DeepRTP: A Deep Spatio-Temporal Residual Network for Regional Traffic Prediction

Zhidan Liu*†, Mingliang Huang†, Zhi Ye†, Kaishun Wu*†
*Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University, China
†College of Computer Science and Software Engineering, Shenzhen University, China
liuzhidan@szu.edu.cn, brness13@gmail.com, scofieldye@163.com, wu@szu.edu.cn

***Abstract*—Accurate traffic prediction can benefit many smart city applications. Existing works mainly consider traffic prediction on each individual road segment, and heavily rely on some statistical or machine learning models, which suffer from either poor prediction accuracy or high computation overheads for predictions of the whole road network. In this paper, we instead consider the region-level traffic prediction that is still useful for many applications. To describe the regional traffic conditions and capture their spatio-temporal dependencies, we present a deep learning based model - *DeepRTP*. Specifically, we use a novel metric called Traffic State Index (*TSI*) to measure regional traffic conditions, and carefully classify traffic data into three categories that are used to capture hourly, daily, and weekly traffic patterns. Furthermore, we employ the convolutional and residual neural networks to model both spatial and temporal dependencies. Experimental results from real-world traffic data demonstrate that *DeepRTP* outperforms five baseline methods and can achieve higher prediction accuracy.**

***Index Terms*—traffic prediction; deep residual network; spatio-temporal dependency**

I. Introduction

Comprehensive traffic information benefit urban citizens' daily life and improve the efficiency of urban transportation. Accurate predictions of such traffic information are of great importance for many smart city applications, *e.g.*, transport network planning, route guidance, and congestion avoidance [22]. In recent years, the popularity of ubiquitous sensing and intelligent transport systems has enabled continuous and large-scale urban traffic monitoring, and thus massive mobility data could be collected for data-driven traffic prediction [18].

In the literature, there are many research efforts made to achieve accurate traffic prediction. These works primarily apply various models to analyze historical and real-time traffic data to predict traffic conditions in the near future. In general, some statistical models, *e.g.*, k-nearest neighbors (KNN) [21] and autoregressive integrated moving average (ARIMA) [17], and machine learning models, *e.g.*, support vector regression (SVR) [20] and artificial neural network (ANN) [13], are leveraged to model the urban traffic [12]. Due to the shallow architectures of these models, however, prior methods cannot well capture the complicated spatio-temporal correlation of urban traffic and thus have low prediction accuracy.

Recently, the emerging deep learning has drawn much attention due to its multi-layer architecture that could discover intricate structures and complex patterns [8]. Various advanced deep learning models, *e.g.*, stacked autoencoder (SAE) [14], convolutional neural networks (CNN) [15], [19], long short-term memory neural network (LSTM) [10], [16], have been used for different kinds of traffic prediction tasks [12]. Although these methods can derive better results than the traditional methods, existing attempts still mainly focus on the traffic prediction of a road segment or a small-scale road network [15]. Specifically, one deep learning model is built for predicting the traffic on a road segment, and thus a large number of such models are needed for the whole road network. As a result, existing works suffer from poor scalability due to the enormous computation overheads involved in the training of many deep learning models [13].

In this paper, instead of predicting traffic condition for each road segment, we would like to predict the coarse traffic information at the region level. Specifically, we divide a road network into a set of regions, where each region may cover a number of road segments, and predict the traffic conditions of all regions simultaneously based on some powerful traffic model. Region-level traffic prediction will greatly reduce the computation complexity, while still sufficiently supports many applications, *e.g.*, attribute-aware similar region search that regards regional traffic condition as one important characteristic of a region [2] and adaptive fastest path computation that needs to firstly pre-compute an area-level path given the traffic conditions of all regions [3]. In addition, regional traffic prediction would benefit prior works by deciding whether to launch fine-grained traffic prediction or not. For example, it is unnecessary to execute the complex deep learning model based traffic predictions for the road segments, whose locating regions have been predicted with good traffic condition, since such road segments should be in good traffic condition as well.

