Transportation Research Part C 115 (2020) 102639

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

A residual spatio-temporal architecture for travel demand forecasting

Ge Guo[a,*], Tianqi Zhang[b]

[a] State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, China
[b] Department of Automation, Dalian Maritime University, China

ARTICLE INFO

Keywords:
Demand forecasting
CNNs
CE-LSTMs
Residual connection

ABSTRACT

This paper proposes a deep architecture called residual spatio-temporal network (RSTN) for short-term travel demand forecasting. It comprises fully convolutional neural networks (FCNs) and a hybrid module consisting of an extended Conv-LSTM (CE-LSTM) that can achieve trade-off of convolutional operation and LSTM cells by tuning the hyperparameters of Conv-LSTM, convolutional neural networks (CNNs) and traditional LSTM. These modules are combined via residual connections to capture the spatial, temporal and extraneous dependencies of travel demand. The end-to-end trainable RSTN redefines the traditional prediction problem as a learning residual function with regard to the travel density in each time interval. Further more, a dynamic request vector (DRV)-based data representation scheme is presented, which catches the intrinsic characteristics and variation of the trend, to improve the performance of forecasting. Simulations with two real-word data sets show that the proposed method outperforms the existing forecasting algorithms, reducing the root mean square error (RMSE) by up to 17.87%.

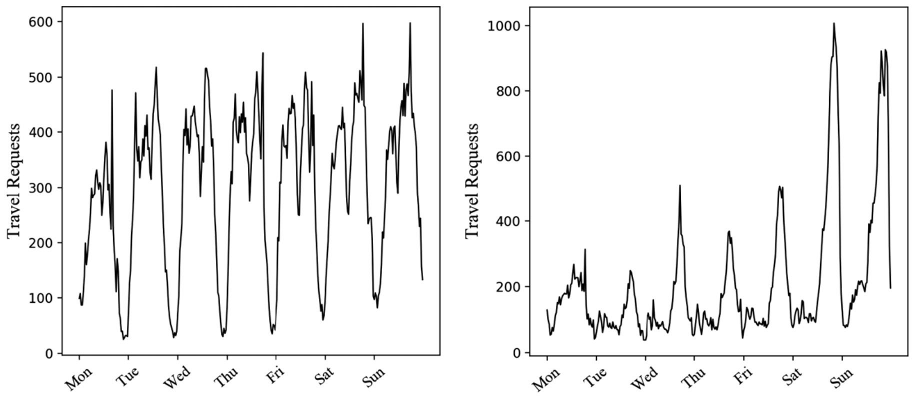
1. Introduction

Traffic forecasting (i.e., prediction of traffic flow and volume, taxi pick-ups and drop-offs, and travel demand) is one of the most fundamental issues in intelligent transportation systems. In particular, short-term travel demand forecasting is of essential importance for both traditional taxi services and the emerging mobility-on-demand systems (e.g., Uber, Lyft, DiDi ChuXing). In recent years, with massive traffic data easily available, which is enabled by technologies of mobile devices and wireless communications, travel demand forecasting has become an increasingly promising tool to balance vehicle supply and travel demand with low cost and high quality of service. The problem is very challenging due to high uncertainty of travel demand, complexity of road networks and spatial–temporal dependency between them. In addition, travel demand patterns in a city are heterogeneous, that is, the travel demands in different regions are quite different (see Fig. 1).

The existing methods for short-term travel demand prediction include time series analysis methods (Li, 2012; Moreira-Matias, 2013; Kaltenbrunner et al., 2010), machine learning methods (Li, 2011; Mukai and Yoden, 2012), and deep learning models (Ke, 2017; Xu, 2018). Most of these methods are implemented by dividing a region into small areas and counting the travel requests in a time interval as the historical demand (Yu et al., 2017; Yao et al., 2018; Ma and Chan, 2019; Liu, 2017; Zhu, 2017; Wu and Guo, 2018). Future travel demand is modeled as a function of the historical data, assuming certain spatial and temporal correlations between them and exogenous factors such as weather and temperature (Zhang et al., 2017b; Davis, 2016; Box and Pierce, 1970). The existing methods are limited in several aspects: (1). *Insufficient representation of travel demand*. Intuitively, using the number of travel

[*] Corresponding author.
E-mail address: geguo@yeah.net (G. Guo).

https://doi.org/10.1016/j.trc.2020.102639
Received 28 March 2019; Received in revised form 17 February 2020; Accepted 28 March 2020
Available online 23 April 2020



(a) (b)

Fig. 1. Travel requests of two different regions in a week. It is obvious that the two regions have distinct demand patterns. The request dynamics in (a) is a weak stationary process, while that in (b) is a non-stationary process, which is rather difficult to forecast.

requests cannot fully reflect the dynamics of the demand and the trend of variation. (2). *Naive modeling*. Modeling future demand as a function of historical data may miss some subtle information such as the trend and rate of variation in the demand. (3). *Insufficient spatio-temporal exploration*. Convolutional neural networks (CNNs), long-short-term memories (LSTMs) and their simple combination are not competent for fitting complicated traffic data. Therefore, a novel method of time-and-space excavation is needed.

The existing methods for short-term travel demand prediction include time series analysis methods (Li, 2012; Moreira-Matias, 2013; Kaltenbrunner et al., 2010), machine learning methods (Li, 2011; Mukai and Yoden, 2012), and deep learning models (Ke, 2017; Xu, 2018). Most of these methods are implemented by dividing a region into small areas and counting the travel requests in a time interval as the historical demand (Yu et al., 2017; Yao et al., 2018; Liu, 2017; Zhu, 2017). Future travel demand is modeled as a function of the historical data, assuming certain spatial and temporal correlations between them and exogenous factors such as weather and temperature (Zhang et al., 2017b; Davis, 2016; Box and Pierce, 1970). The existing methods are limited in several aspects: (1). *Insufficient representation of travel demand*. Intuitively, using the number of travel requests cannot fully reflect the dynamics of the demand and the trend of variation. (2). *Naive modeling*. Modeling future demand as a function of historical data may miss some subtle information such as the trend and rate of variation in the demand. (3). *Insufficient spatio-temporal exploration*. Convolutional neural networks (CNNs), long-short-term memories (LSTMs) and their simple combination are not competent for fitting complicated traffic data. Therefore, a novel method of time-and-space excavation is needed.

