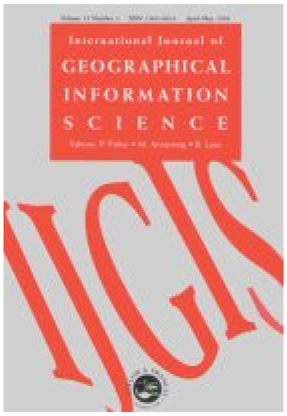


Deep Spatio-Temporal Residual Neural Network for Road-Network-Based Data Modeling

Journal:	*International Journal of Geographical Information Science*
Manuscript ID	IJGIS-2018-0335.R3
Manuscript Type:	Research Article
Keywords:	spatio-temporal modeling, road network, deep learning, residual neural network

Deep Spatio-Temporal Residual Neural Network for Road-Network-Based Data Modeling

Recently, researchers have introduced deep learning methods such as convolutional neural networks (CNN) to model spatio-temporal data and achieved better results than those with conventional methods. However, these CNN-based models employ a grid map to represent spatial data, which is unsuitable for road-network-based data. To address this problem, we propose a deep spatio-temporal residual neural network for road-network-based data modeling (DSTR-RNet). The proposed model constructs locally-connected neural network layers (LCNR) to model road network topology and integrates residual learning to model the spatio-temporal dependency. We test the DSTR-RNet by predicting the traffic flow of Didi cab service, in an 8-km^2 region with 2,616 road segments in Chengdu, China. The results demonstrate that the DSTR-RNet maintains the spatial precision and topology of the road network as well as improves the prediction accuracy. We discuss the prediction errors and compare the prediction results to those of grid-based CNN models. We also explore the sensitivity of the model to its parameters; this will aid the application of this model to network-based data modeling.

Keywords: spatio-temporal modeling; road network; deep learning; residual neural network

1 Introduction

Spatial-temporal modeling has always been of interest to researchers in geographical information science (GIS) with wide applications in modeling and predicting spatio-temporal processes such as biological phenomena (Stockwell 1999), environment information (Cheng and Wang 2008, Cheng and Wang 2009, Xingjian et al. 2015), and distribution of urban elements (Jiang 2009, Chen *et al*. 2018). The rapid proliferation of mobile sensors and Internet technologies continuously generates an exceptionally large amount of spatio-temporal data, which offers unprecedented opportunities for modeling and predicting human activities (Zhu and Guo 2014)(Huang *et al*. 2015, Li *et al*. 2016, Shaw *et al*. 2016, Hoang *et al*. 2016). Because most human movements occur along a road network, the road-network-based data (e.g., traffic flow, crime data, and passenger volume) account for a large portion of spatio-temporal data (Jiang and Liu 2009, Cheng *et al*. 2011, Rosser *et al*. 2016). Modeling and predicting road-network-based data can provide essential references for urban managers to address a variety of problems (Ma *et al*. 2017, Ke *et al*. 2017). Therefore, the development of an effective framework for modeling road-network data is important, particularly to support accurate prediction at the level of road segments.

For the spatio-temporal data, the capability to model the spatial and temporal dependency seamlessly and simultaneously is important to achieve predictions of high accuracy (Cheng and Wang 2009, Zhang *et al*. 2017a). The foundation for most available spatio-temporal data prediction models are statistical and machine learning (ML) methods. An integration of spatial and temporal variables into available statistical models can account for spatio-temporal dependency, such as the space-time autoregressive integrated moving average (ARIMA) (Wang *et al*. 2010, Cheng *et al*. 2014), the space-time support vector regression (Wang *et al*. 2007), kernel-based methods (Haworth *et al*. 2014, Rosser *et al*. 2016), and the space-time artificial neural network models (Cheng and Wang 2009, Wang *et al*. 2016). However, these conventional models are incapable of processing raw spatio-temporal data. When constructing a machine-learning (ML)-based model, feature extractors require precise engineering and substantial domain knowledge to transform raw data into proper internal representations for spatio-temporal dependency detection. This procedure is called feature engineering (LeCun *et al*. 2015). With regard to big data, the feature engineering procedure of spatio-temporal data is particularly challenging.

Deep learning (DL) addresses this challenge (Hinton and Salakhutdinov 2006). A typical DL model can accept input data in a raw format and automatically discover the required features level-by-level. Called “end-to-end” learning, it significantly simplifies feature engineering (LeCun *et al*. 2015). The deep CNN was specifically designed to capture spatial dependency; it has achieved significant successes in image recognition (Krizhevsky *et al*. 2012). Previous studies have attempted to introduce CNN to model spatio-temporal dependency. Zhang *et al*. (2016) used a grid map to represent the spatio-temporal flow volume of a city and inputted several typical historical maps into a multi-layer CNN structure to model the spatial and temporal dependencies simultaneously; their objective was to achieve “end-to-end” prediction of citywide spatio-temporal flow volume. Furthermore, Zhang *et al*. applied a deep residual network (ResNet) (He *et al*. 2016a) to increase the depth of DeepST to model the dependency from more distant regions; they called it ST-ResNet (Zhang *et al*. 2017b). In predicting the citywide taxi flow volume in

Beijing, China, ST-ResNet exhibited an accuracy higher that of DeepST by 7.09%. Studies then widely adopted the CNN-based methods with the grid representation of spatial data (point, road network, and polygon) (Yu *et al*. 2017, Ke *et al*. 2017). Ma *et al*. (2017) proposed another method to transform the road-network-based data into a two-dimensional image with the horizontal axis representing time tags and the vertical axis representing road segments. This image was then inputted into a CNN framework to model the spatio-temporal dependency using a convolution operator.

The CNN-based methods limit the organization of the data to a grid or image format; this is suitable for modeling spatio-temporal data such as those of crowd flow and urban population. However, the CNN-based methods exhibit two problems with the road-network data. Grids or images cannot accurately represent the spatial structure of road networks. For example, Zhang *et al*. (2017b) adopted a grid cell of 1-km square; it may cover dozens of road segments in an urban district, diminishing road segments on the system. Ke *et al*. (2017) used a 10-m grid cell, which could not accurately represent the junctions and overpasses. Ma *et al*.'s (2017) image represented each road segment separately; however, it omitted the network topology. Secondly, the convolution operator of the CNN was not optimal for modeling the spatial patterns of the road because the authors did not consider the road network topology.

To address these problems, we propose a deep spatio-temporal residual neural network for road-network-based data modeling (DSTR-RNet). We design a locally-connected neural network layer to model road network topology (LCNR) in order to capture the local spatial dependency of each road segment. Moreover, we employ residual learning (He *et al*. 2016a) to form a deep residual LCNR (ResLCNR) unit to model the spatial dependency from near to distant neighbors. Finally, three sub-models based on the ResLCNR unit capture the spatial and temporal dependency in an integrated manner from different temporal patterns, forming the final DSTR-RNet. Compared with previous models, the proposed model exhibits the following two advantages. First, ResLCNR presents a deep-learning-based spatio-temporal modeling method at the road-network level rather than the grid level; this maintains the spatial precision of road-network-based data. Second, the model accounts for the inherent topology of the road network to improve feature extraction for a more accurate prediction.

The rest of this paper presents the following. Section 2 defines the problem of road-network-based data modeling. Section 3 explains the proposed LCNR, residual LCNR unit, and framework of the DSTR-RNet model. Section 4 introduces the case study that models and predicts the traffic flow of the Chengdu, China road network. Section 5 discusses the spatial distributions of the prediction errors. The section also describes a comparison of grid-based CNN models and the proposed DSTR-RNet. Finally, Section 6 summarizes the conclusions and directions for future work.