However, it is challenging to achieve efficient and accurate regional traffic prediction for at least two reasons. First, prior works mainly use traffic speed or traffic volume to measure the traffic condition of a road segment, there exists no effective metric to well describe the traffic condition of a region. Second, the traffics in a road network are extremely complex due to the spatial and temporal dependencies, and how to encode such spatio-temporal dependencies into the traffic model for accurate regional traffic prediction is important yet difficult.

To tackle above challenges, in this paper we present *DeepRTP* – a *Deep* spatio-temporal residual network model for *R*egional *T*raffic *P*rediction. The contributions of our work can be summarized as follows:

- We consider and formally define the problem of regional traffic prediction, and adopt a novel metric named Traffic State Index (*TSI*) to measure the traffic congestion of a region. To the best of our knowledge, this is the first work to predict traffic condition at region-level.
- *DeepRTP* models the spatio-temporal traffic dependencies with deep convolutional and residual neural networks [4]. Specifically, it encodes the traffic conditions of all regions into a traffic matrix, and adopts convolutional layers to automatically extract the spatial dependencies among regions. In addition, we summarize temporal dependencies of regional traffics into three categories, corresponding to hourly, daily, and weekly traffic patterns. *DeepRTP* employs three residual networks to model these properties respectively, and wisely fuses their outputs to derive the final traffic prediction of the whole road network.
- We conduct experiments with a real-world traffic dataset to evaluate *DeepRTP*. Experimental results demonstrate that *DeepRTP* can achieve accurate traffic prediction and significantly outperforms the five baseline methods.

The rest of this paper is organized as follows. We review the related works in Section II, and present the problem statement in Section III. The design of *DeepRTP* is detailed in Section IV. In Section V, we conduct experiments to evaluate *DeepRTP*. Section VI concludes this paper.

II. Related Work

Traffic prediction usually applies traffic models to analyze both historical and real-time traffic data for predicting traffic conditions in the future [12], which would benefit drivers and pedestrians [6]. Typically, traffic speed and traffic volume are used as the indicators to measure the traffic conditions, and some statistical or advanced machine learning models are widely adopted to mine traffic data for the prediction [18]. Pan *et al*. propose a hybrid traffic prediction approach that automatically selects ARIMA model or historical average model to predict traffic speeds on road segments by statistically analyzing the traffic data [17]. Yu *et al*. adopt the k-nearest neighbor algorithm for short-term traffic condition prediction [21]. In addition to the statistical models, traditional machine learning models are frequently used for traffic prediction as well. For example, Liu *et al*. propose the dynamic artificial neural network (ANN) modeling to predict traffic speeds of road segments by exploiting the road network correlation [13]. Wu *et al*. apply support vector machine (SVM) for travel time prediction on a highway network [20]. In the past years, many other models, *e.g.*, Kalman filtering and Bayesian model, have been utilized for traffic prediction [18]. These methods, however, may not well model the complex traffics and thus can not achieve high prediction accuracy.

Recently, unprecedented data availability [9] and the ability to rapidly process these data together make possible the immense development of deep learning theory [8]. Various deep learning models have also been successfully used for different kinds of traffic prediction tasks [12]. Lv *et al*. use the stacked autoencoder (SAE) model to learn generic traffic flow features, which are further fed into a logistic regression layer for traffic flow prediction [14]. Ma *et al*. convert traffic dynamics into images and adopt the convolutional neural network (CNN) model for traffic speed prediction [15]. Similarly, Wang *et al*. rely on CNN models for traffic speed prediction and attempt to explore the congestion sources [19]. Taking auxiliary information like POIs and online crowd queries into consideration, Liao *et al*. exploit the long short-term memory (LSTM) neural network [16] for more accurate traffic prediction [10]. Furthermore, Zheng *et al*. firstly apply the residual learning technique for the prediction of citywide crowd flows [23]. Although these works could achieve better results, they mainly focus on the traffic prediction of a road segment or a small-scale road network [15]. They usually build a deep learning model for each individual road segment, and many such models are needed for the entire road network, which will introduce unacceptable computation overheads [13]. Different from previous works, we propose a novel metric to measure the traffic condition at region-level and use only one deep learning model to predict city-scale traffic conditions instantly.

