Deep Spatial–Temporal 3D Convolutional Neural Networks for Traffic Data Forecasting

Shengnan Guo, Youfang Lin, Shijie Li, Zhaoming Chen, and Huaiyu Wan

***Abstract*— Reliable traffic prediction is critical to improve safety, stability, and efficiency of intelligent transportation systems. However, traffic prediction is a very challenging problem because traffic data are a typical type of spatio-temporal data, which simultaneously shows correlation and heterogeneity both in space and time. Most existing works can only capture the partial properties of traffic data and even assume that the effect of correlation on traffic prediction is globally invariable, resulting in inadequate modeling and unsatisfactory prediction performance. In this paper, we propose a novel end-to-end deep learning model, called ST-3DNet, for traffic raster data prediction. ST-3DNet introduces 3D convolutions to automatically capture the correlations of traffic data in both spatial and temporal dimensions. A novel recalibration (Rc) block is proposed to explicitly quantify the difference of the contributions of the correlations in space. Considering two kinds of temporal properties of traffic data, i.e., local patterns and long-term patterns, ST-3DNet employs two components consisting of 3D convolutions and Rc blocks to, respectively, model the two kinds of patterns and then aggregates them together in a weighted way for the final prediction. The experiments on several real-world traffic datasets, viz., traffic congestion data and crowd flows data, demonstrate that our ST-3DNet outperforms the state-of-the-art baselines.**

***Index Terms*— Traffic prediction, spatial–temporal data, neural networks, 3D convolutions, recalibration block.**

I. Introduction

IN MODERN Intelligent Transportation Systems (ITS) [1] and Advanced Traveler Information Systems (ATIS), traffic prediction is regarded as an indispensable part to provide accurate and reliable traffic information for travelers and traffic agencies [2], [3]. Knowing traffic information (e.g., traffic congestion conditions, traffic volumes and crowd flows) in advance, authorities can carry out better traffic management strategies and travelers can make better routing plans. Therefore, accurate traffic forecasting helps to reduce the amount of time cost, economic losses and carbon emission.

The goal of traffic prediction is to provide future traffic information in advance based on historical traffic measurements in order to help people make better travel decisions. However, accurate traffic prediction in the real world is very challenging, affected by the following complex factors:

- **Spatial correlation**: In traffic domain, the observations gained at nearby locations are correlated with each other, leading to local coherence in space.
- **Temporal correlation**: Observations at the adjacent time intervals are highly relevant. For example, a traffic congestion taking place during the evening rush hour at around 17:00 p.m. will probably last to 19:00 p.m.
- **Heterogeneity**: Traffic data show heterogeneity both in space and time, i.e., the contributions of the correlations are not globally the same. For one instance, owing to the variation of the functions and the actual geographical relationships of different locations, the influences of their neighborhood are not invariable. For another instance, traffic data show cyclical patterns due to the effect of human daily routine. Even the strength of temporal periodicity also varies with regions, due to different regional functions.

Actually, over the last few decades, many attempts have been done to make more accurate traffic forecasting. Traffic forecasting methodology can be generally classified into two major categories: model-driven approaches and data-driven approaches. Model-driven approaches are also known as parametric approaches, like time series models, which are predetermined based on strong theoretical assumptions. However, the assumptions are hardly satisfied in practice, which limits their forecasting performance. Recently, the deployment of tremendous traffic sensors such as in-ground loop detectors and GPS devices, makes us easy to get a huge amount of real-time traffic data. This provides a good opportunity to deep understand traffic data through data-driven approaches. Data-driven methods can be divided into two subcategories: traditional machine learning methods and deep learning based methods. Traditional machine learning methods lack the ability of dealing with high-dimensional data and can hardly describe the complex and non-linear changes in traffic data. Besides, the forecasting performance of traditional machine learning heavily relies on the handcrafted features which are highly problem-dependent and rely on expert experience. Hence, this kind of methods has weak generality. Fortunately, the emerging of deep learning theory makes it possible to effectively model high-dimensional spatio-temporal data and to automatically discover intricate features through hierarchical representations [4]. Inspired by this, many researchers begin to turn to deep learning based models to deal with

Manuscript received June 15, 2018; revised January 8, 2019; accepted February 21, 2019. This work was supported by the National Natural Science Foundation of China under Grant 61603028. The Associate Editor for this paper was Y. Lv. *(Corresponding author: Huaiyu Wan.)*

The authors are with the Beijing Key Laboratory of Traffic Data Analysis and Mining, School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China, and the Key Laboratory of Intelligent Passenger Service of Civil Aviation, CAAC, Beijing, 101318, China (e-mail: guoshn@bjtu.edu.cn; yflin@bjtu.edu.cn; shijieli@bjtu.edu.cn; zhaomingchen@bjtu.edu.cn; hywan@bjtu.edu.cn).

Digital Object Identifier 10.1109/TITS.2019.2906365

high-dimensional spatial-temporal traffic data. For instance, convolutional neural networks are adopted to explore spatial features of traffic data; recurrent neural networks are employed for modeling sequential information in traffic data. However, the features from both spatial and temporal dimensions, i.e., the motion information encoded in multiple inputs from contiguous time intervals, are not well considered. Furthermore, besides extracting spatio-temporal correlation in traffic data, taking the heterogeneity of the correlations' contribution into account is also crucial in modeling spatio-temporal traffic data, while it is often ignored in most of the existing methods [5].

To tackle the aforementioned challenges, we propose a deep learning based spatio-temporal traffic forecasting network, named ST-3DNet, to predict future traffic data. The contributions of our study are four-fold:

- We introduce 3D convolutions into traffic prediction domain. ST-3DNet adopts 3D convolutions and residual units to effectively extract features from both spatial and temporal dimensions.
- A novel architectural unit termed as "Recalibration" (Rc) block is proposed to explicitly describe the difference of the contributions of the spatio-temporal correlations in space.
- We consider two temporal properties of traffic data, i.e., local temporal patterns and long-term temporal patterns, and respectively design two components to model them and then aggregate their outputs in a weighted way.
- Extensive experiments are carried out on three real-world traffic datasets. The experimental results verify the state-of-the-art performance of our model on different traffic prediction tasks.

II. Literature Review

A. Traffic Prediction

Traffic prediction is regarded as a fundamental module in ITS, helping to improve traffic management. Over the past few decades, traffic prediction has attracted wide attentions from researchers. In general, traffic prediction methods fall into three major categories: time-series analysis methods, traditional machine learning methods and deep learning based methods.

In time-series analysis methods, ARIMA was used for short-term traffic flow prediction [6]. Afterwards, KARIMA [7], SARIMA [8] were proposed to further improve traffic forecasting performance. However, the time-series analysis models are predetermined according to some ideal assumptions, while in the real world, traffic data is too complicated to satisfy the assumptions. So the time-series analysis methods usually perform poorly in practice.

Compared with the above methods, data-driven methods provide more flexible alternatives for traffic forecasting [9]. For example, SVM [10] and SVR [11] were applied to predict traffic data by mapping low-dimensional nonlinear traffic data into high-dimensional space through a kernel function. And the key of these models is the choice of the kernel function, which greatly affects forecasting performance. Besides, Bayesian approaches [12], [13] and KNN [14], [15] were also employed for traffic prediction. Although traditional machine learning methods have solid mathematical foundations and can help us to understand the mechanisms of data generation, they fail to make accurate predictions when dealing with complex and highly nonlinear data [16] and need careful feature engineering.

In recent years, deep learning models with remarkable ability of handling multi-dimensional and nonlinear data inspire more and more researchers to apply them to traffic data mining. Deep belief networks (DBN) [17] and stacked autoencoder models [18], [19] were used to automatically learn the features of traffic flow. But they only deal with a single region every time. In fact, traffic measurements from nearby or even distant regions have spatial and temporal correlation, so predicting future traffic data only based on information from local region is far from enough. Convolution neural networks (CNNs) as a classical type of deep neural networks have achieved numerous breakthroughs in the field of computer vision [4]. Since CNNs can automatically and hierarchically capture the spatial structural information by convolution operations [20], researchers applied them to traffic data prediction. To address the crowd flows prediction issue, Zhang *et al.* [21] proposed a ST-ResNet, which consists of convolution layers and residual units to model citywide spatial dependencies. While as for temporal features, ST-ResNet simply treats information in adjacent time intervals as multiple channels, so right after the first convolution layer, ST-ResNet loses temporal information of the input. Recurrent neural networks (RNNs) as another classical type of deep neural networks are good at modeling temporal information. However, vanilla RNNs suffer from the vanishing gradient problem which makes it difficult for RNNs to remember long-term information. In order to learn temporal dependencies over a long-range time span, long short term memory neural networks (LSTMs) [22] and gated recurrent unit networks (GRUs) [23] are developed. Ma *et al.* [2] and Zhao *et al.* [24] employed LSTM-based networks to predict short-term traffic speed and traffic flow. However, these models can not capture spatial features automatically and the spatial information must be encoded into input manually. In order to settle spatio-temporal sequence forecasting problems, novel convolutional LSTM networks (ConvLSTMs) [25] were proposed. However, because of their complex architectures, training becomes more difficult when the networks' depths increase, which limits their depths and capabilities of capturing wide-range spatio-temporal correlation. In addition, these LSTM-based networks cannot efficiently capture long-range temporal correlation, such as the periodic and trend patterns in traffic data, which is extremely essential in long-term forecasting.