2 Problem Definition

The research objective is to model the spatial and temporal dependency seamlessly for data on road networks. For each segment of the road network, the spatial dependency originates from both near and distant neighbors. For example, the traffic flow on each road at a crossing affects one another; simultaneously, numerous individuals drive to offices over various distances, generating distant/spatial dependency. The temporal dependency has three parts: recent pattern, daily pattern, and weekly pattern (Ma *et al.* 2014, Zhang *et al*. 2017b, Ke *et al*. 2017). The recent pattern refers to the dependency from several nearest historical time intervals prior to the target time, e.g., the relation between the traffic flow at 08:00 am and that at 07:30 am. The daily pattern refers to the human activities that repeat every 24 h. Similarly, the weekly pattern refers to the activities repeated every week; e.g., whereas weekdays and weekends within a week exhibit different traffic flows, weekdays from different weeks exhibit similar scenarios.

Let x_t represent the road network data corresponding to the t^{th} time interval. Let X_R, X_D, and X_W represent the historical data series of recent, daily, and weekly patterns, respectively. Assuming that the numbers of time intervals in X_R, X_D and X_W are r, d, and w, respectively, we need to predict the target time interval t; moreover, the number of time intervals in one day is m. Then, we can define X_R, $X_{D,}$ and X_W as in Equation (1). Let f denote the transformation of the modeling method and W denote all the parameters to be learned. We can define the problem of predicting scenarios for all the segments as in Equation (2). The purpose of this study is to construct a transformation f that can accurately model the spatio-temporal dependency (W) from historical observations and make accurate predictions with new inputs.

$$X_R = (x_{t-r},...,x_{t-2}\, , x_{t-1})$$
$$X_D = (x_{t-d\cdot m},...,x_{t-2\cdot m},\, x_{t-1\cdot m}) \tag{1}$$

$$X_W = (x_{t-w \cdot 7 \cdot m},\ldots,x_{t-2 \cdot 7 \cdot m}, x_{t-1 \cdot 7 \cdot m})$$

$$x_t = f(X_R,X_D,X_W,W) \qquad (2)$$

3 Method

3.1 Locally-connected neural network for modelling road network topology

The rapid development of DL has popularized neural networks substantially in numerous domains (Schmidhuber and Jurgen 2015, LeCun *et al*. 2015). To represent the road network structure, we transform each road segment of the road network into a node of an artificial neural network (ANN) weight layer; this is a conventional approach that most ANN-based spatio-temporal models adopt (Haworth *et al*. 2014, Wang *et al*. 2016). The number of nodes in a weight layer is equal to the number of road segments in the road network.

Traditionally, the ANN layer is fully-connected; each node of the current layer connects to all nodes of the previous layer. Figure 1 shows an example of a road network with fifteen segments. L_i and L_{i+1} are two neural network layers. For example, node 10 connects to all the nodes of the previous layer (Figure 1b). This conventional structure exhibits two limitations. Firstly, it is challenging to capture the local spatial dependency because each road segment receives information from all the other segments irrespective of being spatially adjacent or not. Secondly, the computational complexity is exponential (=O(N^2), where N is the number of all road segments). For a region with thousands of roads, the number of trainable parameters on a weight layer can be millions or even billions, making the training procedure exceptionally challenging.

Figure 1. Comparison of fully-connected layer and LCNR layer. (a) A road network example with 15 segments. (b) Node 10 connects to all the nodes. (c) Node 10 locally connects to its first-order neighbors.

To overcome these limitations, we integrate the topological adjacency of the road network into selected connections; to achieve this, we propose a locally-connected neural network for modeling road network topology (LCNR). The main concept is similar to the convolution operation; each road interacts only with its local neighbors, rather than all the road segments. Therefore, in an LCNR layer, each neural node sparsely connects to the nodes that are locally adjacent to it in the road network of the previous layer. First-order neighbors (including the element) are used to represent the locality of an LCNR layer. Given a road network, when a node directly links two segments, these segments are termed first-order neighbors (Cheng *et al*. 2011). Figure 1(c) shows that with first-order locality, node 10 connects to only four nodes: 6, 7, 10, and 11. Because the *N*-order neighbors of a segment are the first-order neighbors of its *(N - 1)*-order adjacencies, we can model any high-order spatial dependency by stacking multiple LCNR layers. In summary, Equation (3) defines the transformation that each LCNR neural node performs:

$$y_j = f(\sum_{i=1}^{k} w_{ij} x_i + b_j) \tag{3}$$

where y_j is the output of the j^{th} neural node of the LCNR layer, x_i are the nodes that represent the first-order neighbors of the j^{th} segment, $w_{i,j}$ is the weight of each connection, k is the number of first-order neighbors of the j^{th} segment, b_j is the bias, and f is the non-linear activation rectified linear unit (ReLU).

We can obtain the transformation of an LCNR layer by extending the variables to matrices (Equation (5)). X, Y, B, and W are the matrix representations corresponding to the variables in Equation (3). W_1 is the first-order adjacency matrix of the road network with an $N \times N$ binary (zero or one) matrix; here, non-zero elements signify spatial adjacency (N is the number of road segments of the road network; Figure (2b) shows an example). g initializes W; this generates non-zero initial weights on the non-zero elements of W_1 (Equation (4)) and thus maintains the network topology in W. l_s in Equation (5) refers to a sparse operator, which directly performs multiplication on the non-zero elements of $W_1 \circ W$ and its associated elements of X ("∘" denotes the Hadamard product). Let K denote the number of non-zero elements in W_1, such that the computational complexity of the forward-propagation of an LCNR layer is O(NK). Generally, the number of first-order neighbors of a road network is substantially less than the number of connected road segments. Therefore, O(NK) << O(N^2).

$$W = g(W_1) \tag{4}$$

$$Y = f\left(l_s\left((W_1 \circ W)\, X \right) + B \right) \tag{5}$$

During the back-propagation training procedure, we first calculate the partial derivative of the loss with respect to weight by Equation (6):

$$\frac{\partial E}{\partial W} = W_1 \circ \frac{\partial f}{\partial l_s} \frac{\partial l_s}{\partial W} X \tag{6}$$

where E refers to the loss. As W_1 is a constant binary matrix, the selectivity can be accumulated to $\frac{\partial E}{\partial W}$. The selectivity is then maintained in the updated weight matrix (Equation (7)):

$$W' = W - \eta \frac{\partial E}{\partial W} \tag{7}$$

where W' is the updated weight matrix and η is the learning rate. Figure 2(c) shows the selective connections of an entire LCNR layer, which is feasible to detect the spatial dependency between each segment and its first-order adjacent segments. Compared with the fully-connected layer, the effective connections of the LCNR layer reduce from 225 to 53, simplifying the training complexity significantly.

Figure 2. LCNR layer constructed by first-order spatial adjacency matrix. (a) Road network. (b) First-order spatial adjacency matrix W_1. (c) There are 53 local connections in an LCNR layer.

3.2 Deep residual LCNR for modelling distant spatial dependency

As stated in Section 2, for a road network, the spatial dependency may be present in any two segments, from near to distant. To model all these dependencies, we stack multiple LCNR layers to form a deep LCNR model. As we can derive any high-order neighbors from the first-order neighbors, a deep LCNR model with N weight layers can detect the dependency between each road segment and its N-order adjacent road segments. Figure 3 shows a deep LCNR model with three weight layers. For segment number 10, layer-by-layer transformation can capture the spatial dependency between the road segment and its three-order neighbors (segments 1, 2, 3, ……, 15.).

Figure 3. Deep LCNR model with three weight layers.

However, training a depth structure with numerous weight layers is a significant challenge for a neural network model. As the network depth increases, the vanishing or explosion of the gradient evokes higher training error, degrading the model's learning capability (LeCun *et al*. 2015). However, for the spatio-temporal model on road network data, there are two reasons to construct a deeper LCNR model. First, a deeper LCNR structure can cover a larger spatially receptive field of the road network to model the dependency from distant neighbors. Second, a deeper model can learn more inherent non-linear and non-stationary features than a shallow one. Furthermore, the LCNR layers can integrate residual learning to form a deep residual LCNR model.