Based on the above observations, this paper proposes a novel deep architecture RSTN, which contains three parts: (1). A spatial correlation embedding (SCE) module, which is constructed by employing fully convolutional neural network (FCNs); (2). A spatio-temporal dependencies approximating (STDA) module, which is achieved by involving a hybrid module consisting of extended Conv-LSTMs (CE-LSTMs), LSTMs and CNNs; and (3). A residual connection (RC) module, which reformulates the prediction problem as a learning residual function with regard to the travel density in each time interval. The proposed approach leverages the strength of FCNs for spatial relation mapping, that of CE-LSTM for learning the complicated spatio-temporal dynamics effectively, and that of residual connection for capturing detailed information of changing in the travel demand. To further improve the forecasting performance, a novel representation of historical demand named dynamic request vector (DRV) is given to characterize the travel demand with more details to fully reflect the dynamics and trend of the demand, which is inspired by the method used in word representation in Zahran et al. (2015) and proportional integral differential (PID) control, a widelyused algorithm in control engineering. Empowered by DRV, the deep neural network model can then take advantages of the accumulated error information (I part) and the instantaneous variations (D part) in the prediction errors. Therefore, the new model is capable of eliminating the steady state prediction errors (thanks to the accumulated errors) and reducing future prediction errors more efficiently (fueled by the knowledge of instantaneous variations). Actually, the idea behind theexisting works is nothing but the proportional part (P part) in this paper, whose output is proportional to the difference between two consecutive time intervals. This part alone cannot efficiently and timely reduce prediction errors. In other words, involving the function of anticipating future prediction errors, DRV can provide RSTN with more information, and hence makes it superior in performance to the naive architecture (called nRSTN) with real-valued input.

The contributions of this paper are summarized as follows:

- A novel architecture named RSTN is constructed, which is an end-to-end trainable model that can adequately capture the exogenous dependencies and spatio-temporal correlations of travel demand.
- A residual connection scheme of the RSTN is introduced, yielding an easy-training procedure by reformulating the prediction problem as a learning residual function with regard to the travel density in each time interval.
- A vector representation of the historical travel demand is proposed. It enables more reasonable and logical representation of historical data and reflects more detailed dynamical information of travel requests.

The remainder of this paper is organized as follows. A literature review is given in Section 2. Section 3 gives the problem description and definitions of variables. Section 4 is the proposed architecture and methodology. Section 5 shows the experimental results, which is followed by the conclusions.

2. Literature review

Early research widely on traffic prediction are mostly based on time series analysis methods including, e.g., Holt-Winters (HW) models, fast fourier transform (FFT), ARMR models (auto regressive moving average), seasonal and trend decomposition methods (e.g., STL), and linear regression and exponential smoothing state space models (e.g., ETS and TBATS). To name some, Davis (2016), presented a multi-level time-series clustering technique to tackle the problem of taxi travel demand modeling. Kaltenbrunner et al. (2010), gave an ARMA model to predict the demand for bicycles in a city. An ARIMA-based approach was proposed in Li (2012) to predict the number of passengers in certain hot-spots. Taking the spatio-temporal interactions in a road network into account, Min and Wynter (2011), proposed an ARIMA method for traffic prediction.

Machine learning-based methods were also studied by researchers. For instance, Sun (2006), used a Bayesian network to model the traffic flow in road links. Kai et al. (2016), proposed a spatio-temporal clustering algorithm to provide demand hot-spots suggestion for taxi drivers.

Time series methods and machine learning-based algorithms can achieve good results, however, they are not satisfactory due to incompleteness of explanatory variables or lackness of characterizing capabilities. These shallow models are believed unsuitable for application in big data scenarios (Nguyen, 2018).

In recent years, deep learning methods (Jo et al., 2019; Yuan et al., 2019; Ren et al., 2019) increasingly gained research attention owing to great computing power and capability of characterizing big data. Ke (2017), presented a deep learning approach (named FCL-Net) to short-term passenger demand forecasting, which can address spatial, temporal, and exogenous dependencies simultaneously. Yu et al. (2017), addressed the traffic flow prediction problem by combining CNNs and LSTMs, treating the traffic state as an image. Yao et al. (2018), gave a deep learning architecture (Multi-View Spatial–Temporal Network, or DMVST-Net) for travel demand prediction, capturing the complex spatial–temporal correlations of travel demand via multiple views, i.e., temporal view, spatial view and semantic view. These and other deep models (Liu, 2017; Zhu, 2017) have shown much improved performance in exploring complicated dependencies in traffic data.

Short-term travel demand forecasting depends not only on the spatio-temporal properties of historical data but also on other explanatory variables. Here we propose a RSTN architecture that is advantageous in that: (i). the space–time dependences are more fully explored and a trade-off of the convolutional operation and LSTM cells is achieved, and (ii). travel demand forecasting problem is redefined considering more factors that may have effect, making it a standardized, easy-to-handle problem.

3. Problem description and preliminaries

We first give definitions of the short-term travel demand forecasting problem and description of variables. Table 1 lists some notations to be used.

3.1. Variable definition and data encoding

Partition of region and time. Like the existing methods, we uniformly divide the city into $I \times J$grids, each of which refers to an area. The time period needed for collecting information is defined as the time interval. Based on the partition of region and time, we have the following definitions of variables.

(1) **Demand intensity and dynamic request vector**

Demand intensity $d_t^{ij}$is the intensity of request at time interval t in grid (i, j). Unlike the existing methods where demand intensity is defined as the number of orders in a time interval within a grid, here we introduce an entirely new concept, termed as *dynamic request vector* (DRV), which not only captures the intensity of travel demand but also its trend of variation.

In constructing the DRV, we need to divide each time interval into subintervals and record the intensities, correspondingly. We then have DRV $d_t^{ij} = [d_1, d_2, \cdots, d_N]$, where N represents the number of subintervals, $d_k (k = 1, 2, \cdots, N)$ is the intensity in the kth subinterval. This design allows one to estimate the trend of variation of travel demand by "seeking" in the data.

Table 1
Notations of the variables.

Notation	Meaning
d_t^{ij}	demand intensity at area (i, j) in time interval t
τ_t^{ij}	average travel time rate at area (i, j) in time interval t
p^{ij}	POI of area (i, j)
h_t	attribute of time-of-day, level of demand density in time interval t
r_t	attribute of day-of-week in time interval t, which catches up the distinguished properties between weekdays and weekends
w_t	weather condition in time interval t

Then, the demand intensity in all areas of the city ($I \times J$ grids) at the tth time interval is defined as a 3-D tensor $D_t \in \mathbb{R}^{I \times J \times N}$(where $\mathbb{R}$ is the set of real numbers) with $(D_t)_{ij} = d_t^{ij}$.

(2) **Travel time rate**

Travel demand is complicatedly correlated with traffic congestion condition. The two are positively related to some extent, i.e., when the amount of travels increases, the traffic will become congested. However, in a congested area, people may choose transportation tools like the metro rather than the taxi concerned here.