The development of traffic prediction methods has been motivated by the requirement of simultaneously taking both spatial and temporal correlations of traffic data into account. But until now, the features from both spatial and temporal dimensions have not been adequately described. Furthermore, correlations in traffic data are heterogeneous in time and space. Thus, correctly identifying and quantifying the strength of

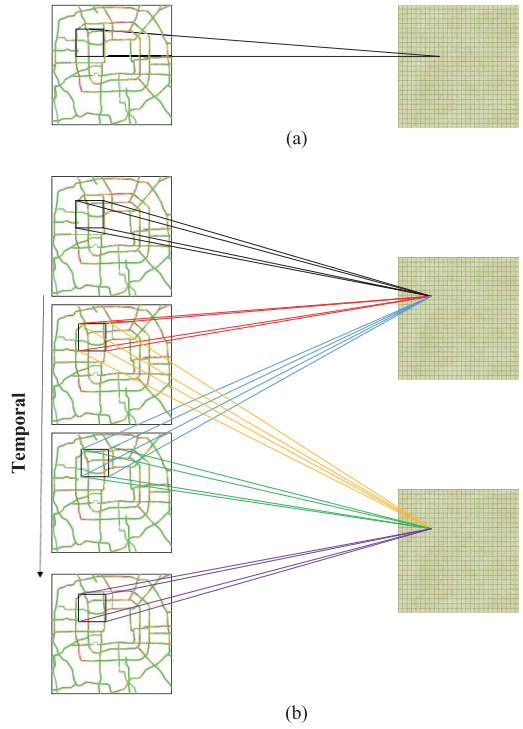


Fig. 1. The difference between (a) 2D and (b) 3D convolutions.

correlations is necessary in modeling traffic data [5]. However, this is often ignored in most existing methods. Actually, most current models assume that the spatial and temporal correlations in traffic data can be described by globally fixed parameters. For instance, in STARIMA [26], the orders of AR and MA are fixed globally both spatially and temporally, which violates the heterogeneity of traffic data. As another example, although ST-ResNet benefits from shared convolution operations to extract spatial features, it still performs convolution operations on the top layer, implying that it considers the effects of channel-wise features in different regions are the same. Yue and Yeh [27] pointed out in order to capture the heterogeneity, related studies can be divided into two categories: those aiming at learning the effective range of correlations [5], [28], and those aiming at capturing the strength of correlations [29], [30].

B. 3D Convolution Neural Networks

How to effectively learn the spatio-temporal information without relying on handcrafted features is also a key concern in video analysis. Tran *et al.* [31] and Ji *et al.* [32] developed 3-dimensional convolutional neural networks (3D CNNs) to learn the spatio-temporal features and they found 3D convolutions are more effective for capturing spatio-temporal features compared to 2D convolutions. Fig. 1 shows a comparison of 2D and 3D convolutions. In Fig. 1(a), a 2D convolution performed at a convolutional layer can only extract features in spatial dimensions from local neighborhood in the previous layer and result in one feature map. When a 2D convolution is applied on a video volume, multiple contiguous frames should be treated as multiple channels, as shown in Fig. 2(b). So after

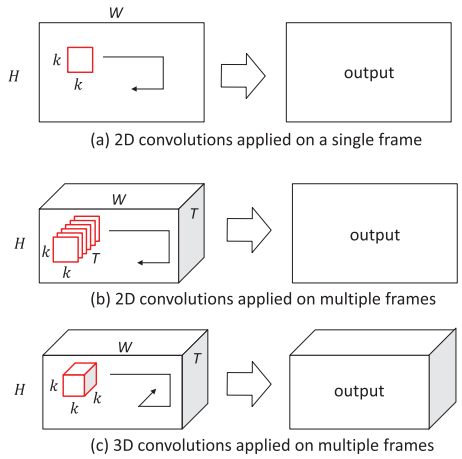


Fig. 2. Convolutions applied on multiple frames.

a 2D convolution operation, multiple frames are compressed into only one feature map and the temporal information is missing. While a 3D convolution operation is to convolve a 3D filter on a cube generated by stacking multiple contiguous frames, as illustrated in Fig. 1(b) and Fig. 2(c). This can effectively capture motion information because one feature map is connected to multiple contiguous frames in the previous layer. Compared to a 2D convolution, applying a 3D convolution still can generate a video volume, which preserves the temporal information. As shown in Fig. 2(c), a 3D convolution that convolves a 3D filter on a cube consisting of multiple contiguous frames can generate a feature map which is also a cube.

Traffic data at each time interval can be analogized as a video's frame. Therefore, motivated by the success of 3D CNNs in video analysis, in this paper we firstly apply 3D convolution in the field of traffic prediction, aiming at automatically modeling spatio-temporal information encoded in traffic data. A novel network structure named spatio-temporal 3D network (ST-3DNet) is proposed to predict spatio-temporal traffic data.

III. Methodology

In this section, we first give the definitions of relevant concepts involved in our paper, and then describe our proposed model.

A. Spatio-Temporal Traffic Raster Data

There are many types of spatio-temporal (ST) data in the real world. Atluri *et al.* [33] divided ST data into four categories, including event data, trajectory data, point reference data and raster data. Among them, raster data refers to observations of a continuous ST field, which is collected at fixed locations in space and fixed points in time. ST raster data is quite common in real-world applications in traffic domain. In this study, we focus on traffic data which is in the form of raster data.

Definition 1 (Traffic Raster Data): In traffic raster data, traffic observations are recorded at fixed intervals in time and

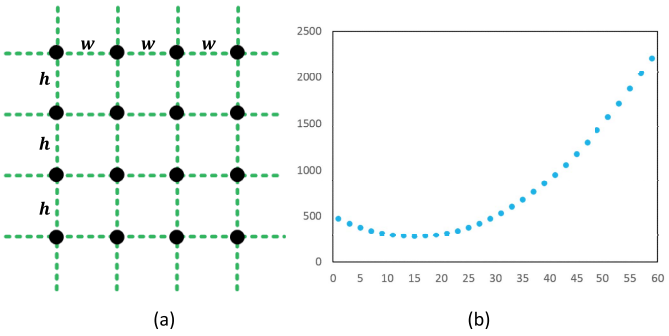


Fig. 3. Traffic raster data.

at fixed locations in space. As shown in Fig. 3(a), consider a set of fixed locations $S = \{s_{ij}\}$, distributed regularly in space with constant distance between adjacent locations, which are like pixels in an image. For every location (i, j), observations $\{x_t^{c,i,j}\}$ are recorded at a fixed set of time intervals $T = \{\tau_t\}$, which are regularly spaced with equal delays between consecutive measurements (see Fig. 3(b)). $x_t^{c,i,j} \in \mathbb{R}$ represents the c-th kind of traffic measurement in location (i, j) at time interval t. Thus, at time interval t, observations in all $I \times J$ locations can be denoted as a tensor $X_t \in \mathbb{R}^{C \times I \times J}$, where $(X_t)_{c,i,j} = x_t^{c,i,j}$.

Many traffic measurements can be represented in raster data. For example, a city can be partitioned into an $I \times J$ grid map according to the longitude and latitude, where a grid denotes a fixed region. When we study the traffic congestion prediction problem, let $c \in \{0\}$ and $x_t^{0,i,j}$ present the traffic congestion condition at time interval t in region (i, j); when we study the problem of crowd flows prediction, let $c \in \{0, 1\}$ and $x_t^{0,i,j}$ and $x_t^{1,i,j}$ respectively be the inflow and outflow of the crowds at time interval t in region (i, j).

B. Problem Definition

Problem 1 (Traffic Raster Data Prediction): Given the historical ST traffic raster data $\{X_t | t = 0, \ldots, n\}$, the goal is to predict $X_{n+\Delta t}$ at time interval $(n + \Delta t)$, where n is the index of the last observed time interval and Δt is the length of time intervals between the last observed one and the predicted one.

C. Deep Spatio-Temporal 3D Convolutional Neural Network (ST-3DNet)

To solve the problem of spatio-temporal traffic prediction, we propose an end-to-end deep learning based model, named ST-3DNet. Fig. 4 displays its overall architecture, which consists of two major components respectively designed to describe two kinds of temporal properties of traffic data, i.e., closeness and weekly period. Closeness refers to the local temporal patterns in traffic data. Obviously, current traffic data is closely related to the recent historical data. Weekly period refers to the periodic and trend patterns in traffic data. For example, Fig. 5 shows the traffic conditions of a region in Beijing in two weeks. In Fig. 5, the periodic patterns in traffic data can be easily noticed, e.g., the repeated Monday morning rush hours. Besides, with the change of seasons, traffic data shows some trend patterns, e.g., when winter comes, the morning rush hours will become later. Traffic data in each region has both the two temporal properties. However, the influence of the two temporal properties on each region are various, e.g., some working areas show obvious periodic patterns, while some entertainment places do not.