Residual learning is a novel structure (He *et al*. 2016a) incorporating successful extensions to the traditional CNN model to form a super deep structure of hundreds of layers; it has established capabilities to handle numerous challenging recognition tasks. Equation (8) defines a typical residual unit with identity mapping:

$$X^{l+1} = F_{res}(X^{l}) + X^{l} \tag{8}$$

where X^l is the input and X^{l+1} is the output of the l^{th} residual unit. F_{res} is a residual function such as a stack of two 3 × 3 convolution layers. The core principle of residual learning is to learn the additive residual function F_{res} concerning X^l (He *et al*. 2016b, Zhang *et al*. 2017a). We employ residual learning to construct a residual LCNR unit (ResLCNR unit). A ResLCNR unit incorporates two stacking LCNR layers with a shortcut connecting the input X^L and output X^{L+1} (Figure 4). The ReLU is the pre-activation function (He *et al*. 2016b). For each road segment, stacking *N* ResLCNR units form a spatially receptive field covering *2N*-order neighbors; it can detect the dependency from its first-order neighbors to *2N*-order neighbors.

Figure 4. A ResLCNR unit.

3.3 DSTR-RNet for network-based spatio-temporal dependency

To integrally model the spatial and temporal dependency, we propose a deep spatio-temporal residual neural network for road-network-based data modeling (DSTR-RNet) based on the ResLCNR unit (Figure 5). We develope three sub-models to model the spatio-temporal features from the recent pattern, daily pattern, and weekly pattern, separately. We then merge these features into a final feature map; a *tanh* function activates the map to predict values. The three sub-models shared an identical structure: 1) an LCNR layer that receives the historical road network data series and outputs a feature map, with the number of elements equal to the number of road network segments and 2) a deep residual LCNR structure with *N* concealed ResLCNR units that models the spatio-temporal dependency on the feature map. The integration of spatial and temporal features on the feature map support the modeling of correlations in space and time simultaneously. We merge three feature maps (we denote them as $STFM_w$, $STFM_d$, and $STFM_r$) by a parameter-based method (Zhang *et al*. 2017a); Equation (9) defines the method:

$$STFM = STFM_w \circ W_w + STFM_d \circ W_d + STFM_r \circ W_r \qquad (9)$$

$$x_t = tanh\,(STFM) \tag{10}$$

where W_w, $W_{d,}$ and W_r are three parameter vectors with shapes identical to those of the three feature maps. *STFM* is the final spatio-temporal feature map. Then, a *tanh* function activates the *STFM* to form the prediction values, x_t (Equation (10)).

Figure 5. Framework of DSTR-RNet.

DSTR-RNet calculates the loss from the ground truths and prediction values. Here, the mean-square error (MSE) represents the loss function. Equation (11) defines it; here, y_i is the ground truth, y_i' is the prediction value, and N is the number of all the prediction values. We divide the input data into three sub-datasets: a training set, validation set, and test set. We feed the training set into the model in batches. For each batch, the model calculates the loss after forward-propagation, and then optimizes all the training parameters by back-propagation with the optimizer Adam (Kingma and Ba 2014). By minimizing the loss function, all training parameters are trained.

$$loss = MSE = \frac{1}{N}\sum_{i\ =1}^{N}(y_i - y_i')^2 \tag{11}$$

4 Case Study

4.1 Experiment data

We validate the proposed DSTR-RNet model by predicting the traffic flow on the road network in the central district of Chengdu, China (Figure 6 (a)), a region of 8-km^2 with 2,616 road segments (Figure 6 (b)).

Figure 6. Location of study area and the road network. (a) Location of study area. (b) Road network of study area. (c) Concrete example of generated traffic flow.

We use the GPS trajectories of Didi taxi cars (Didi Chuxing Corporation) during November 1 to 30, 2016 as the source data to generate the traffic flow. The time interval between two adjacent GPS points of a trajectory is 2-4 s. We downloaded the corresponding road network from OpenStreetMap on January 8, 2018 using the OSMnx tool (Boeing 2017). First, we match the trajectory points to the road network using the typical Hidden Markov Model (Newson and Krumm 2009). We then determine the number of taxis on each road segment during a specified time interval. We generate three datasets with time intervals of 10 min, 20 min, and 60 min (Table 1). Figure 6 (c) shows an example of the traffic flow generated with the 20-min time interval (09:00 am–09:20 am on November 30, 2016). We scale the flow values into the range of [-1, 1] for training and transform all the predicted values to normal values for evaluation.

The training set includes observations from November 01–24, 2016; the validation set is from November 25–26, 2016. We select the last four days, November 27–30, 2016, as the testing period.

Table 1. Descriptions of experimental data sets.

Datasets	**Time Interval**	**Data Range**		**Mean**	**Intervals per Day**
		Min	Max		
Data$_1$	10 min	0	218	8.31	144
Data$_2$	20 min	0	374	16.16	72

Data_3	60 min	0	974	47.56	24

4.2 Evaluation metrics

The root-mean-square error (RMSE) and mean-absolute-percentage-error (MAPE) serve as the evaluation metrics (Equation 12); here, y_i is the ground truth value, y_i' is the prediction value, and N is the number of all the prediction values.

$$\begin{gathered} \text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - y_i')^2} \\ \text{MAPE} = 100\% \times \frac{1}{N}\sum_{i=1}^{N}|y_i - y_i'|/y_i \end{gathered} \tag{12}$$

4.3 Comparative experiment

4.3.1 Comparative models

We select five representative prediction models as the benchmarks for comparison with the proposed DSTR-RNet model (Table 2). The input for the ARIMA model is the entire historical traffic flow time series. For the other models, the inputs are identical to those of the proposed model.

Table 2. Brief introduction of benchmark models.

No.	Name	Description
1	ARIMA	Widely used statistical model for time series forecasting (Box and Pierce 1970).
2	SVR	SVM-based model for prediction. A typical representation of machine-learning methods.

3	LSTM	A widely-used deep RNN; it is suitable for time series prediction. Here, the LSTM model has one weight layer. The model is trained for each road segment.
4	ST-ANN	Identical structure as the proposed DSTR-RNet model; however, its weight layers are conventional fully-connected neural network layers rather than LCNR layers. Residual learning is also not involved.
5	ST-LCNR	Identical structure as the proposed DSTR-RNet model with LCNR as weight layers, albeit without residual learning.
6	CNN	A CNN-based model proposed by Ma et al. (2017); it transforms the road-network-based data into a two-dimensional image with the horizontal axis representing time and vertical axis representing road segment. The model predicts by performing convolutions on the image.

4.3.2 Environment and training settings

We code all the models in Python 3.5. The deep learning libraries for ST-ANN, ST-LCNR, and DSTR-RNet are in TensorFlow (Abadi *et al*. 2016). We implement the LSTM model using Keras (Chollet 2015) with TensorFlow as the backend. The experiment ran on a GPU platform, using NVIDIA GeForce GTX 1080 with 8GB of

GPU memory.

For each of LSTM, ST-ANN, ST-LCNR, CNN, and DSTR-RNet models, the training settings are 100 iterations and early stopping strategy to prevent overfitting (Caruana et al. 2001). We set the sequence lengths of recent, daily and weekly patterns (*r*, *d,* and *w*, defined in Section 2) as $r = 3$, $d = 1$, and $w = 1$, respectively. The initial learning rate is 0.004, and it decays as the iterations increase (Equation (13)):

$$l = L \cdot e^{(-i/S)} \tag{13}$$

where L is the initial learning rate, i is the current iteration epoch, S is the constant decay speed (set to 50), and l is the learning rate of the current iteration. For the experimental datasets, we set the batch sizes as 16 (Data_1), 10 (Data_2), and four (Data_3), so that the numbers of batches are similar. For ST-ANN, ST-LCNR, and DSTR-RNet, the number of weight layers increases gradually; and we record the most effective results.

4.3.3 Experimental results and analysis

Table 3 presents the predictive performances of the proposed DSTR-RNet and the other benchmark models. We calculate the RMSE from all the values in the testing set. For MAPE, we calculate it from the top 10% of the largest testing samples of (MAPE@10%); this is because large values (such as traffic flow at peak hours) attract more attention.