In order to forecast the travel demand, this paper uses average travel time rate (i.e., average travel time per unit travel distance) (Zhang et al., 2017a) to denote traffic congestion condition, which is given by

$$\tau_t^{i,j} = \frac{1}{n}\sum_{s=1}^{n}\frac{t_{t,s}^{i,j}}{d_{t,s}^{i,j}} \tag{1}$$

where $t_{t,s}^{i,j}$ and $d_{t,s}^{i,j}$ are the time and distance of the sth travel request in area (i, j) during time slot t, n is the total number of requests in this time interval. We represent by $\Gamma_t \in \mathbb{R}^{I \times J}$ the average travel time in the whole city, where $(\Gamma_t)_{ij} = \tau_t^{i,j}$.

(3) **Point of interest (POI)**

POI represents the geographical attributes of regions, which are represented by the number of different categories of facilities contained therein. For example, raw POI data in area (i, j) is represented by $(n_1^{ij}, n_2^{ij}, n_3^{ij}, n_4^{ij})$ where n and k are the categories and numbers respectively. In this paper, to make it simple we redesign a new one by employing K-means algorithm then three cluster centers are obtained and used for the simple form

$$p^{ij} = \begin{cases} 0, & \text{if } m_t^{ij} \text{ belongs to } c_1 \\ 1, & \text{if } m_t^{ij} \text{ belongs to } c_2 \\ 2, & \text{if } m_t^{ij} \text{ belongs to } c_3 \end{cases} \tag{2}$$

where $m_t^{ij} = (m_{t1}^{ij}, m_{t2}^{ij}, m_{t3}^{ij}, m_{t4}^{ij})$ is a normalization form wherein $m_{tk}^{ij}\left(k = 1, 2, 3, 4\right) = \frac{n_k}{\sum_{i=1}^{4} n_i}$, $c_r\left(r = 1, 2, 3\right)$ is the rth cluster center. It should be mentioned that the data fed into K-means are also with normalization form. The notation of the whole city is P, where $(P)_{ij} = p^{ij}$.

(4) **Time-of-day, day-of-week and weather**

The definitions of time-of-day, day-of-week and weather are similar to those in Ke (2017). The attribute of time-of-day is given by

$$h_t = \begin{cases} 0, & \text{if } t \text{ belongs to peak hours} \\ 1, & \text{if } t \text{ belongs to off} - \text{peak hours} \\ 2, & \text{if } t \text{ belongs to sleep hours} \end{cases} \tag{3}$$

where peak, off-peak, sleep hours are distinguished based on the empirical demand intensity data.

The attribute of day-of-week is given by

$$r_t = \begin{cases} 0, & \text{if } t \text{ belongs to weekdays} \\ 1, & \text{if } t \text{ belongs to weekends} \end{cases} \tag{4}$$

Weather condition is represented by $w_t = (aw, at, ap)$, where aw, at, aprepresent the abstract of weather, temperature and PM2.5, respectively.

3.2. Problem formulation

Short-term travel demand forecasting problem aims to estimate the number of travel requests in each area of the city for a time $t + 1$, using the historical data collected until time interval t. The problem is defined as follows.

Problem 1. Given the historical and pre-known information $\{D_s, \Gamma_s | s = 1, 2,\cdots,t; h_s, r_s | s = 1, 2,\cdots,t + 1; aw_s, at_s, ap_s | s = 1, 2,\cdots,t; P\}$, predict D_{t+1}.

Assume that the change of demand in adjacent time intervals is related to the historical and pre-known information. Then, the problem becomes to find the following function:

$$D_{t+1} - D_t = \mathcal{F}(D_{1:t}, \Gamma_{1:t}, h_{1:t+1}, r_{1:t+1}, a_{1:t}, P) \tag{5}$$

where subscript 1: t represents the sequence from time 1 to time t.

Before proceeding further, it should be mentioned that we call variables with distinct time-and-space features as **spatio-temporal variables**, such as demand intensity and traffic congestion condition. Variables like time-of-day, day-of-week and weather, which only reflect temporal dependencies, are called **time-series variables**. Similarly, by **spatial variables**, we refer to variables that only show spatial dependencies, such as POI data.

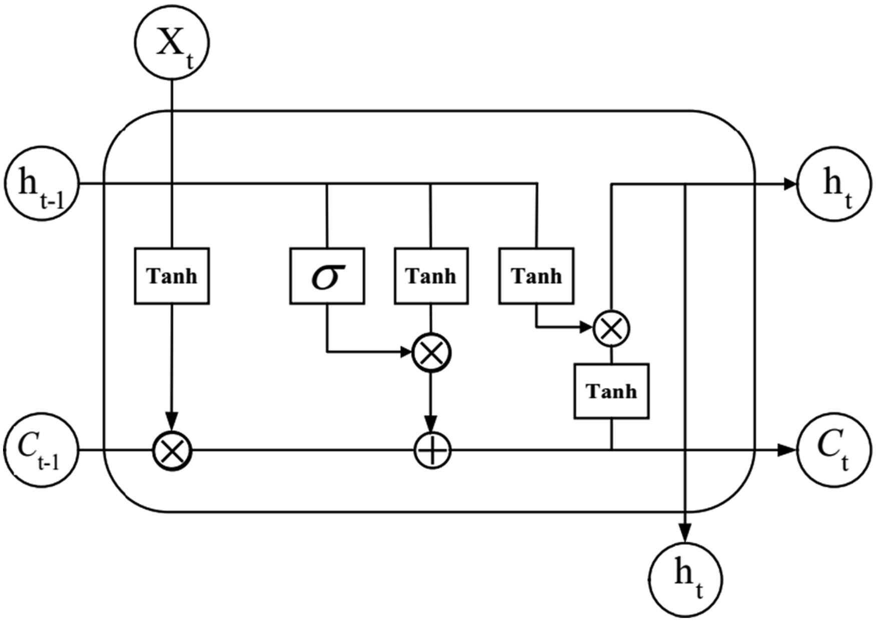


Fig. 2. A cell of LSTM, where σ represents *sigmoid* function in the form of $\sigma(x) = \frac{1}{1+e^{-x}}$, and it is also called a gate in the cell. There are three gates, named, respectively, as "forget gate", "input gate" and "output gate" from left to right. The first two gates determine what to add to the state of the cell C (which is the major feature of LSTM) and what to discard from C, the last gate decides the output and finally obtains the long-and-short-term memory. Symbol "⊗" stands for the Hadamard Product.

