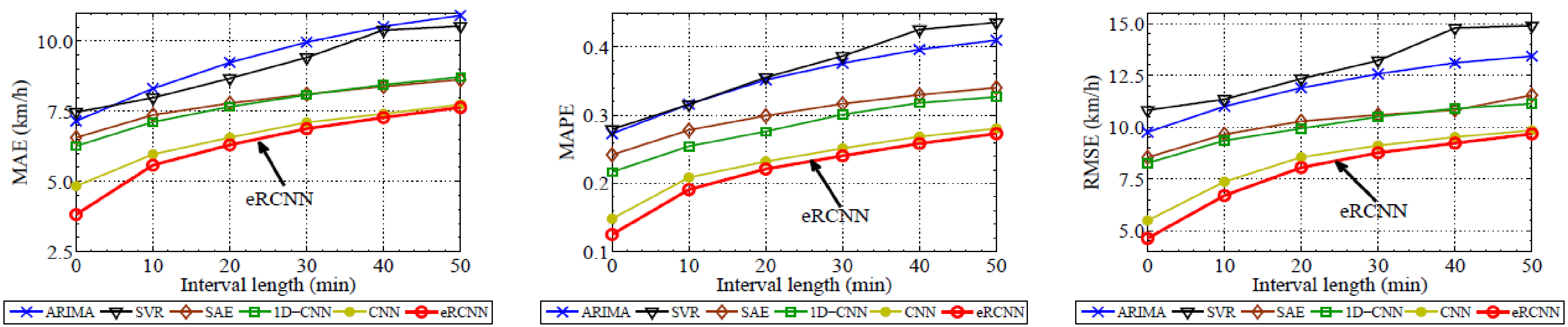
Experiments

- Overall performance: scenario II



(a) MAE Comparison (b) MAPE Comparison (c) RMSE Comparison

The 2nd ring road

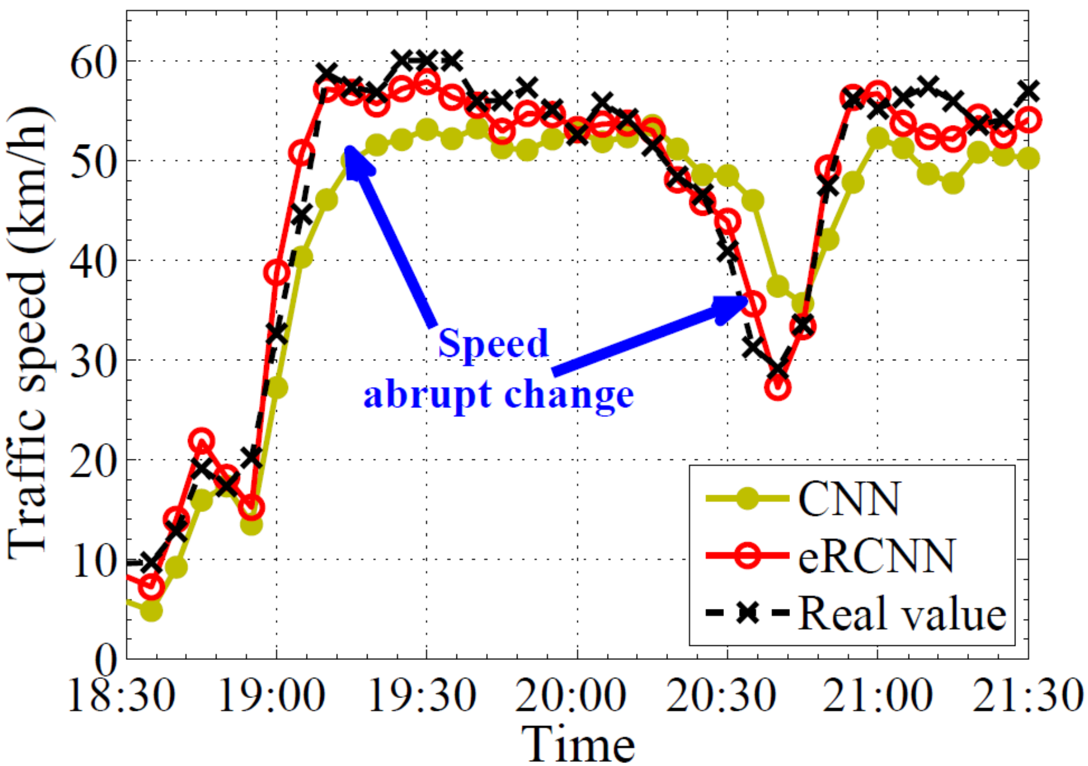


(a) MAE Comparison (b) MAPE Comparison (c) RMSE Comparison

The 3rd ring road

Experiments

- Performance with Time Variation



Prediction delay

- 19:00 to 19:30, the traffic recovers from the last traffic jam of the night peak
- around 20:20, the traffic speed decreases again due to a small accident