Table 3. Experimental results of all prediction models.

Data	Metrics	ST-ANN	ARIMA	SVR	ST-LCNR	LSTM	CNN	DSTR-RNet

Data_1	RMSE	37.49	3.89	3.64	3.57	3.62	3.58	**3.45**
	MAPE@10(%)	48.05	15.93	15.33	14.93	16.32	16.09	**14.83**
Data_2	RMSE	61.57	6.22	6.19	5.69	5.54	5.58	**5.31**
	MAPE@10(%)	34.69	17.44	16.00	11.94	16.2	12.99	**11.72**
Data_3	RMSE	162.73	15.94	14.66	12.55	12.12	12.32	**11.6**
	MAPE@10(%)	27.50	11.23	14.57	9.49	12.71	9.17	**8.89**

The proposed DSTR-RNet model outperforms the other six benchmarks in both RMSE and MAPE. In the fully-connected ST-ANN, the number of training parameters in one weight layer is exceptionally large (approximately 7 million parameters in each weight layer); this hinders the model training procedure from converging. Therefore, the prediction accuracy is significantly lower than those of the results from the two conventional models. With local connections (ST-LCNR), the number of training parameters in a weight layer reduces to 15,000, substantially less than that in ST-ANN. Hence, the ST-LCNR is successfully trained and achieves better results.

However, the ST-LCNR is incapable of capturing dependency from high-order neighbors as it is limited by the shallow structure of the plain neural network. As shown in Figure 7, the RMSE obtained by ST-LCNR increases notably when the number of weight layers is larger than five. Conversely, the RMSE of DSTR-RNet continues to decrease. By integrating residual learning, the ResLCNR units achieve a structure deeper than the plain LCNR layer, so that the dependency from distant neighbors is captured. More weight layers achieve more complex non-linear transformations, also contributing to higher prediction accuracy.

Figure 7. Comparision between ST-LCNR and DSTR-RNet with different numbers of weight layers. (a) Performance difference for $Data_1$. (b) Performance difference for $Data_2$. (c) Performance difference for $Data_3$.

DSTR-RNet exhibits apparent improvements over LSTM. The relative reductions in RMSE of $Data_1$, $Data_2$, and $Data_3$ are 4.7%, 4.2% and 4.3%, respectively. For MAPE, the reductions are 9.13%, 27.65%, and 30.05% for $Data_1$, $Data_2$, and $Data_3$, respectively. Therefore, for road network data prediction, it is unreasonable to consider only temporal dependency while omitting the spatial and temporal correlations. DSTR-RNet overcomes this limitation by capturing spatio-temporal dependency.

DSTR-RNet also outperforms the CNN model that Ma *et al*. (2017) proposed. Although this CNN-based model represents each road segment separately, it does not consider the topology of the road network. In contrast, the proposed DSTR-RNet considers the road network topology and obtains higher prediction accuracy.

In summary, the comparative experiments establishes that the proposed DSTR-RNet model is capable of capturing spatio-temporal dependency from both near/current and distant/past elements of the road network to achieve higher prediction accuracy.

4.4 Sensitivity analysis

We conduct a sensitivity analysis on DSTR-RNet. The investigation examines the number of ResLCNR units representing the spatially receptive field (model depth) and the input length of different patterns representing the temporal perspective.

4.4.1 Number of ResLCNR units

The sensitivity analysis gradually increases the number of ResLCNR units and records the RMSE values. The other parameters are identical to those in Section 4.3.2. The RMSEs first drop and then rise with the increase in ResLCNR units (Figure 8). Data_1 obtains the highest performance with four ResLCNR units. As each ResLCNR unit has two weight layers, by adding the top LCNR layer (Section 3.3, Figure 5), there are nine weight layers. This implies that for Data_1, the spatio-temporal dependency originates mainly from the ninth-order neighbors in the road network. For the other two test datasets, the best spatially receptive fields are eleventh-order neighbors, which is marginally larger than that of Data_1. This is because a large time interval may generate distant interactions. Then, the increase in ResLCNR units does not produce more effective dependency, although it increases the training complexity and overfitting and eventually decrease the performance. Therefore, when the number of ResLCNR units is between four and six, the DSTR-RNet achieves better prediction results.

Figure 8. Performances for different numbers of ResLCNR units. (a) RMSE trend for Data_1. (b) RMSE trend for Data_2. (c) RMSE trend for Data_3.

4.4.2 Input lengths of different temporal patterns

We investigate the impact of different input lengths for the three patterns. We denote the lengths of recent, daily, and weekly patterns as r, d, and w, respectively. To investigate the impact of the recent pattern, we hold the parameters d and w constant ($d = 1$ and $w = 1$), whereas r can vary. For the daily pattern, we hold the parameters r and w constant ($r = 3$ and $w = 1$), whereas d can vary. Similarly, for the weekly pattern, we hold the parameters r and d constant ($r = 3$ and $d = 1$), whereas w can vary. As the period of the experiment is four weeks, we set the maximum value of w as two.

Figure 9 summarizes the performances of Data_1, Data_2, and Data_3. For all the three datasets, the RMSEs with $r = 0$ and $w = 0$ are larger than those with $d = 0$. This implies that the recent and weekly patterns contribute more to the final predictions than the daily pattern. The optimum length for the recent pattern is three for Data_1 and Data_3. For Data_2, it is either three or four because the same RMSEs are found $r = 3$ and $r = 4$. The performances deteriorate as r increases; this is because more inputs of the recent pattern could introduce noise. The optimum length of the daily and weekly patterns is one, implying that short-range periods are beneficial and that long-range periods are likely to introduce noise. As a result, the model achieves higher performance when the input lengths are r =3 (or 4), $d = 1$ and $w = 1$.

Figure 9. Performances with different input lengths. (a) RMSE trend for Data_1. (b) RMSE trend for Data_2. (c) RMSE trend for Data_3.

5 Discussion

5.1 Spatial distribution of errors

In this section, we consider Data$_2$ as an example to explore the distribution of errors. Because the larger values in the daytime are more applicable to urban management, the analysis focuses on the period between 08:00 and 21:00. The reference is the mean ground truth (MGT) (Equation 14):

$$MGT = \frac{1}{T}\sum_{i=1}^{T} y_i \tag{14}$$

where y_i is the ground truth at the i^{th} time interval and T is the number of time intervals. If the ground truth of a road segment is zero, the corresponding MAPE value is set as -20.

Figure 10. (a)-(c) Spatial distributions corresponding to MGT, RMSE and MAPE of Data$_2$ from 08:00 to 21:00. (d) Directions of all road segments.

The RMSEs are proportional to the ground truths (Figure 10); however, the MAPEs are inversely proportional to the ground truths, a common phenomenon (Lv et al. 2014, Yu et al. 2017). Nevertheless, roads 1 and 2, marked by circles in Figure 10 (d), exhibit differences. The truth value of road 1 is larger than that of road 2; meanwhile, for the RMSE, the situations are inverted. A likely reason is the connectivity of the road. Road 2 is surrounded by one-way neighbors (Figure 10(d)), whereas for road 1, most of the neighbors are two-way. The one-order adjacency matrix used to construct the LCNR layer (Section 3.1) is undirected, omitting the one-way situation; this is likely to bring noise to road 2. As a result, road 2 obtains a larger error. In the future, the direction of the road network will be considered to construct the LCNR layer and explore the impact on prediction results.

5.2 Temporal distribution of errors

This section describes the use of Data$_2$ to explore the error distribution in the temporal dimension. The period examined is 08:00-21:00. For each time interval, we calculate the MGT by the truth values of all the road segments. Figure 11 shows the result. Generally, the distribution is similar to the spatial perspective; herein, RMSE is proportional to the ground truth, whereas MAPE is inversely proportional. However, both the RMSE and MAPE of Monday morning are relatively larger (marked by a rectangle in Figure 11 (b) and (c)). This is because the daily pattern input used to predict Monday scenarios is from Sunday, which has no morning peak (Figure 11 (a)). As a result, larger errors are generated. In the future, the workdays and weekends will be split and trained separately to optimize the problem.