4. The RSTN and its training algorithm

4.1. LSTM and CE-LSTM

We use LSTM, a special recurrent neural network (RNN) structure, for processing travel demand sequence data. A LSTM can overcome the long-term memory weakness of RNNs due to vanishing gradients (Bengio, 1994). Fig. 2 shows a LSTM cell based on which the forecasting network is constructed (by stacking the cells). Like a standard RNN, each LSTM cell maps the input vector sequence X to a hidden vector sequence h by T iterations. Denote the forget gate, the input gate, the output gate, and the memory cell vectors by f_t, i_t, o_t, $c_t(t = 1, 2, ...T)$, respectively, which share the same dimension with h_t. We can have the following formulations,

$$f_t = \sigma(W_{xf}X_t + W_{hf}h_{t-1} + b_f) \tag{6}$$

$$i_t = \sigma(W_{xi}X_t + W_{hi}h_{t-1} + b_i) \tag{7}$$

$$o_t = \sigma(W_{xo}X_t + W_{ho}h_{t-1} + b_o) \tag{8}$$

$$g_t = tanh(W_{xc}X_t + W_{hc}h_{t-1} + b_c) \tag{9}$$

$$c_t = c_{t-1}{}^\circ f_t + i_t{}^\circ g_t \tag{10}$$

$$h_t = o_t{}^\circ tanh(c_t) \tag{11}$$

where, symbol $^\circ$ represents Hadamard product, which calculates the element-wise products of two vectors, matrices, or tensors with the same dimensions, σ and *tanh* are nonlinear activation functions given by

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{12}$$

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{13}$$

W_{xf}, W_{hf}, W_{xi}, W_{hi}, W_{xo}, W_{ho}, W_{xc}, W_{hc} are affine transformation parameter matrices, b_f, b_i, b_o, b_c represent the corresponding biases.

For the travel demand forecasting problem with complex spatio-temporal dependencies, one need a powerful LSTM to handle the space–time dependencies. The Conv-LSTM in Shi and Chen (2015) seems to be a good choice. However, the most important thing for Conv-LSTM is to tune the number of convolutional layers to achieve a tradeoff between the convolutional operation and LSTM cells. But this is usually a very tough task in implementation, and hence, makes Conv-LSTM either overfitted or under-fitted, since the two parts play different roles in spatio-temporal dependence exploration. Here, to address the over-fitting/under-fitting issue, we not only extend the structure using a series of convolution operations, but also give an5 effective tuning procedure of the hyper-parameters of ConvLSTMs to obtain optimal values of more variables like DRV, travel time rate and POI. The resulted structure CE-LSTM can explore the spatial dependencies of travel demand more effectively and more suited to **Problem 1** in Section III, and thus avoid over- or under-fitting, which makes implementation easier and more flexible. The dynamics of CE-LSTM are as follows:

$$\mathcal{F}_t = \sigma(C^n(\mathcal{X}_t;W_{xf_{1:n}}) + C^n(\mathcal{H}_t;W_{hf_{1:n}}) + b_f) \tag{14}$$

$$\mathcal{I}_t = \sigma(C^n(\mathcal{X}_t;W_{xi_{1:n}}) + C^n(\mathcal{H}_t;W_{hi_{1:n}}) + b_i) \tag{15}$$

$$\mathcal{O}_t = \sigma(C^n(\mathcal{X}_t;W_{xo_{1:n}}) + C^n(\mathcal{H}_t;W_{ho_{1:n}}) + b_o) \tag{16}$$

$$\mathcal{G}_t = tanh(C^n(\mathcal{X}_t;W_{xc_{1:n}}) + C^n(\mathcal{H}_t;W_{hc_{1:n}}) + b_c) \tag{17}$$

$$\mathcal{C}_t = \mathcal{C}_{t-1}{}^{\circ}\mathcal{F}_t + \mathcal{I}_t{}^{\circ}\mathcal{G}_t \tag{18}$$

$$\mathcal{H}_t = \mathcal{O}_t{}^{\circ}tanh(\mathcal{C}_t) \tag{19}$$

where function $C^n(\cdot)$ is the core of CE-LSTM, which embeds n CNN layers into the traditional LSTM, with n to be chosen to balance validity and adequacy of convolution operation. Specifically, the function is given by

$$C^n(\mathcal{X}_t;w_{1:n}) = w_n * \cdots * w_1 * \mathcal{X}_t \tag{20}$$

where '∗' stands for the convolutional operator, W_{xf}, W_{hf}, W_{xi}, W_{hi}, W_{xo}, W_{ho}, W_{xc} and W_{hc} are the convolutional kernels, b_f, b_i, b_o, b_c represent the corresponding biases. Note that $\mathcal{F}_t$, $\mathcal{I}_t$, $\mathcal{O}_t$, $\mathcal{C}_t$, $\mathcal{H}_t \in \mathbb{R}^{I\times J\times L}$are the improved forget gate, input gate, output gate, cell state and hidden state embedding the spatial dependencies.

4.2. Model description

In this subsection, we describe the details of the proposed RSTN architecture (as shown in Fig. 3), which is used for approximation of function $\mathcal{F}(\cdot)$ in (5). Recall that the RSTN consists of three parts for spatial correlation embedding (SCE), spatio-temporal dependencies approximating (STDA) and residual connection (RC), respcetively.

4.2.1. SCE (the FCN layers)

The demands in different regions of a city may have certain correlations. Intuitively, the travel patterns of adjacent or identical areas may have certain similariteis. Thus, the first step of RSTN is to explore the spatial correlations in the demand density D_t. In this study, we employ fully convolutional neural network (FCN) to construct the SCE part, which allows arbitrary input resolution and output volume feature maps with the same size of the input image and with embedded spatial dependencies. As shown in Fig. 3, demand density D_t is treated as a 3-D tensor at each time slot t (i.e., $D_t \in \mathbb{R}^{I\times J\times N}$, where I and J are the subdivisions in the longitude and latitude of the entire city, N is the length of the proposed DRV), the intermediate variable $V_t \in \mathbb{R}^{I\times J\times L}$ is the output of FCN layer, which is the transformation of D_t and called as the demand volume. Mathematically, we have that,

$$V_t = \mathcal{F}^{L}_{L_{l-1}} \cdots \mathcal{F}^{L_{l-z}}_{L_{l-z-1}} \cdots \mathcal{F}^{L_1}_{N}(D_t) \tag{21}$$

where function $\mathcal{F}^{N}_{M}(\cdot)$ performs a convolutional mapping from $\mathbb{R}^{I\times J\times M}$to $\mathbb{R}^{I\times J\times N}$, which is given by

$$\mathcal{F}^{L_1}_{N}(D_t) = W_{L_1} * D_t \tag{22}$$

with $W_{L_1} \in \mathbb{R}^{f_h\times f_w\times N\times L_1}$ being the convolutional kernel with height f_h and width f_w, N and L_1 being the input and output dimensions,

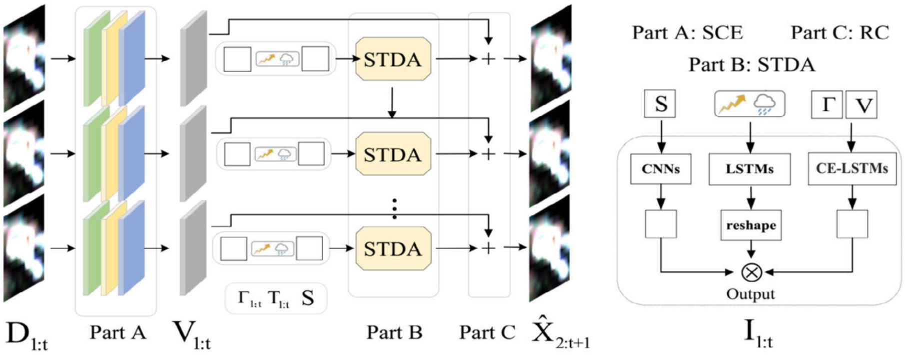


Fig. 3. The RSTN predicts short-term travel demand by taking into account three types of dependencies (spatial, temporal and exogenous) simultaneously. It is realized by treating the spatio-temporal demand density as a series of static images (represented by $D_{1:t}$, where $1{:}\ t$ represents the sequence from time 1 to t, and each demand density map is a frame) and incorporating other factors. In detail, the RSTN encodes the demand density into a density feature map, using variables ($T_{1:t}$, S) to fit the residual demand change with time.

respectively.