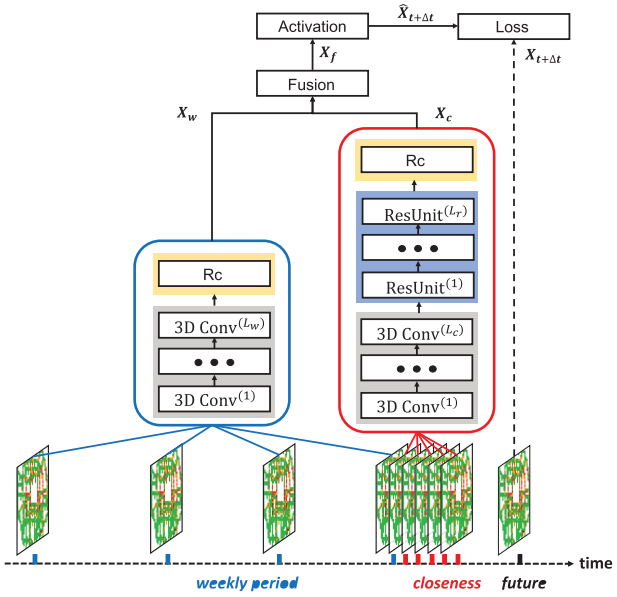


Fig. 4. ST-3DNet architecture. 3D Conv: 3D Convolution; ResUnit: Residual Unit; Rc: Recalibration block.

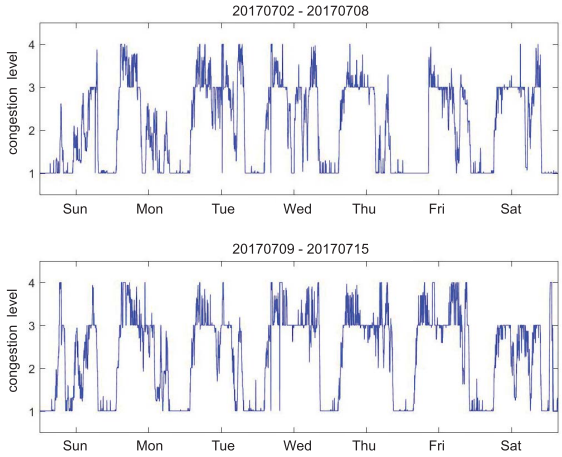


Fig. 5. Weekly periodicity of a road link in Beijing.

The aim of the closeness component is to capture spatio-temporal features based on the recent historical data. Hence, its input is a subsequence of ST raster data in the recent time segment. Let the subsequence be $X_c^{(0)} = [X_{t-d_c}, X_{t-(d_c-1)}, \ldots, X_{t-1}] \in \mathbb{R}^{C \times I \times J \times d_c}$, where d_c is the length of the closeness dependent sequence. The closeness component first stacks 3D convolution layers and 2D residual units to capture spatio-temporal features of traffic data and then uses a Rc block to identify and quantify the contribution of features for each region. On the left part of Fig. 4,

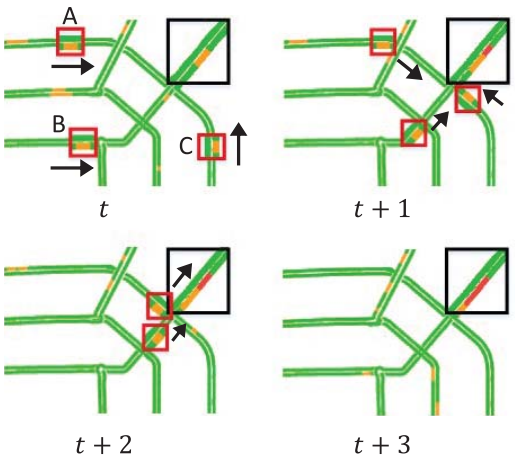


Fig. 6. An example of Traffic congestion propagation.

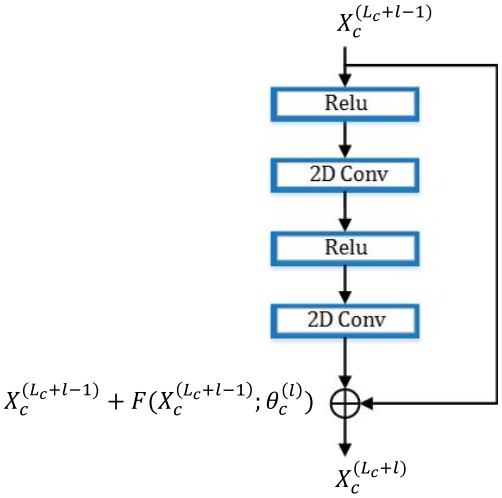


Fig. 7. The architecture of the 2D residual unit.

the weekly period component aims to describe the periodic and trend patterns in traffic data. Its input is a subsequence of ST raster data in the last few weeks, which has the same week attributes as the forecasting target. Define the subsequence as $X_w^{(0)} = \left[X_{t-d_w \cdot p_{week}}, X_{t-(d_w-1)\cdot p_{week}}, \ldots, X_{t-p_{week}}\right] \in \mathbb{R}^{C\times I\times J\times d_w}$, where the period p_{week} is fixed to one week and d_w is the length of the weekly period dependent sequence. The weekly period component uses 3D convolutions to capture temporal patterns and a Rc block to select informative features and to suppress useless features for each region. On the top of ST-3DNet, the outputs of the two components are merged as X_f based on parameter matrices, which are learnable to reflect the difference of the contributions of the two temporal properties in space. Finally, X_f is followed by an activation function.

1) Structure of the Closeness Component: The closeness component consists of three sub-components, including 3D convolutions, 2D residual units and a recalibration block, as shown in Fig. 4.

a) 3D Convolution: In spatio-temporal traffic data, the observations gained at nearby locations and adjacent time intervals are not independent but are correlated with each other. Taking traffic congestion condition for example, we can often observe that a traffic congestion usually covers a continuous area. This kind of purely spatial correlation can be captured by 2D Convolutions [4]. However, as time goes on, the congestion may propagate to other distant regions. As shown in Fig. 6, during the morning rush hours, many people were going to the core commercial area located in the northeast of the city. Suppose at time t, three congestions occurred respectively at link A, B and C, and most of the vehicles on these three links were driving towards the northeast, then the congestions would also propagate towards the northeast. In this case, only modeling the spatial features is far from enough. In ST traffic data analysis, it is critical to capture the motion information encoded in multiple contiguous snapshots. Fortunately, it has been proved that 3D Convolutions performed in video analysis problems can effectively capture the motion information [32] from both spatial and temporal dimensions.

Motivated by this, in the closeness component we first stack L_c 3D convolution layers (i.e., 3D Conv shown in Fig. 4). Formally,

$$X_c^{(l)} = f(W_c^{(l)} * X_c^{(l-1)} + b_c^{(l)}), \quad l = 1, \ldots, L_c \tag{1}$$

where * denotes the operation of 3D convolution. $X_c^{(l-1)} \in \mathbb{R}^{C_{l-1}\times I\times J\times T_{l-1}}$ is the input of the l^{th} 3D convolution layer and C_{l-1} is the number of the channels. When $l = 1$, $C_0 = C$ and $T_0 = d_c$. $W_c^{(l)}$ and $b_c^{(l)}$ are parameters and f is an activation function. By convolving 3D filters to the cube formed by multiple contiguous ST raster data, the feature maps in the 3D convolution layers are connected to multiple contiguous raster data in the previous layer, thereby to extract the motion information.

b) Residual Unit: After 3D convolution operations, temporal information have been fully explored. So afterwards, we continue to perform 2D convolutions to further explore spatial information. One convolution layer can only capture local spatial correlation, so multiple consecutive convolution layers are required to capture distant spatial dependencies. However, it is more difficult to train deeper neural networks. To overcome this problem and make deep neural networks more easily to be optimized, the residual learning proposed by He *et al.* [34] is used in our model. In our work, the residual mapping consists of the combination of activation and 2D convolution twice, as shown in Fig. 7. The residual mapping is denoted as $\mathcal{F}$. And we stack L_r residual units on the last 3D convolution layer. Formally,

$$X_c^{(L_c+l)} = X_c^{(L_c+l-1)} + \mathcal{F}(X_c^{(L_c+l-1)}; \theta_c^{(l)}), \quad l = 1, \ldots, L_r \tag{2}$$

where $\theta_c^{(l)}$ is the set of all the learnable parameters in the l^{th} residual unit. $X_c^{(L_c+l-1)} \in \mathbb{R}^{C'_{L_c+l-1}\times I\times J}$ is the input to the l^{th} residual unit and C'_{L_c+l-1} is the number of the channels. When $l = 1$, in order to perform 2D convolution on $X_c^{(L_c)} \in \mathbb{R}^{C_{L_c}\times I\times J\times T_{L_c}}$, $X_c^{(L_c)}$ is reshaped to $X_c^{(L_c)} \in \mathbb{R}^{C'_{L_c}\times I\times J}$, where $C'_{L_c} = C_{L_c} T_{L_c}$.