Figure 11. (a)-(c) Temporal distributions corresponding to MGT, RMSE, and MAPE of Data$_2$ from 08:00 to 21:00.

5.3 Difference between DSTR-RNet and grid-based CNN model

This section describes our investigation of the difference between the proposed DSTR-RNet model and grid-based CNN model. We again select Data$_2$ as the example. We convert the road-network-based traffic flow to a grid format for processing by the CNN. The cell sizes adopted by the available models include 10 m

(Yu *et al*. 2017), 500 m (Chen *et al*. 2018), 1,000 m (Zhang *et al*. 2018), and 5,000 m (Ke *et al*. 2017). We consider 10 m to be excessively fine, covering approximately the lengths of two cars. Therefore, the cell size is set to 100 m, a relatively fine-grained resolution. There are 78 rows and 86 columns on the grid map. The value of each grid cell is the mean of all the road segments that crosses the cell. We selected ST-ResNet as the grid-based CNN model for comparison (Zhang *et al*. 2017a). The input lengths of the three patterns are identical to those of DSTR-RNet; we set the convolution kernel size to three.

Figure 12. Comparison between DSTR-RNet and CNN-based model ST-ResNet. (a)-(c) Ground truths, predictions of DSTR-RNet and predictions of ST-ResNet. (d)-(f) Ground truths, predictions of DSTR-RNet and predictions of ST-ResNet for region A.

Because the data ranges are entirely different, it is unreasonable to compare the evaluation metrics directly. We select a time interval with larger flow values (14:00–14:20 on November 30, 2016) to investigate the difference in spatial precision. A typical overpass connecting the east-west Second Ring Road and the main north-south Beixing Dadao Road (we have marked it with an *A* in Figure 12) exhibits the details. With the grid-based representation, we could obtain only the mean flow value of all roads (Figure 12 (c)); we could not predict the actual flow of each road segment. The spatial precision of the original road network is not maintained. However, the proposed road-network-based DSTR-RNet model uses the road segments as the unit upon which to make predictions; this maintains the spatial precision(Figure 12 (c)). The CNN-based model does not consider the topology of the road network. ST-ResNet employs numerous grid cells not crossed by any road to conduct the convolution. DSTR-RNet overcomes this limitation by integrating the topological adjacencies of the road network. Therefore, from the perspective of both spatial principles and practical application, the proposed DSTR-RNet is more suitable than the grid-based CNN method, for modeling road-network-based data.

6 Conclusions

This research proposes a deep spatio-temporal residual neural network for road-network-based data modeling (DSTR-RNet), bringing forth a new deep learning solution at the road-network level rather than at the grid level. The proposed LCNR employs the topology of the road network to model the local spatial dependency. Then, we integrate residual learning into the LCNR to form a deeper structure, ResLCNR unit; this enables us to model the spatial dependency from near to distant. Based on the ResLCNR unit, three sub-models integrally model the spatial and temporal dependency from different temporal patterns, forming the final predictions. We test the model on a case study in Chengdu, China by predicting the traffic flow of Didi cab service in an 8-km^2 region with 2,616 road segments. We explore the sensitivity of the DSTR-RNet model to its parameters. We also discuss the spatial and temporal distribution of the prediction errors and compare DSTR-RNet and a grid-based CNN model.

We draw the following conclusions from this research:

(1) The local connections constructed by topology in the proposed LCNR layer considerably reduce the trainable complexity; this enables us to model a neural network representing a larger road network.

(2) The proposed ResLCNR unit increases the model depth, enabling us to model the spatial dependency from near to distant.

(3) The proposed DSTR-RNet achieves a deep-learning-based spatio-temporal modeling method at the road-network level rather than the grid level. This maintaines the spatial precision and topology of the road network and improves the prediction accuracy.

In the future, we intend to optimize the DSTR-RNet by using a directed adjacency matrix of the road network to construct a directed LCNR layer. We also intend to consider external factors such as weather, day (weekday/weekend), and holidays. In addition, we intend to seek more data to test the future optimized model.

Acknowledgments

The authors thank Didi Chuxing for providing the experiment data source. The authors thank Dr. May Yuan, Dr. Huanfa Chen, and the anonymous reviewers for their insightful comments.

Funding

This work is part of the Consumer Data Research Centre (CDRC) project supported by the UK Economic and Social Research Council (ES/L011840/1). It is supported by the Science and Technology Project of Qingdao under Grant number 16-6-2-61-NSH; The first author’s joint Ph.D. research and the third author’s Ph.D research are funded by the China Scholarship Council (CSC). The CSC is a non-profit institution with legal person status affiliated with the Ministry of Education in China.

References

Abadi, M., *et al.* 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Boeing, G. 2017. OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems,* 65, 126-139.

Box, G. E. and Pierce, D. A. 1970. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association,* 65(332), 1509-1526.

Caruana, R., Lawrence, S. and Giles, C. L., Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. ed. *Advances in neural information processing systems*, 2001, 402-408.

Chen, J., *et al.* 2018. Fine-grained prediction of urban population using mobile phone location data. *International Journal of Geographical Information Science*, 1-17.

Cheng, T., Haworth, J. and Wang, J. 2011. Spatio-temporal autocorrelation of road network data. *Journal of Geographical Systems,* 14(4), 389-413.

Cheng, T. and Wang, J. 2008. Integrated Spatio-temporal Data Mining for Forest Fire Prediction. *Transactions in GIS,* 12(5), 591-611.

Cheng, T. and Wang, J. 2009. Accommodating spatial associations in DRNN for space–time analysis. *Computers, Environment and Urban Systems,* 33(6), 409-418.

Cheng, T.*, et al.* 2014. A Dynamic Spatial Weight Matrix and Localized Space–Time Autoregressive Integrated Moving Average for Network Modeling. *Geographical Analysis,* 46(1), 75-97.

Chollet, F. 2015. Keras: Deep learning library for theano and tensorflow. *URL: https://keras. io/k,* 7, 8.

Chuxing, D., *Didi Chuxing* [online]. Available from: https://outreach.didichuxing.com.

Haworth, J.*, et al.* 2014. Local online kernel ridge regression for forecasting of urban travel times. *Transportation Research Part C: Emerging Technologies,* 46, 151-178.

He, K.*, et al.*, Deep Residual Learning for Image Recognition. ed. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 27-30 June 2016 2016a, 770-778.

He, K.*, et al.*, Identity mappings in deep residual networks. ed. *European Conference on Computer Vision*, 2016b, 630-645.

Hinton, G. E. and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *Science,* 313(5786), 504-507.

Hoang, M. X., Zheng, Y. and Singh, A. K., FCCF: forecasting citywide crowd flows based on big data. ed. *The ACM Sigspatial International Conference*, 2016, 1-10.

Huang, W.*, et al.* 2015. Predicting human mobility with activity changes. *International Journal of Geographical Information Science,* 29(9), 1569-1587.

Jiang, B. 2009. Ranking spaces for predicting human movement in an urban environment. *International Journal of Geographical Information Science,* 23(7), 823-837.

Jiang, B. and Liu, C. 2009. Street-based topological representations and analyses for predicting traffic flow in GIS. *International Journal of Geographical Information Science,* 23(9), 1119-1137.

Ke, J.*, et al.* 2017. Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. *Transportation Research Part C: Emerging Technologies,* 85, 591-608.

Kingma, D. and Ba, J. 2014. Adam: A Method for Stochastic Optimization. *Computer Science*.

Krizhevsky, A., Sutskever, I. and Hinton, G. E., ImageNet classification with deep convolutional neural networks. ed. *International Conference on Neural Information Processing Systems*, 2012, 1097-1105.