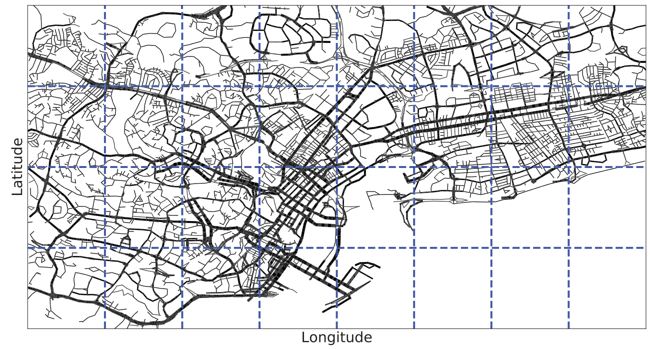


Fig. 1. The testing road network, which is partitioned into 4×8 regions for regional traffic prediction.

III. Problem Statement

We model the underlying road network for traffic prediction as a graph $\mathtt{G}(\mathtt{V}, \mathtt{E})$, where road segments are represented as the vertices and edges are formed between any two physically connected road segments. For regional traffic prediction, we partition graph $\mathtt{G}$ into regions $\mathbb{R} = \{R_1, \cdots, R_M\}$. Figure 1 shows the regions of our testing road network.

Definition 1: (**Region**) A road network is partitioned into an $m \times n$ grid map based on the longitude and latitude, where a grid represents a region and covers a number of road segments.

We adopt a novel metric named *Traffic State Index (TSI)* to measure the traffic condition of a region. For region R_i, the calculation of its TSI_{R_i} will jointly consider several aspects of each road segment r_j in region R_i, *i.e.*,

$$TSI_{R_i} = \frac{\sum_{r_j \in R_i} \frac{l_j \times k_j \times [v_j^f - v_j]^+}{v_j^f}}{\sum_{r_j \in R_i} l_j \times k_j} \times 100, \qquad (1)$$

where l_j, k_j, v_j^f, and v_j are the length of road segment r_j, the number of lanes in road segment r_j, speed limit of road segment r_j, and current traffic speed of road segment r_j, respectively. The function $[x]^+$ in Equation (1) is defined as

$$[x]^+ = max(0, x), \tag{2}$$

which always returns a non-negative value. In principle, a larger TSI_{R_i} means more congested traffics in region R_i, and the road segments in this area are more likely to be in poor traffic condition as well.

For traffic prediction, we divide the time into slots with size of Δ. In the t-th time slot, we make use of the traffic data to calculate *TSI* values for all regions, which can be denoted by an $m \times n$ matrix $\mathbf{X}_t$[1]. Therefore, we formally define the regional traffic prediction problem as follows.

Definition 2: (**Regional Traffic Prediction Problem**) Given the historical and current *TSI* matrices $\mathbf{X}_1, \cdots, \mathbf{X}_t$, the regional traffic prediction problem aims to predict $\mathbf{X}_{t+1}$.

IV. The Design of *DeepRTP*

In this section, we present the system overview of *DeepRTP* and then elaborate each module in the following subsections.

A. System Overview

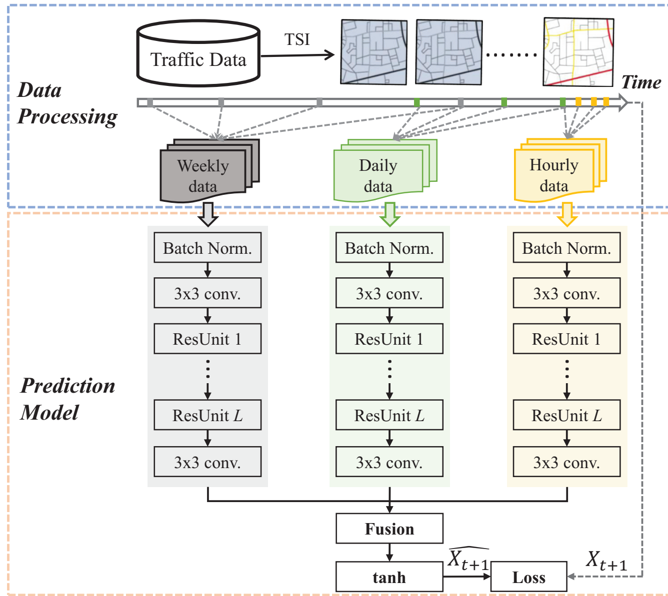


Fig. 2. The system architecture of *DeepRTP*.