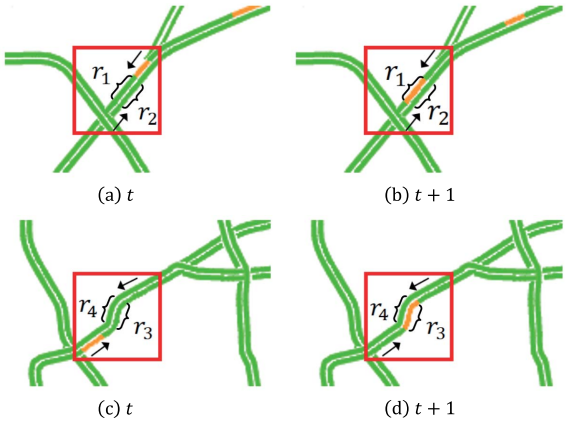


Fig. 8. Different effects of the same feature on different road links.

As our task is to predict future value of every grid in ST traffic raster data, ST-3DNet does not use any subsampling and pooling operations in the spatial dimension. Because these operations will reduce the tensor size and make networks less sensitive to the spatial features [35].

c) Recalibration Block: It should be noticed that the contributions of the neighboring regions on prediction vary spatially across the whole space [5]. If we fail to take the heterogeneity into account, modeling the correlations of traffic data will be insufficient.

We take the traffic condition forecasting problem in Beijing for example. As shown in Fig. 8, the direction of link r_1 is from northeast to southwest while the direction of r_2 is opposite. We use a box in Fig. 8(a) to abstractly represent a convolution filter, detecting there is a congestion in the northeast corner of the local receptive field. Since vehicles on r_1 are driving towards southwest and the congested link is directly connected to r_1, the feature captured by the filter plays an important role in predicting the traffic condition of r_1 at time $t+1$, as show in Fig. 8(b); compared to this, vehicles on r_2 are driving towards northeast and r_2 is not connected to the congested region, so the feature that a congestion taking place in the northeast corner has less impact on r_2 at time $t+1$, as shown in Fig. 8(b). Another filter shown in Fig. 8(c) captures the feature that there is a congestion in the southwest corner of the local receptive field. Because the direction along this congested region is from southwest to northeast and this region is directly connected to r_3, the feature have much more influence on r_3 than r_4. As a result, a detected feature takes various effects on different links' future condition, i.e., the extent of the contribution of same correlation is not invariable in space.

Therefore, identifying and quantifying the varying extent to the roles of features in space is essential for accurately modeling spatial and temporal correlations. Consequently, we present a "Recalibration" (Rc) block, which explores and automatically quantifies the extent of contributions of channel-wise features for each region, in order to improve our model's capacity. Moreover, in the field of computer vision, Hu *et al.* [36] have also proved that the representational power of a neural network can be boosted in modeling channel-wise features. Fig. 9 shows the architecture of Rc block. We stack a Rc block on the last residual unit $\text{ResUnit}^{(\text{L}_\text{r})}$ (as shown in the closeness component in Fig. 4) to perform feature recalibration as,

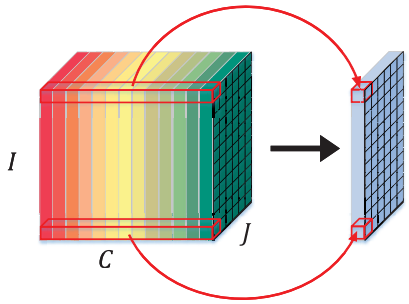


Fig. 9. A recalibration block.

$$X_c = \sum_{k=1}^{C'_{L_c+L_r}} w_c^k \circ x_c^k \tag{3}$$

where the input $X_c^{(L_c+L_r)} = [x_c^1, \ldots, x_c^{C'_{L_c+L_r}}] \in \mathbb{R}^{C'_{L_c+L_r} \times I \times J}$, $x_c^k \in \mathbb{R}^{I \times J}$ and $W_c = [w_c^1, \ldots, w_c^{C'_{L_c+L_r}}]$, $w_c^k \in \mathbb{R}^{I \times J}$, and $\circ$ is the element-wise multiplication. We use the learnable parameter W_c to quantify the extent to the roles of channel-wise features in space. Therefore, through using a Rc block, the difference of the contributions of features in space is well considered. For grid (i, j), its channel-wise features' contributions $[(w_c^1)_{i,j}, \ldots, (w_c^{C'_{L_c+L_r}})_{i,j}]$ are unique and learned based on the historical data.

2) Structure of the Weekly Period Component: Traffic data usually shows obvious periods and trends due to the regular daily routine of people. In Fig. 5, an obvious weekly periodicity can be observed. In addition, as season changes, traffic data also changes according to some trends. For example, in a year from spring to winter, when the sun rises later, people's daily travel also becomes later, which inevitably has impacts on traffic. According to this, 3D convolutions are employed in the weekly period component to learn the periodic and trend patterns of traffic data. Formally,

$$X_w^{(l)} = f(W_w^{(l)} * X_w^{(l-1)} + b_w^l), \quad l = 1, \ldots, L_w \tag{4}$$

where * denotes the operation of 3D convolution. $X_w^{(l-1)} \in \mathbb{R}^{C_{l-1} \times I \times J \times T_{l-1}}$ is the input of the l^{th} 3D convolution layer and C_{l-1} is the number of the channels. When $l = 1$, $C_0 = C$ and $T_0 = d_w$. $W_w^{(l)}$ and $b_w^{(l)}$ are parameters and f is an activation function. Here 3D convolutions are employed to capture weekly periods and trend patterns along the temporal dimension, so the filter size of 3D convolutions along the temporal dimension is set to a positive number larger than 1, while the filter size along the other two dimensions are set to 1. Then, we reshape the final output of stacked 3D convolution layers $X_w^{(L_w)} \in \mathbb{R}^{C_{L_w} \times I \times J \times T_{L_w}}$ to $X_w^{(L_w)} \in \mathbb{R}^{C'_{L_w} \times I \times J}$, where $C'_{L_w} = C_{L_w} T_{L_w}$. Afterwards, we use a recalibration block (i.e., Rc in the weekly period component shown in Fig. 4) again to assign different weights to different features for

different regions:

$$X_w = \sum_{k=1}^{C'_{L_w}} w_w^k \circ x_w^k \tag{5}$$

where the input $X_w^{(L_w)} = [x_w^1, \ldots, x_w^{C'_{L_w}}]$, $x_w^k \in \mathbb{R}^{I \times J}$ and $W_w = [w_w^1, \ldots, w_w^{C'_{L_w}}]$, $w_w^k \in \mathbb{R}^{I \times J}$. W_w are learnable parameters to adjust the degrees affected by different long-term temporal correlations in different locations.

3) Fusion: In this subsection, we discuss how to fuse the two components mentioned above. As shown in Fig. 4, the outputs of the closeness component and the weekly period component are denoted as X_c and X_w respectively. It should be noted that, the extent to the roles of the closeness component and the weekly period component is not globally same in space. For some regions long-term patterns are quite obvious, while for other regions the recent and sudden impacts are more important. So when fusing the two outputs, the extent to the contribution of the two kinds of features should be learned from the historical data. Formally, the two outputs are fused as follows:

$$X_f = W_{fc} \circ X_c + W_{fw} \circ X_w \tag{6}$$

where $\circ$ is the element-wise multiplication, W_{fc} and W_{fw} are learnable parameters that reflect the degrees of the closeness influence and the weekly period influence on the predicted target.

After merging the two outputs, we adopt a final activation layer at the end of our model, formally:

$$\widehat{X}_t = f(X_f) \tag{7}$$

where f is an activation function, e.g., the rectifier function.

4) Loss Function: Our ST-3DNet model is trained by minimizing the loss function, which is defined as the mean squared error (MSE) between the true traffic raster value and the predicted value:

$$L_\theta = \|X_t - \widehat{X}_t\|_2^2 \tag{8}$$

where θ are learnable parameters in ST-3DNet.

5) Training Algorithm: Algorithm 1 demonstrates the training process of ST-3DNet, where Γ is the training set, $X_c^{(0)}$ is the input of the closeness component and $X_w^{(0)}$ is the input of the weekly period component. Firstly, the training instances are constructed from the original data. Then forward propagation and back propagation are repeated applied to train the model. Adam algorithm [37] is chosen for optimization.

IV. Experiments

In this section, to evaluate the performance of ST-3DNet, we use two types of traffic raster data in our experiments. One is about traffic congestion conditions and the other is about crowd flows.

Algorithm 1 ST-3DNet Training Algorithm

Input: The historical traffic raster sequence: $\{X_t | t = 1, 2, \ldots, n\}$; the lengths of closeness and weekly period sequences: d_c, d_w; the time interval between the last observed time interval and the predicted time interval Δt.
Output: Learned ST-3DNet model.