Performance

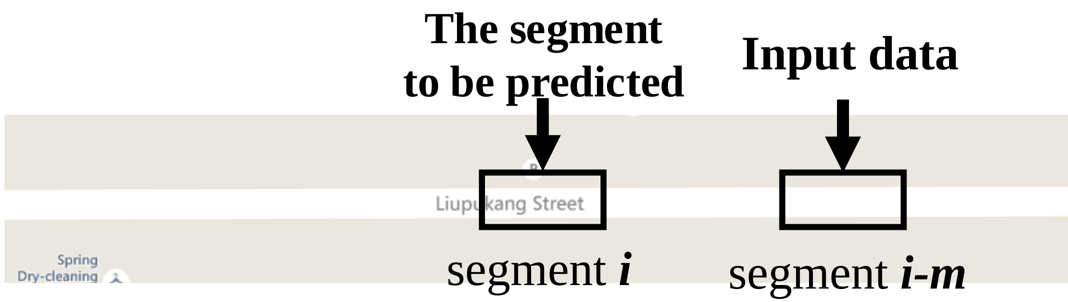
- eRCNN captures the abrupt changes in speeds
- the prediction curve exactly matches the real values
- CNN model does not follow the abrupt changes of traffic speeds

Targets of our work

- Traffic speed prediction
  - Problem: Predicting future traffic speed of a road segment using history speed data
  - Applications: navigation
  - Users: car drivers

- **Congestion source exploration**
  - **Discovering segments that cause traffic congestions**
  - **Users: urban planners**

Importance analysis for road segments



We use the traffic speed of the segment ***i-m*** to predict the speed of the segment ***i***

imply

The segment ***i-m*** has a **influence** to the segment ***i***

We define the ***influence*** of segment ***i*** to segment ***j*** as the derivative of v_j to v_i, i.e.

$$I_i(j) = \frac{\mathrm{d}f(v_i)}{\mathrm{d}v_i} = \lim_{\varepsilon \to 0} \frac{f(v_i) - f(v_i - \varepsilon)}{\varepsilon}$$

We approximately calculate the ***influence*** of the segment ***s−m*** to ***s*** at time ***t*** as

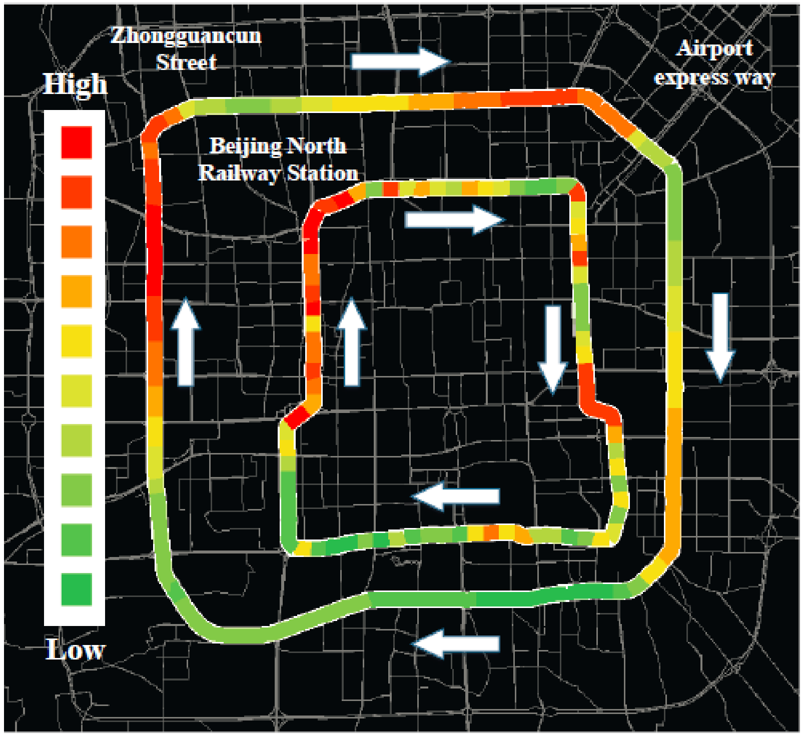
$$I_{s-m,t}(s) = \sum_{k=t}^{t-n} \left| \frac{\partial o_{s,t+1}}{\partial v_{s-m,k}} \right|$$