The system architecture of *DeepRTP* is illustrated in Figure 2. At the high level, *DeepRTP* takes historical and real-time traffic data as the input, and predicts the regional traffic conditions of the next time slot. Specifically, the *Data Processing* module calculates the *TSI* values for all regions with traffic data using Equation (1). In particular, the traffic data of each time slot are converted to a 2-channel image-like traffic matrix. All traffic matrices are then categorized into three groups according to their temporal properties, which are used to capture the temporal dependencies of traffics from three aspects of weekly, daily, and hourly. These traffic matrices are then fed into three components of the *Prediction Model* module to separately capture the temporal dependencies. The three components share the same network structure, which is built with convolutional layers and residual units. Such a network structure can well capture spatial dependency of traffics and residual units could support deeper networks. The outputs of the three components are fused together based on some learned weight matrices. Finally, the aggregated result is inputted into the `tanh` function to derive the final prediction.

B. Data Processing

We divide the entire time duration covered by all traffic data into a series of time slots with a given slot size Δ. For the t-th time slot, we compute *TSI* value for each region based on the average traffic speeds of all road segments in the region using Equation (1). The *TSI* values of all regions form a traffic matrix $\mathbf{X}_t$, which describes the general traffic conditions of the whole road network. Existing studies have demonstrated that there exist traffic patterns across time of the day and day of the week [11], [23]. Therefore, we categorize all traffic matrices into three groups, namely *hourly data*, *daily data*, and *weekly data*, which implicitly capture traffic patterns from different temporal aspects to predict $\mathbf{X}_{t+1}$. Specifically, the traffic matrices are classified into different groups as follows.

- *Hourly data.* The traffic conditions in the $(t+1)$-th time slot are strongly influenced by the traffics in the past time slots. Thus, *DeepRTP* takes the recent h traffic matrices, *i.e.*, $\mathbf{X}_t, \mathbf{X}_{t-1}, \cdots \mathbf{X}_{t-h+1}$, to learn such a temporal closeness dependency for traffic prediction.
- *Daily data.* To capture the daily traffic pattern, *DeepRTP* groups the traffic matrices of the same time slot of the past few days, which implicitly reveals the traffic trends. Specifically, assume that there are totally T_d time slots in a day, and thus such traffic matrices in recent d days, *i.e.*, $\mathbf{X}_{t+1-T_d}, \mathbf{X}_{t+1-2\times T_d}, \cdots, \mathbf{X}_{t+1-d\times T_d}$, will be grouped into this category.
- *Weekly data.* Similarly, *DeepRTP* also builds a group to capture weekly traffic pattern. The traffic matrices of the same time slot of a day and the same day of a week are used to learn such a periodical pattern. Specifically, this category includes traffic matrices in recent w weeks, *i.e.*, $\mathbf{X}_{t+1-7\times T_d}, \mathbf{X}_{t+1-2\times 7\times T_d}, \cdots, \mathbf{X}_{t+1-w\times 7\times T_d}$.

C. The Prediction Model

As shown in Figure 2, *DeepRTP* adopts the same deep learning network structure to model temporal dependencies of traffics in hourly, daily, and weekly three aspects. The deep learning model is composed of two components: convolutional layer and residual unit. The outputs of the three networks are then fused for the final prediction through the `tanh` function. In addition, a *loss* function is designed to train this prediction model. Next, we will introduce each component of the model in detail as follows.