$\varnothing \rightarrow \Gamma$
// construct training samples
for all available time interval $t (0 \leq t \leq n)$ **do**
  $X_c^{(0)} = [X_{t-d_c}, X_{t-(d_c-1)}, ..., X_{t-1}]$
  $X_w^{(0)} = [X_{t-d_w \cdot p_{week}}, X_{t-(d_w-1)\cdot p_{week}}, ..., X_{t-p_{week}}]$
  //$X_{t+\Delta t}$ is the target at time interval $t + \Delta t$
  put a sample $(\{X_c^{(0)}, X_w^{(0)}, X_{t+\Delta t}\})$ into Γ
end for
// train the model
initialize all learnable parameters θ in ST-3DNet
repeat
  randomly select a batch of samples Γ_b from Γ
  find θ by minimizing the objective L_θ with Γ_b
until stopping criteria is met
output the learned ST-3DNet model

A. Settings

We implement our ST-3DNet based on Keras[1] which uses Tensorflow[2] as its backend engine.

1) Dataset: Two types of traffic raster data are used in our experiments, including traffic congestion conditions and crowd flows.

a) Citywide Traffic Congestion Condition Data: TrafficBJ is a set of citywide traffic condition data on the road network in Beijing, which is scratched from a public website Baidu Maps.[3] It is collected from May 14th to December 5th in 2017, with an updating frequency of six minutes. In practice, from 11 p.m. to 6 a.m. when most citizens are sleeping, traffic condition is generally smooth. So in this study, we only focus on the time period from 6 a.m. to 11 p.m. when traffic congestion often occurs and reliable traffic predictions are highly desired for travelers. In TrafficBJ, the road network is located within the Fifth Ring Road in Beijing and consists of 904 road links. The links on the road network are partitioned according to the lengths and intersections. The lengths of the links may be not the same, but are all in around [500,1000] meters. Through necessary preprocessing steps (see Appendix), the traffic conditions of all the 904 road links are calculated and arranged into raster data with size 53×42, according to their actual geographical locations. Here, we term the raster data as traffic condition snapshots. In a traffic condition snapshot, there is only one channel, so $C = 1$ and for a grid (i, j), $(X_t)_{0,i,j} = 0$ means no link passes this region. Otherwise, $(X_t)_{0,i,j} \in [1, 4]$ represents the average traffic condition value of the road link located in grid (i, j), where 1 means the traffic condition is smooth and 4 means severely crowded. The statistics of the dataset is described

[1] https://keras.io/
[2] https://www.tensorflow.org/
[3] https://map.baidu.com/

TABLE I
STATISTICS OF THE TRAFFIC CONDITION DATASET

Dataset	TrafficBJ
Location	In Beijing, within the Fifth Ring Road
Number of road links	904
Time span	14th May. 2017 - 5th Dec. 2017
Time interval	6 minutes
Raster size	(32, 32)
Number of available time intervals	36,040

in Table I. Finally, we obtain 36,040 available traffic condition snapshots and take the last 10% data as our test set and the rest as the training set.

b) Citywide Crowd Flow Data: TaxiBJ and BikeNYC are both citywide crowd flows datasets published in [21] and [38], as shown in Table II. Similarly, a city is divided into an $I \times J$ grid map according to the longitude and latitude. Then crowd flows data are represented in the format of ST traffic raster data in which a grid denotes a region and the values of a grid have two channels, respectively denoting the crowd inflow and outflow. TaxiBJ data are crowd flows obtained based on the taxicab GPS data in Beijing from four time intervals: 1st Jul. 2013 to 30th Oct. 2013, 1st Mar. 2014 to 30th Jun. 2014, 1st Mar. 2015 to 30th Jun. 2015, and 1st Nov. 2015 to 10th Apr. 2016. The data from the last four weeks are taken as the test set and the rest are taken as the training set. BikeNYC data are crowd flows obtained based on the trip data taken from the NYC Bike system in 2014, in which the last 10 days are regarded as test data and the remains as training data.

2) Baselines: We compare our ST-3DNet with the following 6 baseline methods:

- **HA**: The predicted value is the average of the recent historical traffic data at the corresponding time.
- **ARIMA**: Autoregressive Integrated Moving Average is a classical method in time series analysis [39].
- **LSTM**: Long short-term memory network (LSTM) [22], a special kind of RNN. A LSTM unit consists of a cell, an input gate, an output gate and a forget gate.
- **GRU**: Gated-recurrent-unit network (GRU) [23], a special kind of RNN.
- **ConvLSTM**: Convolution LSTM [25] is a kind of LSTM which is good at capturing spatio-temporal features.
- **ST-ResNet**: A deep neural network based prediction model for spatio-temporal data proposed in [21] and [38]. It shows state-of-the-art performance on the two crowd flow prediction tasks.

All of these baselines have their own characteristics. In general, when HA, ARIMA, LSTM and GRU are applied to make predictions, their inputs are usually from one entity. While other methods, i.e., ConvLSTM, ST-ResNet and our ST-3DNet can simultaneously process information from related entities, because convolutions in these models can deal with grid-based data. More specially, HA merely captures periodic patterns of traffic data but ignores recent temporal correlations while ARIMA is just the opposite. LSTM and GRU only model temporal features and fail to take spatial information into account. By contrast, ConvLSTM and ST-ResNet can capture both the temporal and spatial correlations. However, ConvLSTM also does not explicitly model periodic and trend information. Besides, ST-ResNet treats traffic data snapshots in adjacent time intervals as different channels, so its ability to describe temporal dependencies among adjacent data is limited. Compared with these baseline methods, our ST-3DNet can efficiently capture recent spatio-temporal correlation as well as periods and trends.

Root mean square error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE) are used as the evaluation metrics to evaluate the models.

B. Comparison and Analysis of Results

1) Traffic Congestion Condition Prediction: In this experiment, 3D convolution layers use 32 filters with size $3 \times 3 \times 3$, and 2D convolution layers in residual units use 32 filters with size 3×3. All the activation functions in the model are ReLU. The length of the closeness dependent sequence d_c is set to 31, which means that the traffic conditions within the last 180 minutes are used for forecasting. The length of the weekly period dependent sequence d_p is set to 4, which means that the traffic conditions within the last 4 weeks are considered for prediction. The input data is scaled into range $[-1, 1]$. In the experiment, we perform two versions of ST-3DNet, i.e., ST-3DNet-C only containing the closeness component, and ST-3DNet-CT containing both the closeness component and the weekly period component.

Fig. 10 shows the results of our model and other six baselines on short-term (6–36 min) traffic forecasting. It can be observed that ST-3DNets consistently achieve better performance than the baseline methods in terms of all evaluation metrics. More specifically, in average our ST-3DNets have relatively 9.86% lower RMSE, 7.35% lower MAE and 11.67% lower MAPE than the three temporal models, including ARIMA, LSTM and GRU. The reason is that these models only capture the temporal dependencies of a region and fail to take the spatial dependencies among regions into account. Then we compare ST-3DNets against previous state-of-the-art spatio-temporal models, including ConvLSTM and ST-ResNet. Although ConvLSTM can simultaneously capture spatial and temporal information, it is worse than ST-ResNet and our ST-3DNets. This is due to the complicated structure of ConvLSTM, so it can only stack a few layers. Thus, ConvLSTM only considers the nearby spatial information and recent temporal information. ST-ResNet simply treats traffic raster data from adjacent time intervals as different channels, so after the first convolution operation, the temporal correlation in data is missing. Besides, the top layer in each component of ST-ResNet is a convolutional layer, indicating that ST-ResNet regards all channel-wise features have the same influence on traffic forecasting of different regions. This is obviously not corresponding with the fact. Compared to them, our ST-3DNets employ 3D convolutions to maintain the information in the temporal dimension until the temporal features and the motion information are fully explored. Besides, ST-3DNets append a recalibration block at the end of each component to adaptively recalibrate different features for different regions.

TABLE II
STATISTICS OF THE CROWD FLOW DATASET

Dataset	TaxiBJ	BikeNYC
Location	Beijing	New York
Time span	1st Jul. 2013 - 30th Oct. 2013; 1st Mar. 2014 - 30th Jun. 2014 1st Mar. 2015 - 30th Jun. 2015; 1st Nov. 2016 - 10th Apr. 2016	1st Apr. 2014 - 30th Sept. 2014
Time interval	30 minutes	1 hour
Raster size	(32, 32)	(16, 8)
Number of available time intervals	22,459	4392

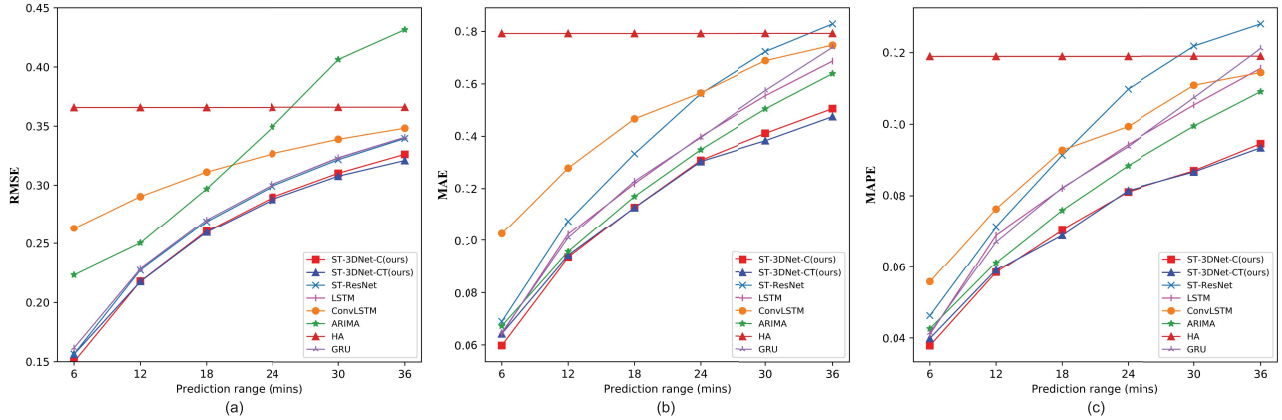


Fig. 10. Comparison of ST-3DNets with other baselines on the short-term prediction.