4.2.2. STDA (hybrid module)

Our model is constructed based on the assumption that the change of demand can be represented by a function entering the historical data, i.e., function $\mathscr{F}(\cdot)$ in (5). We name this part as spatio-temporal dependencies approximating (STDA), which performs a spatial and temporal dynamic mapping from historical information to the trend of travel demand changing. The rightmost diagram in Fig. 3 shows the structure of the STDA layer, wherein S denotes the spatial variable P defined in Sec. III and is processed via CNNs. To capture other extraneous temporal dependencies such as time-of-day, day-of-week and weather information (T), some stacked LSTM cells are used. Finally, historical traffic data such as demand volume (V) and travel time rate (Γ) are treated with the proposed CE-LSTMs. Mathematically, we have that,

$$S' = \sigma(\mathcal{F}_{L_{l-1}}^{1}\cdots\mathcal{F}_{L_{l-z-1}}^{L_{l-z}}\cdots\mathcal{F}_{N}^{L_1}(S)) \tag{23}$$

$$T_t' = \sigma(L_{l-1}^{H}\cdots L_{l-k-1}^{l-k}\cdots L_{G}^{H_1}(T_t, H_{t-1}, C_{t-1})) \tag{24}$$

$$V_t' = \mathcal{L}_{l-1}^{1}\cdots\mathcal{L}_{l-k-1}^{l-k}\cdots\mathcal{L}_{G}^{H_1}(V_t \oplus \Gamma_t, \mathcal{H}_{t-1}, C_{t-1}) \tag{25}$$

where $\mathcal{F}(\cdot)$ is the same as (22), function $L_M^N(\cdot)$ is the abstract of a traditional LSTM cell with dynamics in Eqs. (6)-(11), function $\mathcal{L}_M^N(\cdot)$ is the abstract of a CE-LSTM cell with dynamics in Eqs. (14)-(19), '$\oplus$' is the concatenating operator.

In our opinion, among the factors influencing the trend of changes in travel requests, the historical travel and traffic conditions play the dominating role. Thus, S' and T' are obtained through a *sigmoid* function and treated as two scales. So, the output of STDA module is

$$I_t = V_t' \times T_t' \times S'. \tag{26}$$

4.2.3. RC layer

The final prediction is made at this final layer based on the feature maps obtained from SCE part and STDA layer in a residual manner as described below:

$$\widehat{X}_{t+1} = \mathcal{P}(I_t + V_t) \tag{27}$$

where $\mathcal{P}(\cdot)$ is the forecasting function, which is based on a convolutional operator. The loss function is the mean square error between the estimated and the real demand intensities, namely,

$$L(\widehat{X}_{t+1}, D_{t+1}) = ||\widehat{X}_{t+1} - D_{t+1}||_2^2 \tag{28}$$

Algorithm 1 outlines the RSTN training process.

It should be mentioned that all the components except SCE are used to build function $\mathscr{F}(\cdot)$ by introducing residual connection. In other words, the core of RSTN is to approximate a function in terms of the difference of travel demands between adjacent time intervals. Thus, it can be said that the traditional travel demand forecasting problem is transformed into an entirely new one.

5. Experiments

In this section, we evaluate the proposed model on real-world data sets of taxi requests and see how well it can predict short-term travel demand. Meanwhile, we compare our architecture with several other models. Ablation study is also performed to demonstrate the proposed method.

5.1. Experimental setup

The evaluation uses two real-world data sets (NYC taxi trip data (Commission) collected in NewYork City and DiDi ChuXing travel data (ChuXing) collected in Hai Kou, China). The NYC taxi trip data set contains about 15 million travel records per month. We choose the recent 12 months data as our training set and the other 6 months data for test. The DiDi ChuXing data set contains five-months travel requests. The region of interest is divided into 11 × 11 grids uniformly, i.e., $I = J = 11$. The time interval is set at 30 min and the sub-time interval is 10 min, hence the length of DRV is 3 ($k = 3$). Before training the deep models, the collected data is normalized by z-score process. Our architecture is built using Tensorflow and Python and trained on a server computer with NVIDIA TITAN Xp GPU and Intel(R) i9-9980XE CPU. The total training time is about 6 h and 2.5 h for the two data sets, respectively.

For the convenience of model building and considering limitation of computing power, the sequence length is set at 48 (24 h). All the steps are optimized by Adam optimizer and the initial learning rate is 0.01 for the first 10 epochs and decays at a rate of 0.1 for every 10 epochs.

5.2. Performance metrics

To evaluate the proposed model in comparison with other methods, we use performance metrics including Symmetric Mean Absolute Percentage Error (sMAPE) (Moreira-Matias, 2013), mean absolute error (MAE) (Ke, 2017), and Root Mean Square Error

(RMSE) (Lv et al., 2015), which are defined as below:

$$sMAPE_{ij} = \frac{1}{T}\sum_{t=1}^{T}\frac{|Y_t^{ij} - \widehat{Y}_t^{ij}|}{Y_t^{ij} + \widehat{Y}_t^{ij} + c} \tag{29}$$

$$MAE_{ij} = \frac{1}{T}\sum_{t=1}^{T}|Y_t^{ij} - \widehat{Y}_t^{ij}| \tag{30}$$

$$RMSE_{ij} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(Y_t^{ij} - \widehat{Y}_t^{ij})^2} \tag{31}$$

where Y_t^{ij} and $\widehat{Y}_t^{ij}$ are the real and predicted travel demands at time-step t in area (i, j), c is a small number to avoid division by zero. Note that these metrics are defined for a single area (i, j) for simplicity, and their values corresponding to the whole region can be readily obtained.

5.3. Model comparisons and ablation study

An ablation study is performed first to show the effectiveness of the proposed model. Several benchmark algorithms are also constructed and tested, which are listed as follows:

RSTN: the architecture proposed with DRV input. nRSTN: the naive architecture of RSTN using real data as input rather than DRV.