Lecun, Y., Bengio, Y. and Hinton, G. 2015. Deep learning. *Nature,* 521(7553), 436-444.

Li, X.*, et al.* 2016. T-DesP: Destination Prediction Based on Big Trajectory Data. *IEEE Transactions on Intelligent Transportation Systems,* 17(8), 2344-2354.

Lv, Y.*, et al.* 2014. Traffic Flow Prediction With Big Data: A Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems*, 1-9.

Ma, X.*, et al.* 2017. Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction. *Sensors,* 17(4).

Ma, Z.*, et al.* 2014. Predicting short-term bus passenger demand using a pattern hybrid approach. *Transportation Research Part C: Emerging Technologies,* 39, 148-163.

Newson, P. and Krumm, J., Hidden Markov map matching through noise and sparseness. ed. *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, 2009, 336-343.

Rosser, G., *et al.* 2016. Predictive Crime Mapping: Arbitrary Grids or Street Networks? *Journal of Quantitative Criminology,* 33(3), 569-594.

Schmidhuber, Jurgen, 2015. *Deep learning in neural networks.* Elsevier Science Ltd.

Shaw, S.-L., Tsou, M.-H. and Ye, X. 2016. Editorial: human dynamics in the mobile and big data era. *International Journal of Geographical Information Science,* 30(9), 1687-1693.

Stockwell, D. 1999. The GARP modelling system: problems and solutions to automated spatial prediction. *International Journal of Geographical Information Science,* 13(2), 143-158.

Wang, J., *et al.*, STARIMA for journey time prediction in London. ed. *Proceedings of the 5th IMA conference on mathematics in transport*, 2010.

Wang, J., Cheng, T. and Li, X., Nonlinear Integration of Spatial and Temporal Forecasting by Support Vector Machines. ed. *International Conference on Fuzzy Systems and Knowledge Discovery*, 2007, 61-66.

Wang, J., Tsapakis, I. and Zhong, C. 2016. A space–time delay neural network model for travel time prediction. *Engineering Applications of Artificial Intelligence,* 52(C), 145-160.

Xingjian, S., *et al.*, Convolutional LSTM network: A machine learning approach for precipitation nowcasting. ed. *Advances in neural information processing systems*, 2015, 802-810.

Yu, H., *et al.* 2017. Spatiotemporal Recurrent Convolutional Networks for Traffic Prediction in Transportation Networks. *Sensors (Basel),* 17(7).

Zhang, J., Zheng, Y. and Qi, D., Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. ed. *AAAI*, 2017a, 1655-1661.

Zhang, J., *et al.*, DNN-based prediction model for spatio-temporal data. ed. *ACM Sigspatial International Conference on Advances in Geographic Information Systems*, 2016, 92.

Zhang, J., *et al.* 2017b. Predicting Citywide Crowd Flows Using Deep Spatio-Temporal Residual Networks. *arXiv preprint arXiv:1701.02543*.

Zhang, J., *et al.* 2018. Predicting Citywide Crowd Flows Using Deep Spatio-Temporal Residual Networks ☆. *Artificial Intelligence*.

Zhu, X. and Guo, D. 2014. Mapping Large Spatial Flow Data with Hierarchical Clustering. *Transactions in GIS,* 18(3), 421-435.

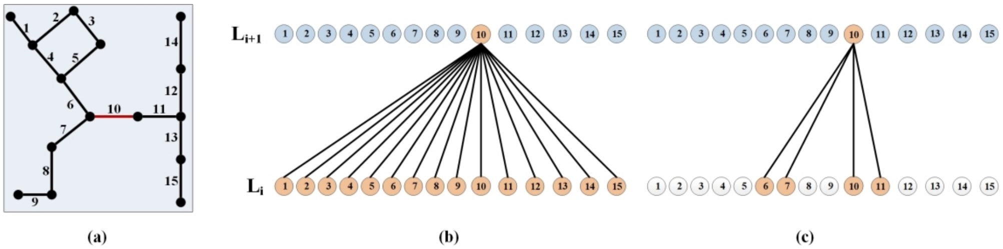
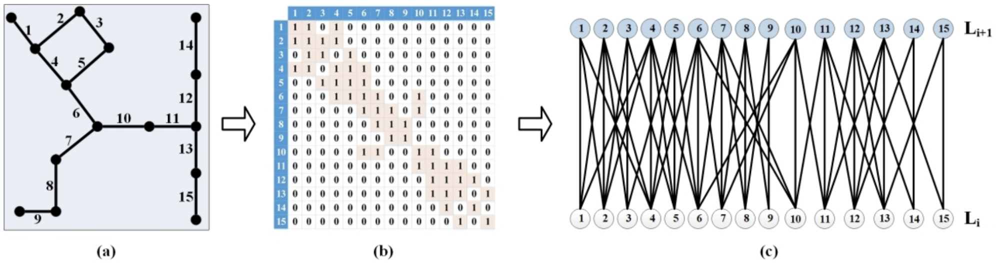


Figure 1. Comparison of fully-connected layer and LCNR layer. (a) A road network example with 15 segments. (b) Node 10 connects to all the nodes. (c) Node 10 locally connects to its first-order neighbors.

208x52mm (300 x 300 DPI)



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
3	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0
4	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0
5	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
6	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0
7	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0
8	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
9	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
10	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0
11	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
12	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
13	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1
14	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
15	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Figure 2. LCNR layer constructed by first-order spatial adjacency matrix. (a) Road network. (b) First-order spatial adjacency matrix W1. (c) There are 53 local connections in an LCNR layer.

191x50mm (300 x 300 DPI)

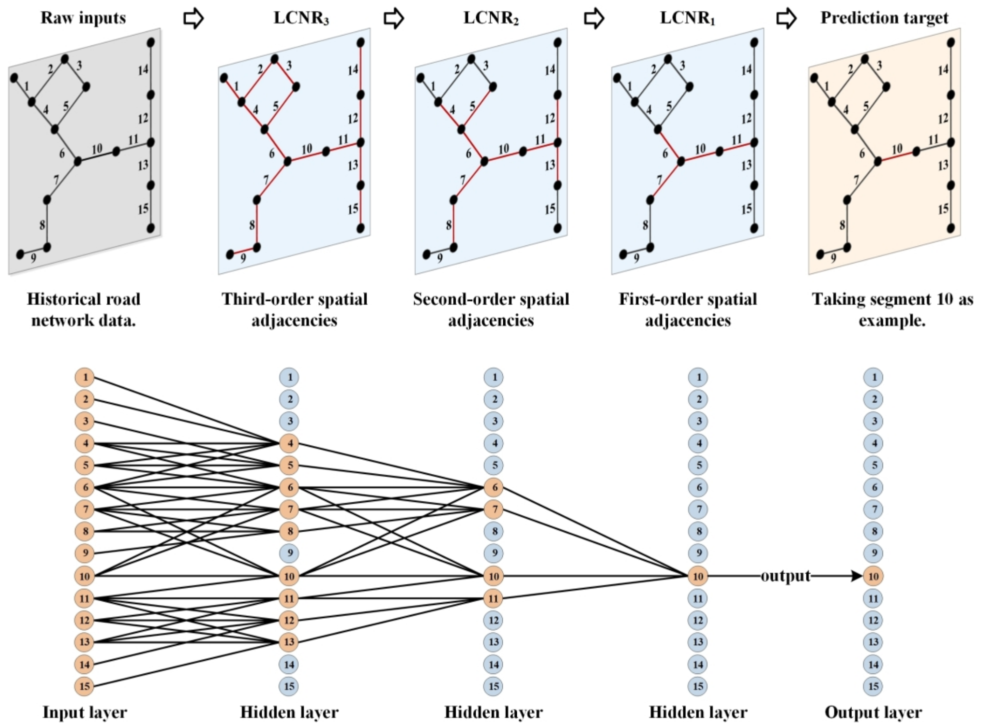

Raw inputs
LCNR$_3$
LCNR$_2$
LCNR$_1$
Prediction target
Historical road network data.
Third-order spatial adjacencies
Second-order spatial adjacencies
First-order spatial adjacencies
Taking segment 10 as example.
output
Input layer
Hidden layer
Hidden layer
Hidden layer
Output layer

Figure 3. Deep LCNR model with three weight layers.