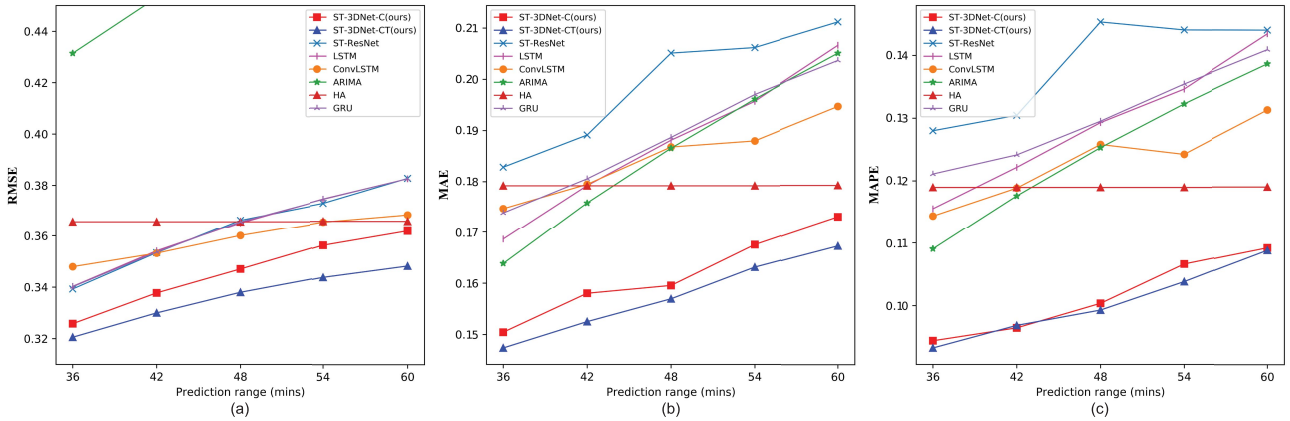


Fig. 11. Comparison of ST-3DNets with other baselines on the long-term prediction.

Therefore, ST-3DNets are more sophisticated than the two spatio-temporal models.

Moreover, in short-term traffic forecasting, ST-3DNet-CT has extremely slight advantages over ST-3DNet-C, although it explicitly models the periodicity and the trend in traffic data. This may be because the nearby future traffic conditions are heavily depended on the just past historical traffic conditions, thus the closeness component almost captures all the useful information and the extra weekly period component does not bring a significant performance improvement.

Fig. 11 shows the result comparison on long-term traffic predictions. In long-term (36–60 min) traffic forecasting, the performances of the temporal models drop dramatically, which makes the advantage of HA obvious. Such phenomenon demonstrates that the weekly information, i.e., the periodicity and the trend of traffic data is quite essential in long-term traffic forecasting. Although LSTM and GRU can capture long-term dependencies in theory, in practice, it is difficult for them to be well trained when the input sequence is too long and redundant. So their long-term traffic forecasting performance is poor. In fact, only a few previous key snapshots have great influence on the prediction. Consequently, in ST-3DNet we use an independent component to explicitly model the weekly periodic and trend patterns based on the corresponding historical traffic raster data. Besides, the performance of ConvLSTM drops relatively slower than other baseline methods, indicating both the spatial and temporal correlations should be well considered in the long-term prediction. Among all

TABLE III
COMPARISON OF ST-3DNETS WITH OTHER BASELINES ON TAXIBJ AND BIKENYC

Model	TaxiBJ			BikeNYC		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE
HA	46.41	23.70	0.45	10.73	5.84	0.69
ARIMA	22.80	16.31	0.42	9.98	6.02	0.70
LSTM	20.79	11.98	0.33	9.66	5.94	0.59
GRU	22.46	12.97	0.31	9.16	5.79	0.61
ConvLSTM	19.83	11.22	0.34	7.07	4.96	0.54
ST-ResNet	17.48	10.06	0.26	6.70	3.91	0.44
ST-ResNet-Ext	17.20	9.67	0.24	6.30	3.70	0.42
ST-3DNet (ours)	**16.21**	**9.36**	**0.23**	**5.80**	**3.43**	**0.42**
ST-3DNet-Ext (ours)	**16.09**	**9.33**	**0.23**	**5.81**	**3.43**	**0.43**

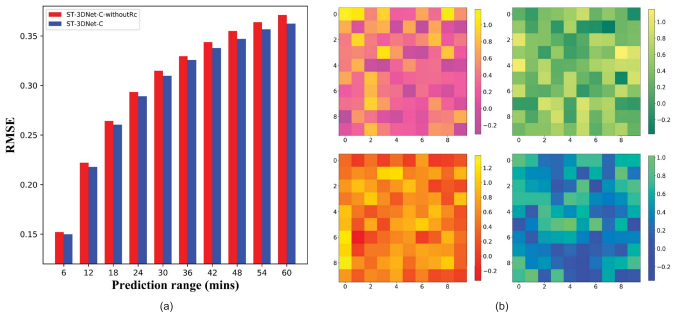


Fig. 12. Effect of the recalibration block.

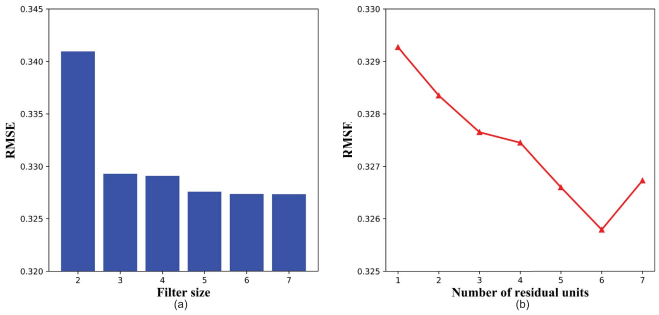


Fig. 13. Effect of different network configurations.

the methods, our ST-3DNets show the best generalization performance in terms of all the three valuation metrics.

In addition, it is noticed that in long-term traffic forecasting, the gap between ST-3DNet-C and ST-3DNet-CT becomes wider. That is because with the increasing of the prediction time interval, contributions of recent historical traffic conditions for forecasting become less. By contrast, the periodic and trend patterns in traffic data is more significant and should be paid more attention.

a) Effect of the Recalibration Block: To show the effect of the Rc block, we conduct a comparative study. ST-3DNet-C-withoutRc is a degraded version of ST-3DNet-C, which removes the Rc block. Fig. 12 (a) shows the results of ST-3DNet-C-withoutRc and ST-3DNet-C on traffic condition forecasting. With the prediction interval changing from 6 mins to 60 mins, due to lack of modeling the differences of links, ST-3DNet-C-withoutRc always performs worse than ST-3DNet-C. By contrast, thanks to the Rc block which can automatically recalibrate the importance of channel-wise features for different regions, ST-3DNet-C achieves better forecasting performance. In order to further investigate the role of the Rc block intuitively, we perform a case study: picking out a partial region with 10×10 grids from TrafficBJ, and showing their weight matrix in the Rc layer. We take ST-3DNet-C on the 36-min prediction as an example and randomly select four channels (features). As shown in the Fig. 12 (b), in each weight matrix, the value of (i, j) represents the contribution of the corresponding feature on the grid (i, j). We can observe that a same feature has various contributions for each grid, and different features have different contributions for a same grid. Thus, the Rc layer automatically learn the contributions of the channel-wise features based on the historical data.

b) Effect of Different Network Configurations: To verify how the filter size impacts the performance of ST-3DNet, we set the number of residual unit to 1 and change the size of the convolution filter from 2×2 to 7×7, as shown in Fig. 13 (a). Taking ST-3DNet-C on the 36-min prediction as an example, when the filter size increases from 2 to 3, the RMSE drops dramatically and when the filter size increases from 3 to 7, the RMSE does not change a lot. Since large filter size will cost much more time to train the model, we set it to 3. Fig. 13 (b) shows the impact of network depth. Taking ST-3DNet-C on the 36-min prediction for example, when the network becomes deeper i.e., the number of residual units increases, the RMSE of ST-3DNet-C decreases, demonstrating that appropriately increasing network depth can improve model performance.