Convolutional layer. Convolutional neural network (CNN) is widely used to hierarchically capture spatial structural infor-

[1]The entries for regions with no road segments are filled with zeros.

mation and has already been successfully applied to computer vision domain for image processing [8], [15]. Thus we adopt several convolutional layers to capture the spatial dependency of traffics with kernel size of convolution as 3×3, *i.e.*, the traffic of one region is spatially correlated with the traffics of its neighboring regions. One convolutional layer captures near spatial dependency for a region, and thus many convolutional layers could capture distant spatial dependency [23].

For each category of traffic matrices, each traffic matrix $\mathbf{X}$ is firstly processed by the *Batch Normalization* [5] with its output denoted by $\mathbf{X}^0$, and then fed into the convolutional layer with the following transformation:

$$\mathbf{X}^1 = f(W^1 * \mathbf{X}^0 + b^1), \tag{3}$$

where $*$ denotes the convolutional operation; f is the rectifier function as the activation function, *i.e.*, $f(x) = \max(0, x)$; and W^1 and b^1 are the learnable parameters in the first convolutional layer [7].

Residual unit. To avoid the problem of vanishing gradients in most deep learning models [8], we make use of the residual learning technique [4] to support a very deep network to well capture traffic dependencies. Deep residual learning allows the CNN model to have a super deep structures of over 100-layers, and has been successfully applied in some challenging tasks, *e.g.*, image classification and object detection. Specifically, *DeepRTP* stacks L residual units after the first convolutional layer as shown in Figure 2. Each residual unit transforms the input data $\mathbf{X}^\ell$ as follows:

$$\mathbf{X}^{\ell+1} = \mathbf{X}^\ell + \mathcal{F}(\mathbf{X}^\ell, \theta^\ell),\ \ell = 1, \cdots, L, \tag{4}$$

where $\mathcal{F}$ is the residual function; and θ^ℓ is the parameter to be learned in the ℓ-th residual unit. We adopt the similar residual unit structure as [23], which uses two combinations of one *ReLU* function and one convolutional layer.

Fusion and Loss function. The outputs of the three deep learning components for weekly data, daily data, and hourly data are denoted by $\mathbf{X}_w^{L+2}$, $\mathbf{X}_d^{L+2}$, and $\mathbf{X}_h^{L+2}$, respectively. *DeepRTP* fuses these outputs using the following equation:

$$\mathbf{X}_{all} = W_w \circ \mathbf{X}_w^{L+2} + W_d \circ \mathbf{X}_d^{L+2} + W_h \circ \mathbf{X}_h^{L+2}, \tag{5}$$

where $\circ$ represents the function of element-wise multiplication; W_w, W_d, and W_h are the learnable parameters, which adjust the influence weights of weekly pattern, daily pattern, and hourly pattern on the future traffics.

DeepRTP predicts the $(t+1)$-th *TSI* values of all regions, denoted by $\hat{\mathbf{X}}_{t+1}$, with the hyperbolic tangent function, *i.e.*, `tanh`. The model is trained with historical traffic data using the mean squared error to design the loss function as

$$\mathcal{L}(\lambda) = ||\mathbf{X}_{t+1} - \hat{\mathbf{X}}_{t+1}||_2^2, \tag{6}$$

where λ are all learnable parameters in the prediction model.

V. Performance Evaluation

In this section, we conduct extensive experiments to evaluate *DeepRTP* with the real-world traffic data.

A. Experimental Setup

Dataset. We use an open traffic speed dataset of our testing city to evaluate *DeepRTP*. We export the road network of the testing city with the open-sourced OpenStreetMap [1], and focus on the downtown area for the regional traffic prediction. Figure 1 shows the testing road network, which in total has 17017 road segments. For each road segment, we have a series of average traffic speeds for 15-minute time slots, with a duration of two months. We partition the testing road network into regions and calculate their *TSI* values using Equation (1). The *TSI* values are scaled into the range $[-1.0, 1.0]$ with the Min-Max normalization method. We thus derive a series of traffic matrices. For the performance evaluations, we choose traffic matrices from the last week as the testing data and treat all the remaining traffic matrices as the training data.