STN: a spatio-temporal model using SCE and STDA without residual connection.

LSTM-C: an architecture in Xu (2018), which employs LSTM to model the temporal dynamics of travel demand and uses MDN for prediction.

ARIMA: a time series model in Box and Pierce (1970).

ST-ResNet: a spatio-temporal residual network in Zhang et al. (2017b) for crowd flow forecasting, which is rebuilt for travel demand prediction in our simulations.

FCL-Net: a fusion convolutional LSTM in Ke (2017) with three types of dependencies for short-term travel demand forecasting.

FCL-Net-v: FCL-Net with DRV input.

We first show the metrics of the whole region under different models of New York city and Hai Kou in Table 2. Note that all the models are terminated after 100 epochs in order to ensure fairness of comparisons. The first three rows in the table show the results of ablation study, from which we can see that the methods we proposed (DRV and RC) are indeed effective. The metrics (sMAPE, MAE and RMSE) of the proposed RSTN model are better than those of nRSTN and STN. Specifically, compared with nRSTN, the model trained without DRV-based input, the RSTN model reduces RMSE by 17.87% on NYC data. We also see that introducing RC can improve the model performance in comparison with STN.

The proposed model shows clear superiority over LSTM-C and ARIMA (as shown in the fourth and fifth rows in Table 2), which are both time-series-analysis algorithms incapable of dealing with spatial and extraneous dependencies. For both data sets, the performance metrics of RSTN decrease by 0.034/0.064, 0.984/0.920 and 1.099/2.224, respectively, compared with LSTM-C, and by 0.130/0.207, 1.430/3.551 and 1.922/6.998 compared with ARIMA. Also, our method is advantageous than deep neural network-based spatio-temporal methods ST-ResNet and FCL-Net, where ST-ResNet is in unfolded convolutional neural network with residual connections, and FCL-Net is based on Conv-LSTM without residual connections. It can be seen (from the sixth and seventh rows in Table 2)) that their performances are almost excellent but RSTN performs a little better. To further validate the effectiveness of the DRV-based representation, an additional experiment is implementated and the result is reported in the last row in Table 2. By taking the DRV as model input, the performance of FCL-Net is improved with negligible cost of calculation.

The results shown in Figs. 4–6 are metrics collected per hour from all the test data over New York city. Note that we use NYC data set in the remainder discussions due to its completeness and representativeness. It can be seen from these figures that RSTN is more stable and robust compared with LSTM-C and FCL-Net. Even for peak hours (6 am to 10 am, 4 pm to 8 pm), RSTN still has high

Table 2
Model comparison evaluated on two real-word datasets.

Model	NYC Taxi Trip Data			DiDi ChuXing Travel Data		
	MAE	RMSE	sMAPE	MAE	RMSE	sMAPE
RSTN	**0.541**	**1.082**	**0.078**	**1.301**	**3.007**	**0.093**
nRSTN	0.632	1.326	0.093	1.431	3.663	0.121
STN	0.560	1.102	0.085	1.352	3.109	0.118
LSTM-C	1.436	2.161	0.112	2.221	5.231	0.157
ARIMA	1.982	3.011	0.209	4.852	10.005	0.300
ST-ResNet	0.688	1.367	0.098	1.795	4.267	0.129
FCL-Net	0.594	1.113	0.088	1.401	3.241	0.123
FCL-Net-v	0.580	1.107	0.080	1.310	3.102	0.120

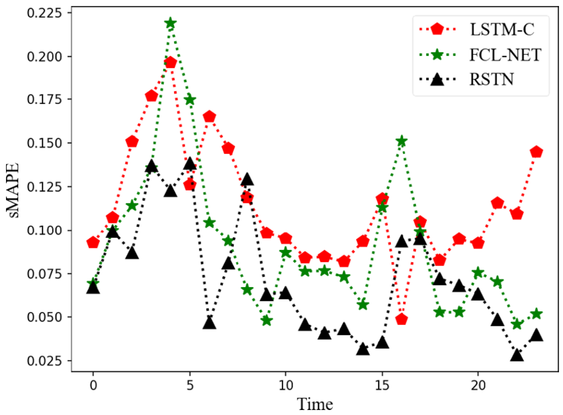


Fig. 4. Prediction performance of different approaches according to sMAPE.

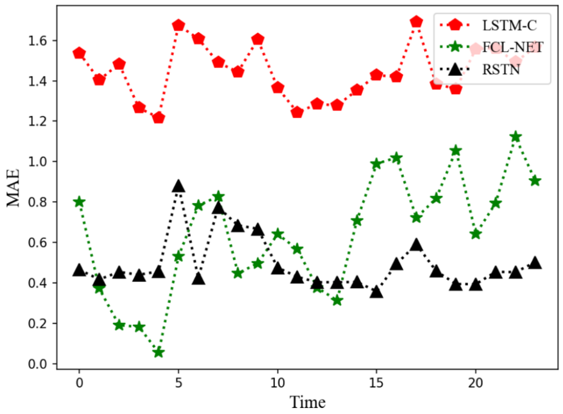


Fig. 5. Prediction performance of different approaches according to MAE.

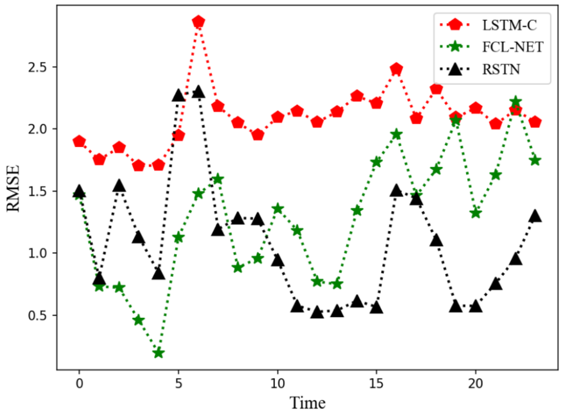


Fig. 6. Prediction performance of different approaches according to RMSE.

prediction accuracy. The performances of these models in two specific regions (one with high travel demand and the other with low demand) in New York city are shown in Table 3, where the maximum and minimum travel demands in a region are the mean values per hour calculated from the data set. In both regions, RSTN performs well with lower prediction errors except that MAE in region 1 is slightly worse than FCL-Net.

5.4. Architecture details

This subsection shows some details (i.e., parameters of the neural networks) about the end-to-end trainable model RSTN with DRV-based input.

Table 3
Performances in two specific regions of New York city.

model	region1 (min = 16, max = 162)			region2 (min = 70, max = 1270)		
	sMAPE	MAE	RMSE	sMAPE	MAE	RMSE
RSTN	0.075	0.693	1.377	0.130	0.215	0.427
LSTM-C	0.109	1.659	2.500	0.322	1.079	1.451
FCL-Net	0.077	0.598	1.522	0.200	0.219	0.627

Table 4
Results under different DRV lengths.