2) Crowd Flows Prediction: In this experiment, the first 3D convolution layer uses 64 filters with size $l \times 3 \times 3$, where l equals to the size of the input data in the temporal dimension. The other 3D convolution layers use 64 filters with size $3 \times 3 \times 3$, and the 2D convolution layers in residual units use 64 filters with size 3×3. All the activation functions in the model are ReLU. We let the length of the closeness dependent sequence $d_c \in \{3, 4, 5, 6, 7, 8, 9\}$ and the length of the weekly period depend sequence $d_p \in \{2, 3, 4\}$. The input data is scaled into range $[-1, 1]$. We also consider external features according to [21]. We present the comparison against other 6 baselines on the TaxiBJ and BikeNYC datasets, as shown in Table III.

The experimental results on TaxiBJ shows that without taking external factors into account, our ST-3DNet reduces the RMSE to 16.21, MAE to 9.36 and MAPE to 0.23, which has already been better than the previous best model ST-ResNet. Further considering the external features, ST-3DNet-Ext achieves slightly better results than ST-3DNet. When applied to another crowd flows data BikeNYC, ST-3DNet still shows the best generalization performance in terms of all evaluation metrics, which verifies again our idea that 3D convolutions can efficiently capture the spatio-temporal features in traffic raster data and the recalibration unit which recalibrates the features for each region in the raster area can further improve the capacity of the prediction model.

V. Conclusion

In this work, we propose a novel deep learning based spatio-temporal neural network (ST-3DNet) for predicting traffic raster data. ST-3DNet employs 3D convolutions to extract features from both spatial and temporal dimensions. Considering traffic data show obvious cyclical patterns and trends, ST-3DNet explicitly models the temporal properties of traffic data, including closeness and weekly period. In order to describe the heterogeneity of traffic data in space, a new Rc block is proposed to select and recalibrate features for every region. We evaluate our model on three real datasets in traffic domain and the experimental results show that our model achieves the best performance against other baselines.

It is worth noting, ST-3DNet is a general-purpose model, which is suitable for many traffic forecasting problems as long as the data can be represented in the form of spatio-temporal traffic raster data. So it can be widely applied in ITS to provide future reliable traffic guidance information and to help improving the safety and efficiency of ITS.

Appendix
Preprocessing on TrafficBJ Dataset

Here, some relevant concepts and detailed preprocessing on TrafficBJ dataset are given. Through preprocessing, the traffic conditions of all the road links are organized into raster data according to the actual spatial relations among the links. Suppose there are totally K road links on the road network, and x_t^k denotes the average traffic condition of link k at time t.

Definition 2 (Grid Map): A city is partitioned into a $M \times N$ grid-based map M_G according to the longitude and latitude, as shown in Fig. 14. Each grid represents a spatial region. For a grid (m, n) at the m-th row and the n-th column, $x_t^{(m,n)}$ denotes its traffic condition at time t. For each grid (m, n), $x_t^{(m,n)} = 0$ means no link passes this region, otherwise $x_t^{(m,n)} \in \{1, 2, 3, 4\}$ denotes the grade of the traffic condition in this region, where 1 to 4 respectively means that the traffic condition is smooth, slow, congested and severely congested. The traffic congestion level is prescribed by the web mapping service provider, such as Baidu Maps[4] and AutoNavi.[5]

[4]https://en.wikipedia.org/wiki/Baidu_Maps/
[5]https://en.wikipedia.org/wiki/AutoNavi

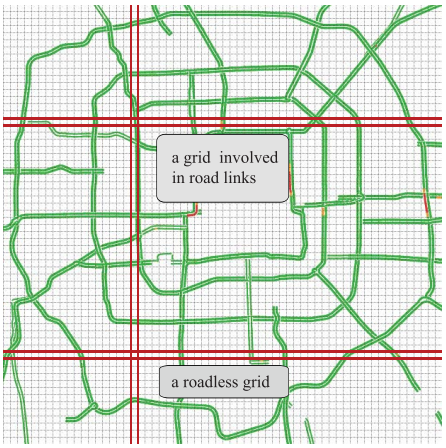


Fig. 14. The grid map of Beijing.

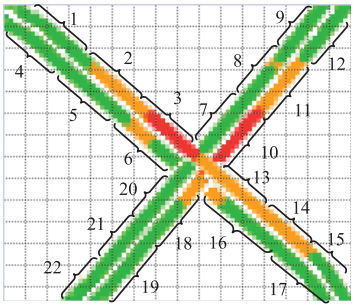


Fig. 15. Grids involved in road links.

Definition 3 (Average Traffic Condition): Assume $\phi_k = \{(m, n)\}_k$ is a set of grids involved in road link k. As shown in Fig. 15, each road link may cover some grids. We define the average traffic condition x_t^k on link k at time t as:

$$x_t^{(k)} = \frac{\sum_{(m,n)\in\phi_k} x_t^{(m,n)}}{|\phi_k|} \tag{9}$$

where $|\phi_k|$ is the number of grids involved in link k.

When m and n are set to small values, the partitioned grids in M_G become larger, so road areas and roadless areas likely appear in one grid. Thus, the calculated average traffic condition is coarse-grained. In order to get traffic information at a fine granularity, m and n should be set to relatively large values. In this way, there are more grids in M_G, which greatly increases the computing complexity of the subsequent operations. Besides, since road links only cover a little proportion of area in a city, most of the grid values in M_G are zero. But actually we are not very concerned about these roadless regions. Therefore, a citywide traffic condition compressed representation method is proposed to get rid of worthless information and to reduce the storage space. It rearranges road links in relatively small raster data and well retains the spatial topological relations among links.

Definition 4 (Link Map): We arrange all road links into a reduced raster structure M_L according to their actual locations and $M_L \in \mathbb{R}^{I\times J}$ is termed as Link Map. So we need to define a map function $f : M_G \to M_L$, where $(M_L)_{i,j} = \emptyset$ indicates $(M_L)_{i,j}$ is a roadless area, and $(M_L)_{i,j} = \phi_k$ indicates

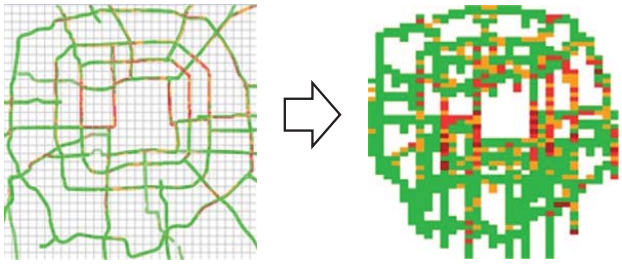


Fig. 16. An example of the transformation from grid map to link map.

that the grid set ϕ_k of link k is mapped into the location (i, j) in M_L.

A compression algorithm is proposed to convert $M_G \in \mathbb{R}^{M\times N}$ into $M_L \in \mathbb{R}^{I\times J}$ to reduce the storage space and data sparsity. As shown in Algorithm 2, the input of the algorithm includes the longitude and latitude of all links $\{(lng^k, lat^k)|k = 1, \ldots, K\}$, the grid map M_G and J, the number of columns of the link map M_L. Three operations, including division, sort and alignment are involved in the algorithm. In the division operation, $M_G \in \mathbb{R}^{M\times N}$ is divided to J equal longitudinal parts, so all the road links are divided into J subsets according to their longitude. Subsequently, we sort the links in each subset by their latitude. The max_number represents the maximum value of the subsets' sizes. In order to arrange all the links contained in the largest subset, the initial value of the row count in M_L is set to max_number. Finally, in last alignment operation, every link in each subset are arranged into a location according to its latitude. In the compression algorithm, J is a hyperparameter that needs to be set in advance, and a proper value of J is needed to ensure that the topological structure of links in M_L is very similar to that in the original M_G.

Fig. 16 gives a demonstration of Algorithm 2. Take TrafficBJ for example, which contains the traffic conditions of the area within the Fifth Ring Road in Beijing. The distance between the north and the south of the area is around 30 kilometers, so is the distance between the east and the west. Thus, when the area is divided into a 620×620 grid map, the height and width of each grid are around 48 meters, which is a proper size that just contains one trunk road link segment. Therefore, in the study case, when we partition the city, the hyperparameters M and N are set to 620. Afterwards, in the compression algorithm, the hyperparameter J is set to 42 by trial. As shown in the left part in Fig. 16, the original grid map M_G is almost a square, thus the corresponding compressed link map which retains the spatial information of the original grid map M_L should also be an approximated square with nearly equal height and width. By trial, when M_L is divided into 42 equal longitudinal subsets, we find that max_number of the subsets is 42, equal to J, which ensures that the link map M_L is also an approximated square. Then, according to Algorithm 2, we divide M_G into 42 equal parts along the latitude and then orderly arrange links of each subset into M_L according to their latitude. The algorithm finally rearranges all the road links into M_L of size 53×42, where the row count of M_L is automatically calculated according to the alignment operation. As shown in Fig. 16, compared to left grid map M_G, the spatial relations of the road links are well remained in the right link map M_L

Algorithm 2 Transformation From Grid Map to Link Map

Input: The longitude and latitude of all links $\{(lng^k, lat^k)|k = 1, \ldots, K\}$; grid map M_G; the number of columns J of the link map M_L.