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

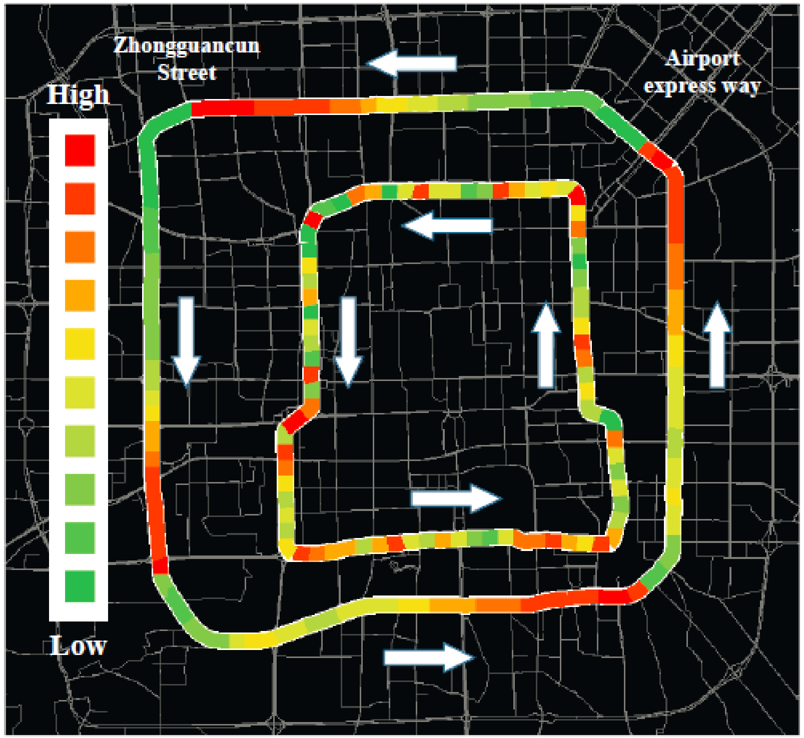
$$\mathrm{Importance}_k = \sum_t \sum_{s \neq k} \mathrm{I}_{k,t}(s).$$

Importance analysis for road segments

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



(a) The Inner Loop (clockwise)



(b) The Outer Loop (anti-clockwise)