199x145mm (300 x 300 DPI)

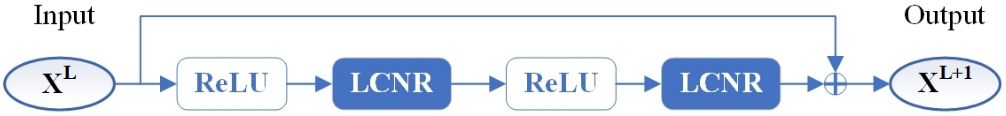


Figure 4. A ResLCNR unit.

95x11mm (300 x 300 DPI)

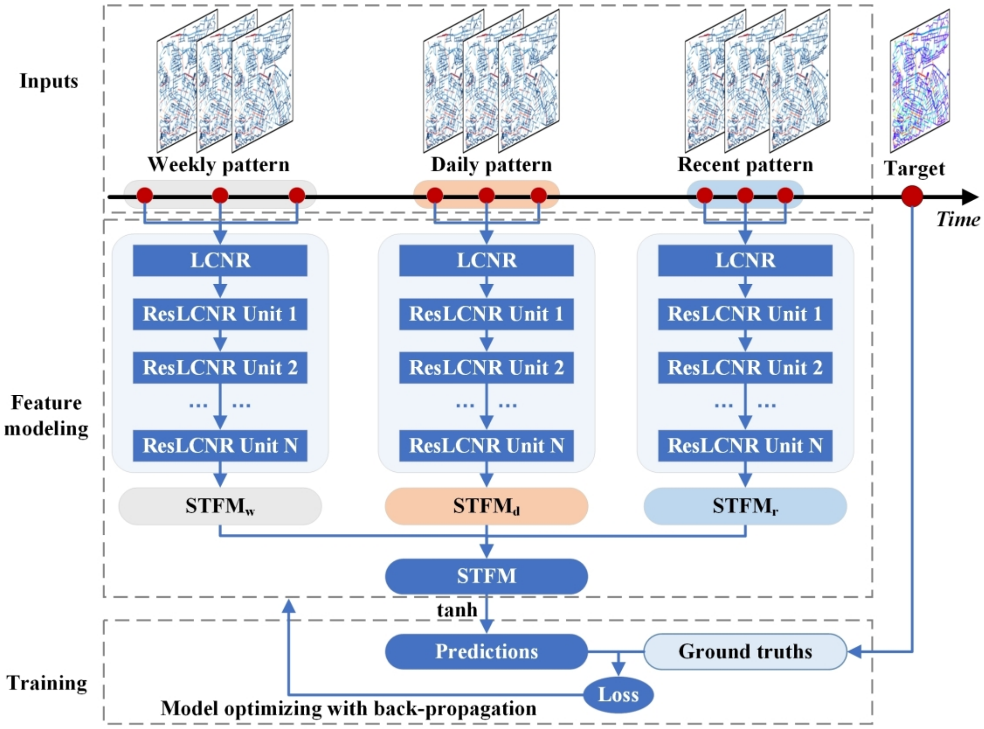


Figure 5. Framework of DSTR-RNet.

196x142mm (300 x 300 DPI)

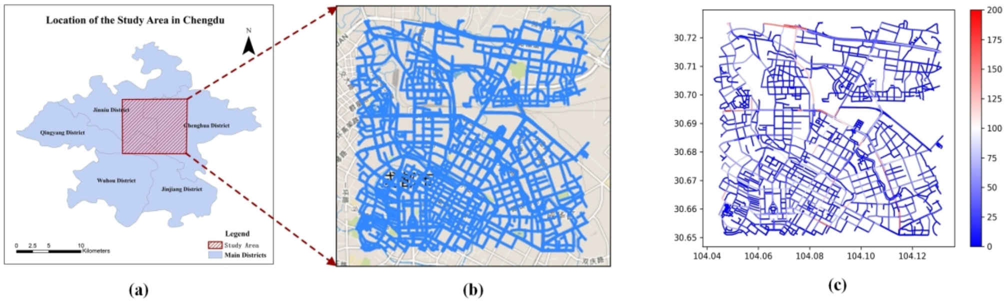


Figure 6. Location of study area and the road network. (a) Location of study area. (b) Road network of study area. (c) Concrete example of generated traffic flow.

203x62mm (300 x 300 DPI)

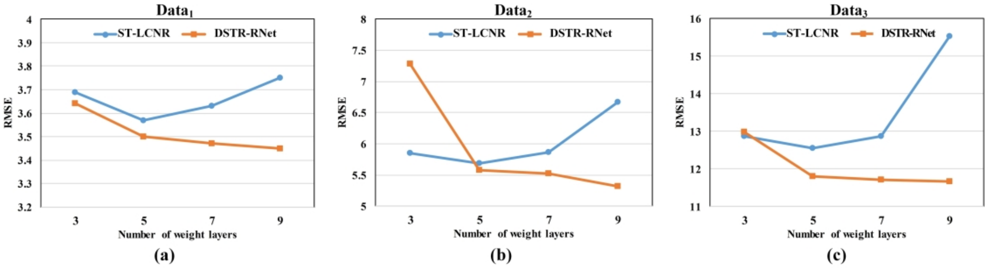


Figure 7. Comparision between ST-LCNR and DSTR-RNet with different numbers of weight layers. (a) Performance difference for Data1. (b) Performance difference for Data2. (c) Performance difference for Data3.

177x49mm (600 x 600 DPI)

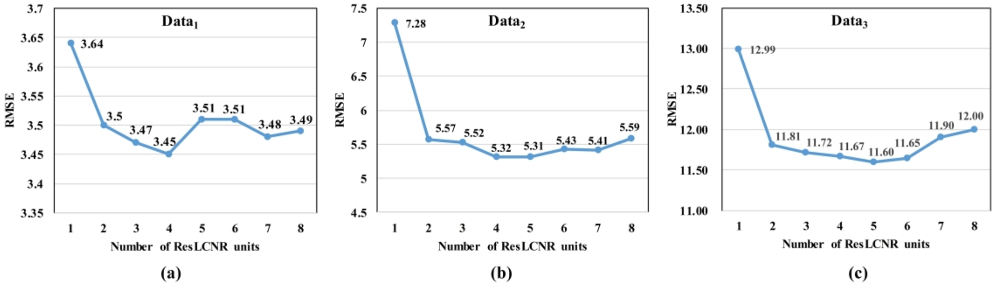
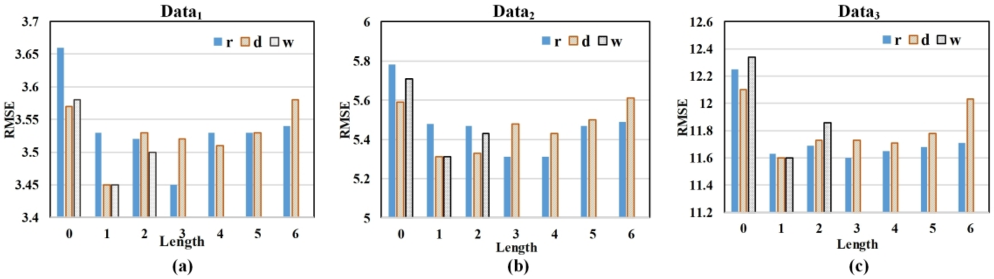


Figure 8. Performances for different numbers of ResLCNR units. (a) RMSE trend for Data1. (b) RMSE trend for Data2. (c) RMSE trend for Data3.

239x69mm (600 x 600 DPI)



(a)

(b)

(c)

Figure 9. Performances with different input lengths. (a) RMSE trend for Data1. (b) RMSE trend for Data2. (c) RMSE trend for Data3.