Input	DRV_1	DRV_2	DRV_3	DRV_4	DRV_5	DRV_6
sMSPE	0.090	0.087	0.078	0.078	0.079	0.079
MAE	0.632	0.606	0.541	0.539	0.542	0.543
RMSE	1.326	1.143	1.082	1.081	1.084	1.084

Table 5
Convolutional effectiveness exploration.

Model	$RSTN_{1,1}$	$RSTN_{1,2}$	$RSTN_{1,3}$	$RSTN_{2,1}$	$RSTN_{2,2}$	$RSTN_{2,3}$	$RSTN_{3,1}$	$RSTN_{3,2}$	$RSTN_{3,3}$
sMSPE	0.132	0.120	0.121	0.091	**0.078**	0.083	0.102	0.140	0.175
MAE	1.667	1.511	1.513	0.642	**0.539**	0.600	1.330	2.741	3.588
RMSE	2.761	2.444	2.450	1.355	**1.082**	1.284	2.167	4.368	7.156

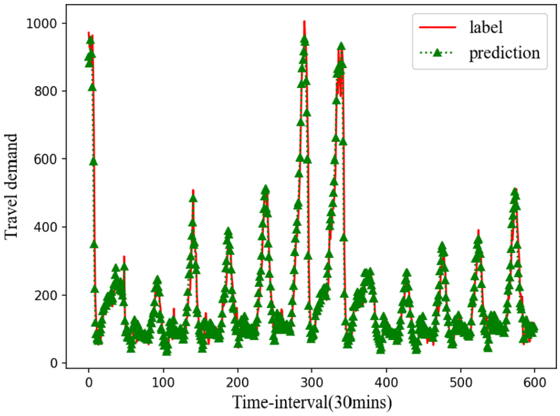


Fig. 7. prediction example of nonstationary area.

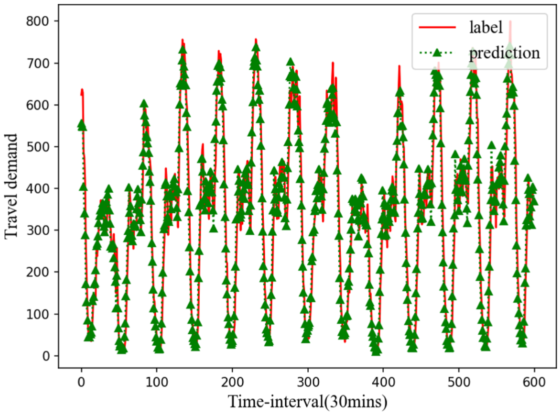
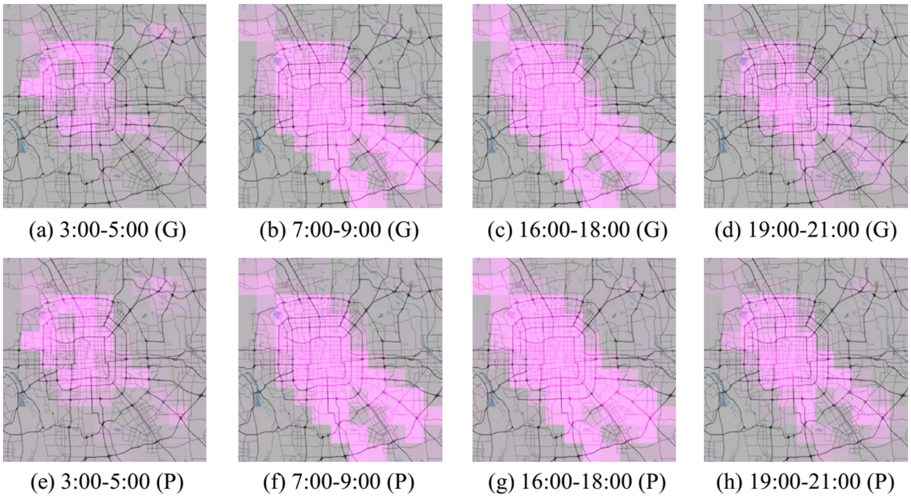


Fig. 8. prediction example of stationary area.


(a) 3:00-5:00 (G) (b) 7:00-9:00 (G) (c) 16:00-18:00 (G) (d) 19:00-21:00 (G)

(e) 3:00-5:00 (P) (f) 7:00-9:00 (P) (g) 16:00-18:00 (P) (h) 19:00-21:00 (P)

Fig. 9. Prediction examples over the city.

5.4.1. Length of DRV

The length of DRV, i.e., the number of sub-intervals in a time slot, N, is an important parameter affecting the performance. Table 4 reports the performance with different input lengths, where DRV_{N-1} stands for input length N. It can be seen that the performance metrics decrease with N when $N \leqslant 3$. To tradeoff complexity-accuracy, we use $N = 3$ in the simulations. It should be mentioned that RSTN reduces to nRSTN when $N = 1$.

5.4.2. Convolutional effectiveness

The convolutional layers and cell memory of LSTM play significant roles in spatio-temporal dependencies exploration for travel demand forecasting. We now investigate how many convolutional layers in a CE-LSTM cell should be used and how many CE-LSTM cells should be stacked together. They should to be determined simultaneously to balance the convolutional operation and the number of LSTM cells. Denote by n and nc, respectively, the two parameters in the architecture. Table 5 reports the performance of different models with different n and nc (denoted by $\text{RSTN}_{n,nc}$). It can be seen that LSTM-based models are subject to the curse of over-fitting and under-fitting. As shown in the table, $\text{RSTN}_{2,2}$ performs the best, meaning that one should set $n = 2$ and $nc = 2$.

5.5. Prediction examples

We first show prediction examples with RSTN in specific regions in Figs. 7 and 8. We use second-order weak stationarity condition to verify the stationarity of travel demand in each region by calculating the positive definiteness of the autocovariance matrix:

$$\Gamma_n = \begin{bmatrix} \gamma_0 & \gamma_1 & \gamma_2 & \cdots & \gamma_{n-1} \\ \gamma_1 & \gamma_0 & \gamma_1 & \cdots & \gamma_{n-2} \\ \gamma_2 & \gamma_1 & \gamma_0 & \cdots & \gamma_{n-3} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \gamma_{n-1} & \gamma_{n-2} & \gamma_{n-3} & \cdots & \gamma_0 \end{bmatrix} \tag{32}$$

where $\gamma_k = cov[d_t, z_{d+k}] = E[(d_t - \mu)(d_{t+k} - \mu)]$, $\mu = E[d_t]$ and d_t is the observation of demand at time t in a specific area. Then two representative areas are selected to show the results. Fig. 7 reports the prediction example in a non-stationary area, while Fig. 8 shows a stationary one. It is obvious that RSTN works well in both two scenarios, demonstrating its effectiveness and generalizability. The results over the whole city is shown in Fig. 9 for four different time periods in the day, wherein Fig. 9(a)–(d) are the ground-truth values and the others are their predictions. We assign deeper color for larger travel requests. It can be clearly seen that RSTN performs well for the whole city.