By default, we set the size of time slots as $\Delta = 15\, minutes$, and partition the experimental road network into 4×8 regions. We set $L = 2$ as the number of residual units. In addition, we set $h = 3$, $d = 3$, and $w = 1$ to classify traffic matrices into hourly, daily, and weekly data groups. To train the prediction model, we set batch size and learning rate as 4 and 0.0002, respectively.

Baselines. We compare *DeepRTP* with the following five alternative methods.

- *HisAve* predicts the *TSI* value of a region as the average value of all *TSI* values of this region in the same time slot of each historical day [17].
- *HisAve-w* works similarly as *HisAve* while it takes the weekly pattern into consideration. This method predicts the *TSI* value of a region as the average value of all *TSI* values of this region in the same time slot of a day and the same day of a week in history [23].
- *ARIMA* is a well-known time series analysis model for forecasting future values [17].
- *SARIMA* represents the seasonal ARIMA, which takes the weekly pattern into consideration as well [23].
- *VAR* is the advanced vector autoregressive model that captures pairwise-relationships among traffics of all regions. It will introduce heavy computation overheads due to the huge number of parameters [20].

We implement and run all methods on a server with an Intel Core i7 6700@$3.4\, GHz$ and $8\, GB$ memory. For each baseline method, we optimize its parameters to achieve the best performance.

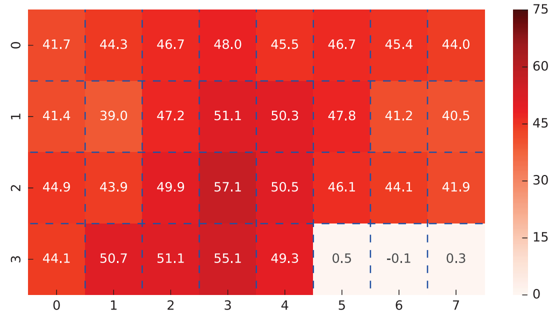
Performance metric. We use the rooted mean squared error (*RMSE*) as the metric to evaluate all methods, *i.e.*,

$$RMSE = \sqrt{\frac{1}{N}\sum_i (x_i - \hat{x}_i)^2}, \tag{7}$$

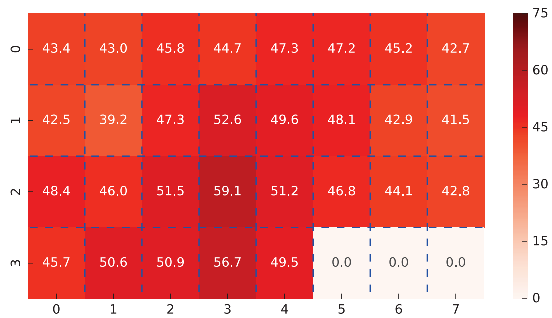
where x_i and $\hat{x}_i$ are the ground truth and prediction, respectively; N is the total number of predictions. Note that $\hat{x}_i$ is the normal value by re-scaling the output of the prediction model.

B. Experimental Results

Visualization of traffic predictions. We have implemented a simple traffic visualization platform to demonstrate the traffic



(a) Predicted *TSIs*



(b) Real *TSIs*

Fig. 3. Visualizations of traffic prediction results and the ground truth. The three regions at the bottom-right have no road segments and thus their *TSI* values are filled with zeros.

TABLE I
THE PERFORMANCE COMPARISONS AMONG DIFFERENT METHODS.

Method	***RMSE***
HisAve	2.40
HisAve-w	2.25
ARIMA	2.05
SARIMA	2.10
VAR	1.91
DeepRTP	1.02
DeepRTP without hourly component	1.12
DeepRTP without daily component	1.04
DeepRTP without weekly component	1.05