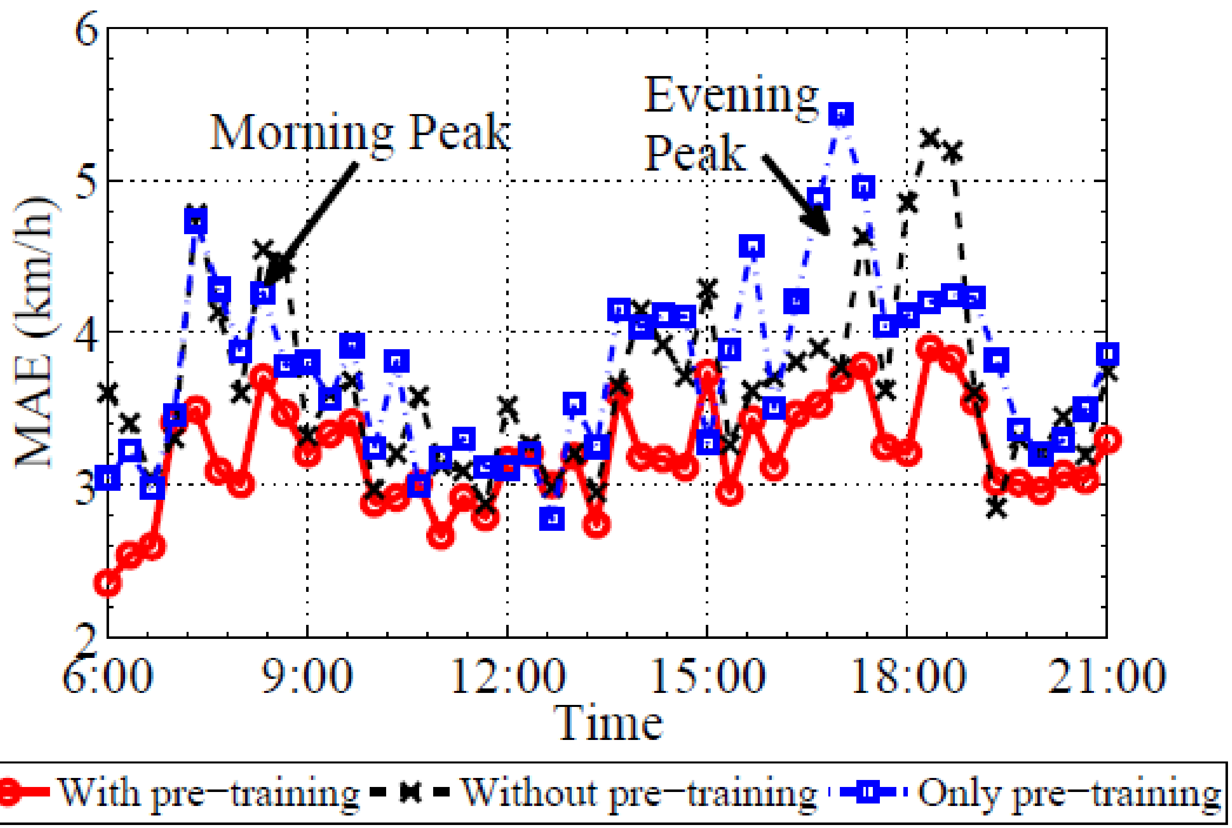
Conclusion

- In this paper, we proposed a novel deep learning method called eRCNN for traffic speed prediction of high accuracy.
- Experiments on real-world traffic speed data of the ring roads of Beijing city demonstrated the advantages of eRCNN to the excellent competitors.
- In particular, we illustrated how to explore the congestion sources from eRCNN.

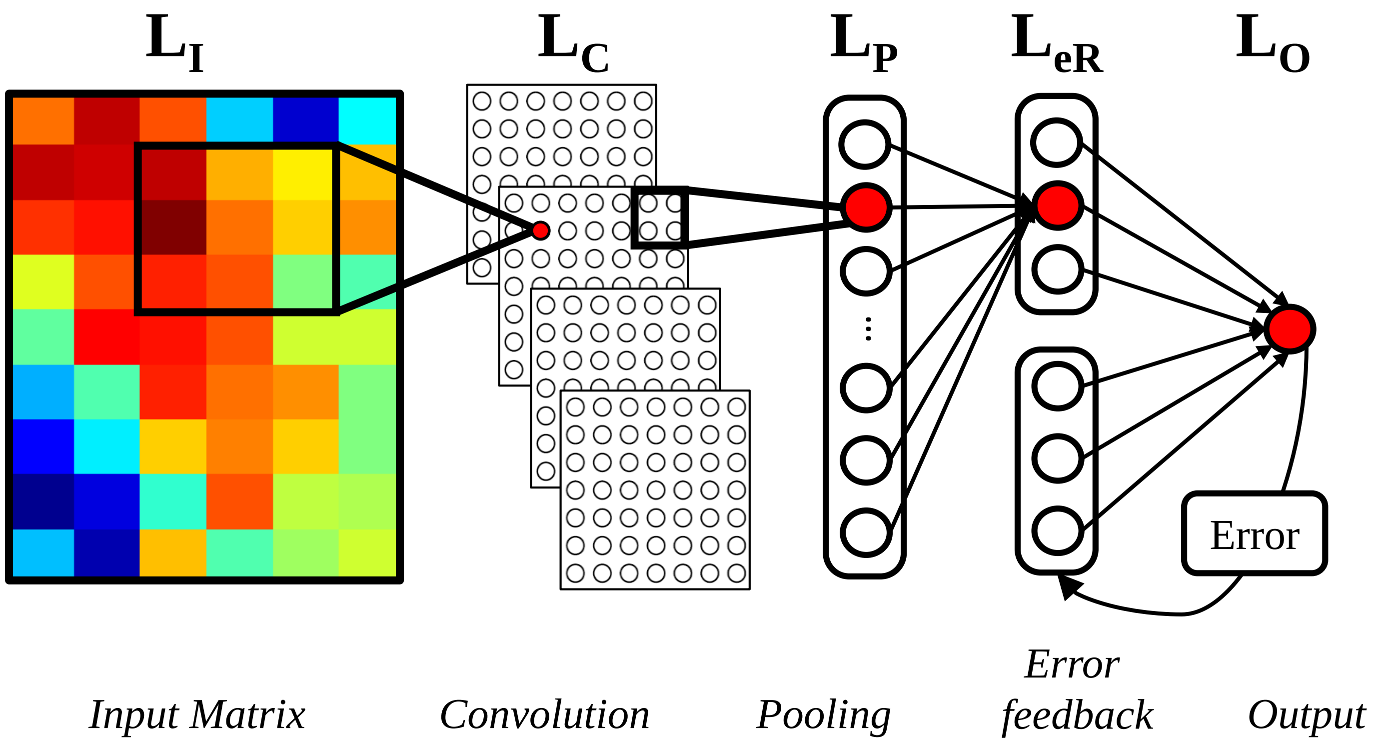
THANK YOU!

Experiments

- Performance with Pre-Training
  - eRCNN is greatly enhanced by the pre-training scheme even facing the drastic speed changes during the morning and evening peaks.



Importance analysis for road segments



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

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

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

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

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

Error Feedback Recurrent CNN

- The Error-Feedback Recurrent Layer