6. Conclusions

In this study, we propose an easy-to-construct deep architecture named residual spatio-temporal network (RSTN) for short-term travel demand forecasting, which employs FCNs and a hybrid module consisting of LSTMs, CE-LSTMs and CNNs connected in a residual way. The model takes three types of dependencies into account by fitting the relationship among all types of variables (spatio-temporal, spatial, temporal) and changes of travel demand. A novel representations of demand density termed DRV is also proposed to improve the performance. The validity of the model is demonstrated using two real-world data sets. We can conclude that, compared with several algorithms, i.e., LSTM-C, ST-ResNet, FCL-Net and ARIMA, the RSTN performs better. In addition, the proposed method works well in a set of ablation studies.

This paper is mainly based on the taxi data to explore the time–space patterns of travel demand in a city. A promising research topic is to apply the results to intelligent transportation systems such as Automated-Mobility-on-Demand (AMoD) systems, to improve traffic efficiency and throughput. However, the existing data does not fully reflect practical traffic behavior. More in-depth research along this direction calls for more comprehensive data. In addition, this paper uses a vector-based method found in the area of natural language processing for travel demand prediction. Are there other ways by which the dynamics of travel demand can be better captured? This is another interesting research topic worth invesitgation.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 61573077 and U1808205). The authors also gratefully acknowledge DiDi Chuxing GAIA Initiative for the travel data.

References

Bengio, Y.S.P.F.P., 1994. Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Network 5, 157–166.

Box, G.E.P., Pierce, D.A., 1970. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. Publicat. Am. Stat. Assoc. 65, 1509–1526.

ChuXing, D., Didi chuxing travel request data. <https://gaia.didichuxing.com/>.

Commission, N.T.L., Taxi and limousine commission (tlc) trip record data. <http://www.nyc.gov/html/tlc/html/>.

Davis, N.R.G.J.K., 2016. A multi-level clustering approach for forecasting taxi travel demand. In: IEEE International Conference on Intelligent Transportation Systems, pp. 223–228.

Jo, D., Yu, B., Jeon, H., Sohn, K., 2019. Image-to-image learning to predict traffic speeds by considering area-wide spatio-temporal dependencies. IEEE Trans. Veh. Technol. 68, 1188–1197.

Kai, Z., Feng, Z., Chen, S., Huang, K., Wang, G., 2016. A framework for passengers demand prediction and recommendation. In: 2016 IEEE International Conference on Services Computing (SCC).

Kaltenbrunner, A., Meza, R., Grivolla, J., Codina, J., Banchs, R., 2010. Urban cycles and mobility patterns: exploring and predicting trends in a bicycle-based public transport system. Pervasive Mobile Comput. 6.

Ke, J.Z.H.Y.H., 2017. Short-term forecasting of passenger demand under on-demand ride services: a spatio-temporal deep learning approach. Transport. Res. Part C: Emerg. Technol. 85, 591–608.

Li, X.P.G.W.Z., 2012. Prediction of urban human mobility using large-scale taxi traces and its applications. Front. Comput. Sci. 6, 111–121.

Li, B.Z.D.S.L., 2011. Hunting or waiting? Discovering passenger-finding strategies from a large-scale realworld taxi dataset. IEEE PerCom 21–25.

Liu, L., C.R., 2017. A mrt daily passenger flow prediction model with different combinations of influential factors. In: International Conference on Advanced Information Networking and Applications Workshops.

Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.Y., 2015. Traffic flow prediction with big data: a deep learning approach. IEEE Trans. Intell. Transp. Syst. 16, 865–873.

Ma, J., Chan, J.R.G., 2019. Bus travel time prediction with real-time traffic information. Transport. Res. Part C: Emerg. Technol. 105, 536–549.

Min, W., Wynter, L., 2011. Real-time road traffic prediction with spatio-temporal correlations. Transport. Res. Part C: Emerg. Technol. 19, 606–616.

Moreira-Matias, L.G.J., 2013. Predicting taxi–passenger demand using streaming data. IEEE Trans. Intell. Transp. Syst. 14, 1393–1402.

Mukai, N., Yoden, N., 2012. Taxi demand forecasting based on taxi probe data by neural network. Springer, Berlin Heidelberg, pp. 589–597.

Nguyen, H.K.L.W.T., 2018. Deep learning methods in transportation domain: a review. IET Intel. Transport Syst. 12, 998–1004.

Ren, Y., Cheng, T., Zhang, Y., 2019. Deep spatio-temporal residual neural networks for road-network-based data modeling. Int. J. Geogr. Inform. Sci. 1–19.

Shi X, Chen Z, W.H., 2015. Convolutional lstm network: a machine learning approach for precipitation nowcasting. Adv. Neural Inform. Process. Syst., pp. 802–810.

Sun, S.Z.C.Y.G., 2006. A bayesian network approach to traffic flow forecasting. IEEE Trans. Intell. Transp. Syst. 1, 124–132.

Wu, X., Guo, J.F.X.K., 2018. Hierarchical travel demand estimation using multiple data sources: a forward and backward propagation algorithmic framework on a layered computational graph. Transport. Res. Part C: Emerg. Technol. 96, 321–346.

Xu, J.R.R.B.L., 2018. Real-time prediction of taxi demand using recurrent neural networks. IEEE Trans. Intell. Transp. Syst. 99, 1–10.

Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., Gong, P., Ye, J., Li, Z., 2018. Deep multi-view spatial-temporal network for taxi demand prediction.

Yu, H., Zhihai, W., Shuqin, W., Yunpeng, W., Xiaolei, M., 2017. Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. Sensors 17, 1501–1510.

Yuan, C., Yu, X.L.D., Xi, Y., 2019. Overall traffic mode prediction by vomm approach and ar mining algorithm with large-scale data. IEEE Trans. Intell. Transp. Syst. 20, 1508–1516.

Zahran, M.A., Magooda, A., Mahgoub, A.Y., Raafat, H., Rashwan, M., Atyia, A., 2015. Word Representations in Vector Space and their Applications for Arabic.

Zhang, Shuaichao, Chen, Xiqun, (Michael), Zahiri, Majid, 2017a. Understanding ridesplitting behavior of on-demand ride services: an ensemble learning approach. Transport. Res. Part C: Emerg. Technol. 76, 51–70.

Zhang, J., Yu, Z., Qi, D., 2017b. Deep spatio-temporal residual networks for citywide crowd flows prediction. AAAI 1655–1661.

Zhu, L., L.N., 2017. Deep and confident prediction for time series at uber. IEEE Int. Conf. Data Mining Workshops, 103–110.