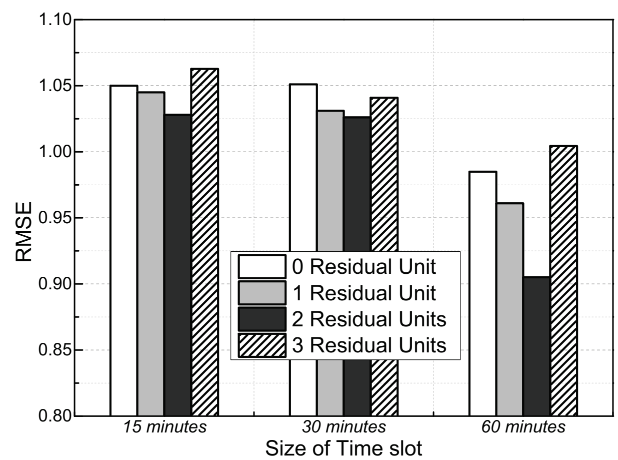


Fig. 4. Performance impacts of model settings with different time slot sizes and different numbers of residual units.

prediction results. Figure 3 presents an instance of the traffic prediction results for a time slot in the peak hours of a typical workday. By comparing the predicted *TSIs* in Figure 3(a) and the ground truth *TSIs* in Figure 3(b), we find that *DeepRTP* can accurately predict the regional traffic conditions.

Comparison results. We compare *DeepRTP* with the baseline methods and present their results in Table I. By comparing *HisAve* and *HisAve-w*, we see that the weekly pattern information can indeed improve the prediction accuracy, by reducing *RMSE* from 2.40 to 2.25. *ARIMA*-like methods perform slightly better than the historical average based methods, while *VAR* has the best *RMSE* result among the five baselines. From Table I, we find that our method *DeepRTP* significantly outperforms these baselines, with at least 87% reduction on the *RMSE* value.

Furthermore, we conduct extra experiments to examine three variants of *DeepRTP* by disabling one of the three components, *i.e.*, the three networks to model hourly pattern, daily pattern, and weekly pattern. Table I shows that each component indeed contributes to the overall prediction accuracy, while the hourly pattern has the largest influence on the prediction performance of *DeepRTP*, since disabling the component of hourly pattern leads to the maximum increase on the *RMSE* value. Obviously, the traffic conditions of the next time interval are more related with the recent traffic conditions.

Impact of model settings. We also study the performances of *DeepRTP* by varying the time slot size Δ and the number of residual units, and plot the results in Figure 4. Specifically, we derive the traffic speeds for each road segment by averaging every 2 and 4 original traffic speeds when we set $\Delta = 30\,minutes$ and $\Delta = 60\,minutes$, respectively. In general, we find the *RMSE* values become smaller when we enlarge Δ from $15\,minutes$ to $60\,minutes$, which means that larger time slot size leads to higher prediction accuracy. This is because traffic condition of a region becomes more stable within a long period, and thus it will be more predicable.

Figure 4 also shows the performances of *DeepRTP* with different number of residual units, *i.e.*, the setting of L. We see that the *RMSE* value decreases as the number of residual units increases. With no residual unit, the *RMSE* value of *DeepRTP* is pretty large and these results demonstrate that residual learning indeed can help CNN models to avoid the gradient vanishing problem and achieve better performances. For the traffic matrix of size 4×8, we find $L = 2$ can well capture city-scale traffic dependencies and thus achieve the best performances for all Δ settings. A larger L, however, will impair the performance, as reported in Figure 4.

Impact of road network partitioning. Figure 5 shows the performances of *DeepRTP* when we partition the road network into different numbers of regions. The traffic prediction results become worse as the number of regions increases. With finer partitioning, traffic dependencies among regions become more complicated and much deeper network may be needed.

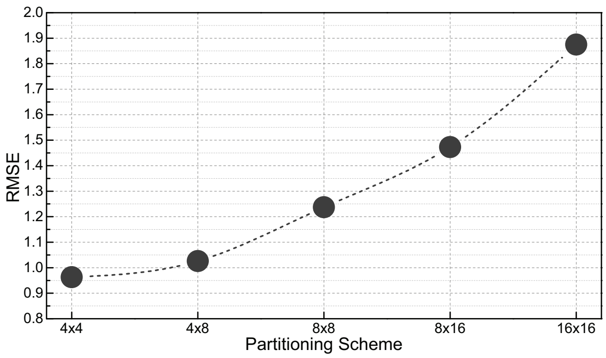


Fig. 5. Performance comparisons of different partitioning schemes.