234x68mm (300 x 300 DPI)

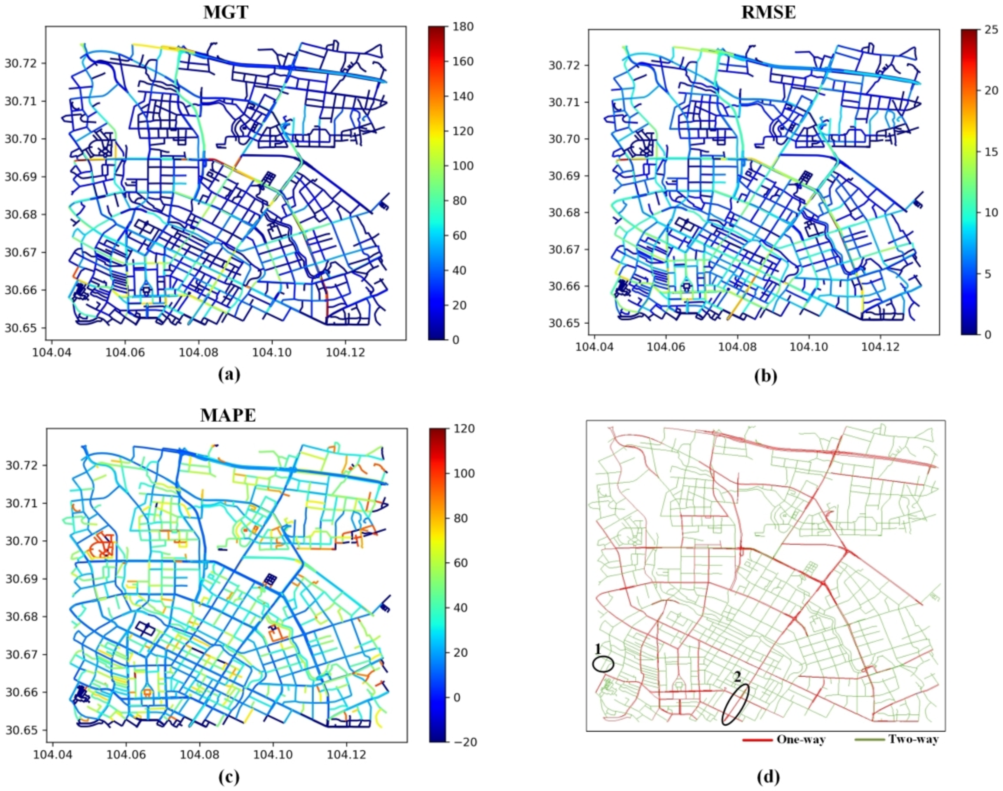


Figure 10. (a)-(c) Spatial distributions corresponding to MGT, RMSE and MAPE of Data2 from 08:00 to 21:00. (d) Directions of all road segments.

219x176mm (300 x 300 DPI)

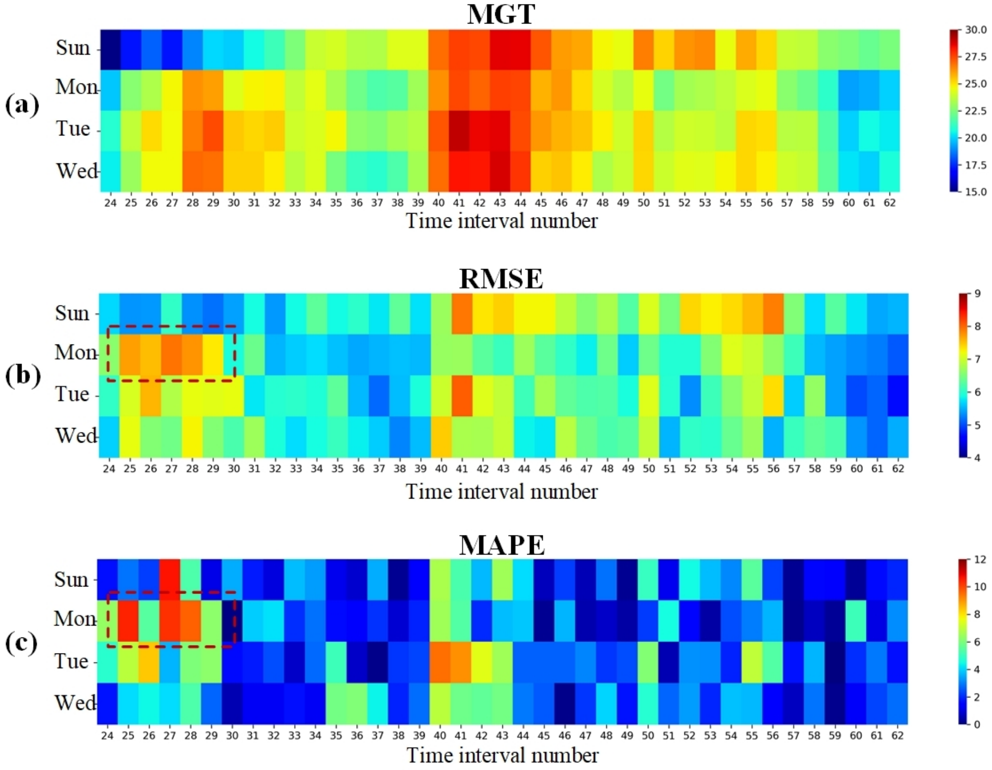


Figure 11. (a)-(c) Temporal distributions corresponding to MGT, RMSE, and MAPE of Data2 from 08:00 to 21:00.

95x75mm (300 x 300 DPI)

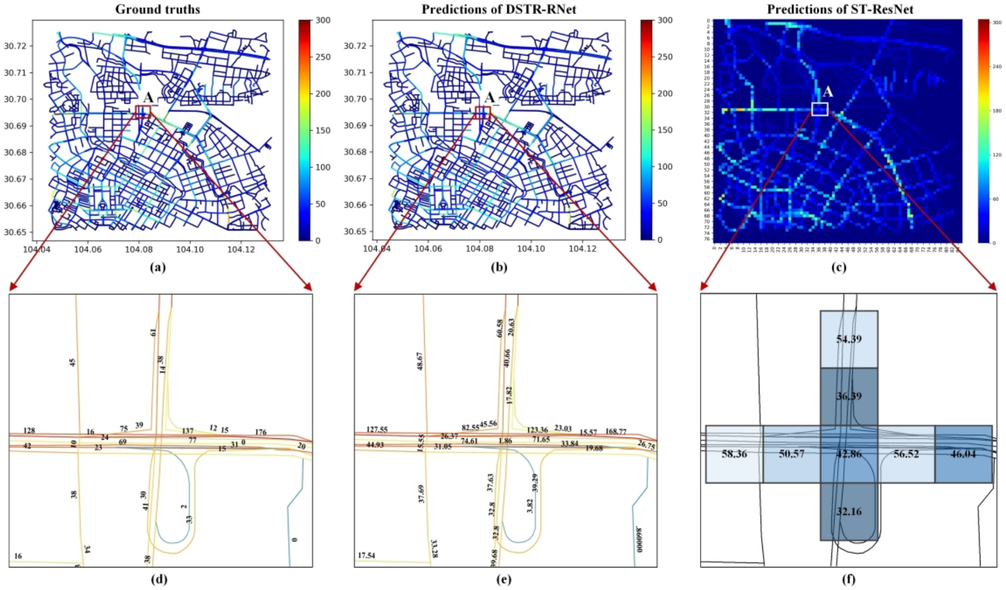


Figure 12. Comparison between DSTR-RNet and CNN-based model ST-ResNet. (a)-(c) Ground truths, predictions of DSTR-RNet and predictions of ST-ResNet. (d)-(f) Ground truths, predictions of DSTR-RNet and predictions of ST-ResNet for region A.

310x184mm (300 x 300 DPI)