Output: Link map M_L

// divide $\{(lng^k, lat^k)\}$ into J subsets $\{SubSet_j\}_{j=1}^{J}$
// divide M_G into J equal parts $\{P_1, P_2, \ldots, P_J\}$ according to the longitude
for each road link k **do**
  if $lng^k \geq P_j.lng^k$ and $lng^k < P_{j+1}.lng^k$ **then**
    $SubSet_j.add((lng^k, lat^k))$
    $lng\_index^k = j$
  end if
end for
// sort each subset
for each $SubSet_j$ **do**
  sort all (lng^k, lat^k) in $SubSet_j$ by lat^k
end for
$max\_number = \max_j |SubSet_j|$
$step = \frac{M}{max\_number}$
divide M_G into max_number equal parts $\{Q_1, Q_2, \ldots, Q_{max\_number}\}$ along the latitude
// alignment
$I = max\_number$
for each $SubSet_j$ **do**
  for each (lng^k, lat^k) in $SubSet_j$ **do**
    $lat\_index^k = \lfloor \frac{lat^k}{step} \rfloor$
    if $(M_L)_{j,lat\_index^k} = \emptyset$ **then**
      $(M_L)_{j,lat\_index^k} = \phi_k$
    else
      while $(M_L)_{j,lat\_index^k} \neq \emptyset$ **do**
        lat_index^k++
        if $lat\_index^k > I$ **then**
          $I = lat\_index^k$
        end if
      end while
      $(M_L)_{j,lat\_index^k} = \phi_k$
    end if
  end for
end for
return M_L

Definition 5 (Traffic Condition Snapshot): A traffic condition snapshot of the link map M_L at time t is denoted as X_t, where $X_t \in \mathbb{R}^{1\times I\times J}$ is a tensor. X_t is defined as follows:

$$(X_t)_{0,i,j} = \begin{cases} 0 & (M_L)_{i,j} = \emptyset \\ x_t^k & (M_L)_{i,j} = \phi_k \end{cases} \tag{10}$$

Hence, for a grid (i, j) in X_t, $(X_t)_{0,i,j} = 0$ means no link passes this region. Otherwise, $(X_t)_{0,i,j} \in [1, 4]$ represents the average traffic condition of the road link k located in (i, j) in M_L.

Therefore, after the preprocessing step described above, a series of traffic condition snapshots of size 53×42 are got. The traffic condition snapshots not only comprehensively reflect the traffic conditions of all the links on the road network in a relatively small-scale raster data, but also retain the spatial information of the original road network.

ACKNOWLEDGMENT

The authors sincerely thank three undergraduates Mingfei Jiang, Yan Lin and Xiaohui Liang in Beijing Jiaotong University for collecting and preprocessing the experimental datasets.

REFERENCES

[1] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011.

[2] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 187–197, May 2015.

[3] M. Fouladgar, M. Parchami, R. Elmasri, and A. Ghaderi, "Scalable deep traffic flow neural networks for urban traffic congestion prediction," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2251–2258.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[5] T. Cheng, J. Haworth, and J. Wang, "Spatio-temporal autocorrelation of road network data," *J. Geograph. Syst.*, vol. 14, no. 4, pp. 389–413, Oct. 2012.

[6] M. M. Hamed, H. R. Al-Masaeid, and Z. M. B. Said, "Short-term prediction of traffic volume in urban arterials," *J. Transp. Eng.*, vol. 121, no. 3, pp. 249–254, May 1995.

[7] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining Kohonen maps with ARIMA time series models to forecast traffic flow," *Transp. Res. C, Emerg. Technol.*, vol. 4, no. 5, pp. 307–318, 1996.

[8] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, Nov. 2003.

[9] N. G. Polson and V. O. Sokolov, "Deep learning for short-term traffic flow prediction," *Transp. Res. C, Emerg. Technol.*, vol. 79, pp. 1–17, Jun. 2017.

[10] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, Aug. 2004.

[11] T. Evgeniou, M. Pontil, and T. Poggio, "Regularization networks and support vector machines," *Adv. Comput. Math.*, vol. 13, no. 1, p. 1, Apr. 2000.

[12] B. S. Westgate, D. B. Woodard, D. S. Matteson, and S. G. Henderson, "Travel time estimation for ambulances using Bayesian data augmentation," *Ann. Appl. Statist.*, vol. 7, no. 2, pp. 1139–1161, 2013.

[13] O. Anacleto, C. Queen, and C. J. Albers, "Multivariate forecasting of road traffic flows in the presence of heteroscedasticity and measurement errors," *J. Roy. Stat. Soc., C*, vol. 62, no. 2, pp. 251–270, Mar. 2013.

[14] G. A. Davis and N. L. Nihan, "Nonparametric regression and short-term freeway traffic forecasting," *J. Transp. Eng.*, vol. 117, no. 2, pp. 178–188, Mar. 1991.

[15] H. Chang, Y. Lee, B. Yoon, and S. Baek, "Dynamic near-term traffic flow prediction: Systemoriented approach based on past experiences," *IET Intell. Transp. Syst.*, vol. 6, no. 3, pp. 292–305, Sep. 2012.

[16] M. G. Karlaftis and E. I. Vlahogianni, "Statistical methods versus neural networks in transportation research: Differences, similarities and some insights," *Transp. Res. C*, vol. 19, no. 3, pp. 387–399, Jun. 2011.

[17] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014.

[18] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[19] H.-F. Yang, T. S. Dillon, and Y.-P. P. Chen, "Optimized structure of the traffic flow forecasting model with a deep learning approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2371–2381, Oct. 2017.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

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

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proc. NIPS Deep Learn. Represent. Learn. Workshop*, Dec. 2014.

[24] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, "LSTM network: A deep learning approach for short-term traffic forecast," *IET Intell. Transp. Syst.*, vol. 11, no. 2, pp. 68–75, Mar. 2017.

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

[26] J. Wang, T. Cheng, B. G. Heydecker, and J. Howarth, "STARIMA for journey time prediction in London," in *Proc. 5th IMA Conf. Math. Transp.*, 2010.

[27] Y. Yue and A. G.-O. Yeh, "Spatiotemporal traffic-flow dependency and short-term traffic forecasting," *Environ. Planning B, Planning Des.*, vol. 35, no. 5, pp. 762–771, Oct. 2008.

[28] Q. Y. Ding, X. F. Wang, X. Y. Zhang, and Z. Q. Sun, "Forecasting traffic volume with space-time ARIMA model," *Adv. Mater. Res.*, vol. 156, pp. 979–983, 2011.

[29] X. Min, J. Hu, Q. Chen, T. Zhang, and Y. Zhang, "Short-term traffic flow forecasting of urban network based on dynamic STARIMA model," in *Proc. 12th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2009, pp. 1–6.

[30] X. Min, J. Hu, and Z. Zhang, "Urban traffic network modeling and short-term traffic flow forecasting based on GSTARIMA model," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2010, pp. 1535–1540.

[31] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.

[32] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[33] G. Atluri, A. Karpatne, and V. Kumar, "Spatio-temporal data mining: A survey of problems and methods," *ACM Comput. Surv.*, vol. 51, no. 4, Sep. 2018, Art. no. 83.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[35] I. Goodfellow, A. C. Courville, and Y. Bengio, *Deep Learning*, vol. 1. Cambridge, MA, USA: MIT Press, 2016,

[36] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7132–7141.

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, Dec. 2014, pp. 1–15.

[38] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, "Predicting citywide crowd flows using deep spatio-temporal residual networks," *Artif. Intell.*, vol. 259, pp. 147–166, Jun. 2018.

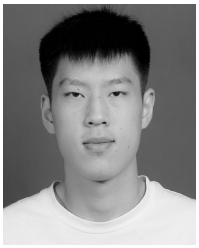
[39] G. E. P. Box and D. A. Pierce, "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models," *J. Amer. Statist. Assoc.*, vol. 65, no. 332, pp. 1509–1526, Apr. 1970.

Shengnan Guo received the B.S. degree in computer science from Beijing Jiaotong University, Beijing, China, in 2015, where she is currently pursuing the Ph.D. degree with the School of Computer and Information Technology.

Her research interests focus on traffic data mining and intelligent transportation technology.

Youfang Lin received the Ph.D. degree in computer science and technology from Beijing Jiaotong University, Beijing, China, in 2003.

He is currently a Professor with the School of Computer and Information Technology, Beijing Jiaotong University. His main fields of expertise and current research interests include big data technology, intelligent systems, complex networks, and traffic data mining.



Shijie Li is currently pursuing the bachelor's degree with the School of Computer and Information Technology, Beijing Jiaotong University.

His research interests focus on machine learning and data mining.

Zhaoming Chen is currently pursuing the bachelor's degree with the School of Computer and Information Technology, Beijing Jiaotong University.

His research interests focus on machine learning and data mining.

Huaiyu Wan received the Ph.D. degree in computer science and technology from Beijing Jiaotong University, Beijing, China, in 2012.

He is currently an Associate Professor with the School of Computer and Information Technology, Beijing Jiaotong University. His current research interests focus on traffic data mining and social network mining, and user behavior analysis.