VI. Conclusion

This paper presents *DeepRTP* for regional traffic predictions. *DeepRTP* adopts a novel *TSI* metric to measure the traffic condition of a region, and proposes a deep spatio-temporal residual network model for accurately predicting the *TSI* values of all regions. Experimental results from real-world traffic data demonstrate that *DeepRTP* significantly outperforms five baseline methods and can achieve high prediction accuracy.

Acknowledgment

This work was supported in part by China NSFC Grant (No.61802261); NSF Grant of Shenzhen University (No.2018061); and Tencent "Rhinoceros Birds" - Scientific Research Foundation for Young Teachers of Shenzhen University. This work was also partially supported by China NSFC Grant (No.61872248); Guangdong NSF Grant (No.2017A030312008); Fok Ying-Tong Education Foundation for Young Teachers in the Higher Education Institutions of China (No.161064); and GDUPS (2015).

References

[1] Openstreetmap. http://www.openstreetmap.org/.

[2] K. Feng, G. Cong, C. S. Jensen, and T. Guo. Finding attribute-aware similar regions for data analysis. In *VLDB*, 2019.

[3] H. Gonzalez, J. Han, X. Li, M. Myslinska, and J. P. Sondag. Adaptive fastest path computation on a road network: a traffic mining approach. In *VLDB*, 2007.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE CVPR*, 2016.

[5] S. Ioffe and C. Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[6] Y. Jiang, Z. Li, and J. Wang. PTrack: enhancing the applicability of pedestrian tracking with wearables. In *IEEE ICDCS*, 2017.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.

[8] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[9] Z. Li, M. Li, J. Wang, and Z. Cao. Ubiquitous data collection for mobile users in wireless sensor networks. In *IEEE INFOCOM*, 2011.

[10] B. Liao, J. Zhang, C. Wu, D. McIlwraith, T. Chen, S. Yang, Y. Guo, and F. Wu. Deep sequence learning with auxiliary information for traffic prediction. In *ACM SIGKDD*, 2018.

[11] Z. Liu, Z. Li, M. Li, W. Xing, and D. Lu. Mining road network correlation for traffic estimation via compressive sensing. *IEEE Transactions on Intelligent Transportation Systems*, 17(7):1880–1893, 2016.

[12] Z. Liu, Z. Li, K. Wu, and M. Li. Urban traffic prediction from mobility data using deep learning. *IEEE Network*, 32(4):40–46, 2018.

[13] Z. Liu, P. Zhou, Z. Li, and M. Li. Think like a graph: real-time traffic estimation at city-scale. *IEEE Transactions on Mobile Computing*, 18(10):2446–2459, 2019.

[14] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang. Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):865–873, 2014.

[15] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang. Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, 17(4):818, 2017.

[16] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, 54:187–197, 2015.

[17] B. Pan, U. Demiryurek, and C. Shahabi. Utilizing real-world transportation data for accurate traffic prediction. In *IEEE ICDM*, 2012.

[18] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias. Short-term traffic forecasting: where we are and where we're going. *Transportation Research Part C: Emerging Technologies*, 43:3–19, 2014.

[19] J. Wang, Q. Gu, J. Wu, G. Liu, and Z. Xiong. Traffic speed prediction and congestion source exploration: a deep learning method. In *IEEE ICDM*, 2016.

[20] C.-H. Wu, J.-M. Ho, and D.-T. Lee. Travel-time prediction with support vector regression. *IEEE transactions on intelligent transportation systems*, 5(4):276–281, 2004.

[21] B. Yu, X. Song, F. Guan, Z. Yang, and B. Yao. K-nearest neighbor model for multiple-time-step prediction of short-term traffic condition. *Journal of Transportation Engineering*, 142(6):04016018, 2016.

[22] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen. Data-driven intelligent transportation systems: a survey. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1624–1639, 2011.

[23] J. Zhang, Y. Zheng, and D. Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *AAAI*, 2017.