



Predicting vacant parking space availability zone-wisely: a hybrid deep learning approach

Yajing Feng¹ · Yingying Xu^{1,2} · Qian Hu^{1,2}  · Sujatha Krishnamoorthy³ · Zhenzhou Tang^{1,2}

Received: 4 August 2021 / Accepted: 11 February 2022 / Published online: 18 March 2022
© The Author(s) 2022

Abstract

Precise prediction on vacant parking space (VPS) information plays a vital role in intelligent transportation systems for it helps drivers to find the parking space quickly to reduce unnecessary waste of time and excessive environmental pollution. By analyzing the historical zone-wise VPS data, we find that for the number of VPSs, there is not only a solid temporal correlation within each parking lot, but also an obvious spatial correlation among different parking lots. Given this, this paper proposes a hybrid deep learning framework, known as the dConvLSTM-DCN (dual Convolutional Long Short-Term Memory with Dense Convolutional Network), to make short-term (within 30 min) and long-term (over 30 min) predictions on the VPS availability zone-wisely. Specifically, the temporal correlations of different time scales, namely the 5-min and daily-wise temporal correlations of each parking lot, and the spatial correlations among different parking lots can be effectively captured by the two parallel ConvLSTM components, and meanwhile, the dense convolutional network is leveraged to further improve the propagation and reuse of features in the prediction process. Besides, a two-layer linear network is used to extract the meta-info features to promote the prediction accuracy. For long-term predictions, two methods, namely the direct and iterative prediction methods, are developed. The performance of the prediction model is extensively evaluated with practical data collected from nine public parking lots in Santa Monica. The results show that the dConvLSTM-DCN framework can achieve considerably high accuracy in both short-term and long-term predictions.

Keywords Vacant parking space availability · Deep Learning · ConvLSTM · Dense Network

Introduction

Parking is a headache for drivers. It is difficult for drivers to, frequently, find a vacant parking space (VPS) quickly after arriving at their destination. It takes 3–15 min for drivers to find a VPS on average [1]. Drivers looking for parking account for high percentages of traffic congestion [2]. Cruising the parking not only increases the traffic flow by

25–40% [3], it also results in increased exhaust emissions. Many intelligent parking assistant systems have been developed to provide real-time VPS locations to ease traffic issues [4–8]. Equipped with sensor devices and surveillance equipment, these systems collect and display the numbers of VPSs in real time to help drivers. However, real-time VPS information is not satisfactory, because the number of VPSs in a given parking lot is likely to have changed by the time the driver has arrived. Consequently, accurate VPS prediction is considered a more promising technology as it enables drivers to learn the VPS availability in advance. In view of this, recently more research has been reported on VPS availability predictions. These works can be divided into two categories, namely the statistical learning model-based prediction methods and the Machine Learning (ML) and Deep Learning (DL)-based prediction methods.

Statistical learning model-based prediction methods Assuming that the processes of vehicle arrival and departure follow the Poisson distributions, Caliskan et al. [9] employed the continuous-time homogeneous Markov model to pre-

Yajing Feng and Yingying Xu contributed equally to this work.

✉ Qian Hu
huqian@wzu.edu.cn

✉ Zhenzhou Tang
mr.tangzz@gmail.com

¹ College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou 325035, China

² Innovation Research Center for Intelligent Networking, Wenzhou University, Wenzhou 325035, China

³ Department of Computer Science, Wenzhou Kean University, Wenzhou 325060, China

dict the number of VPSs. Xiao et al. [10] employed the continuous-time Markov M/M/C/C queue model to effectively estimate the arrival and departure rates of vehicles, with which the number of VPSs was further estimated. Calicedo et al. [11] proposed a centralized system to estimate the number of VPSs according to the request allocation and simulated parking duration of the discrete Gamma distribution. However, these statistical learning models were highly dependent on the assumptions of arrival and departure processes. Therefore, they can hardly be applied to parking lots with rich time-varying statistical characteristics.

ML/DL-based prediction methods By leveraging the incredible learning ability of ML/DL models, some ML/DL-based methods have been proposed to predict the VPS availabilities. In [12], an Auto-Regressive Integral Moving Average (ARIMA) model was proposed to predict the unoccupied parking space. And in [13], three ML models were leveraged to predict the number of VPSs: Regression Tree, Back Propagation Neural Network (BPNN), and Support Vector Machine (SVM). Ji et al. [14] put forth a short-term available parking space prediction model based on Wavelet Neural Networks (WNN). In [15], multi-dimensional features, including historical VPS information, weather condition, and traffic data, were uploaded to a Bayesian Regularization Neural Network (BRNN) to predict the VPS availability. Fan et al. [16] proposed an SVR model whose essential parameters were fine-tuned by a fruit fly optimization algorithm to predict the number of VPSs. In [17], an extensively fine-tuned Long Short-Term Memory (LSTM) model was developed to predict the number of VPSs.

The aforementioned ML/DL-based methods provided good predictions, but only made use of the temporal correlation of the historical data in the single target parking lot and ignored the spatial correlation among multiple parking lots within the same zone. Although methods proposed in [18,19] took the spatial-temporal correlation into consideration, nevertheless, they likewise only predicted the VPS availability of a single rather than multiple parking lots in the same zone.

Motivated by the above, in this paper, a hybrid DL framework named dConvLSTM-DCN which involves two parallel convolutional LSTM networks (ConvLSTMs) [20], a two-layer linear network and a Dense Convolutional Network (DCN) [21], is proposed to make zone-wise predictions on VPS availability for all parking lots in the zone simultaneously. This work is a further deep extension of our previous work [22]. Specifically, the improvements are threefold. First, we improve the model by developing the parallel dual-ConvLSTMs (dConvLSTM) to capture the recent and daily temporal correlations, respectively. Second, we build a meta-info feature extraction component to utilize the meta-info consisting of the day of the week, the hour of the day, the minute of the hour and weekday/weekend, since the number

of VPSs shows an obvious daily periodicity. Third, two different long-term prediction methods, namely the direct and iterative prediction methods, are investigated and compared in this work.

The main contributions of this paper are summarized as follows:

- We analyze both the temporal correlation of the historical VPS information in each parking lot and the spatial correlation among all the parking lots in the same zone. The spatial-temporal correlation imposes a great significance on the prediction model.
- We propose the novel dConvLSTM-DCN framework to predict the numbers of VPS for all parking lots within the target zone simultaneously. To be specific, the parallel dual-ConvLSTM component is developed to capture the spatial-temporal correlations of historical VPS information, the two-layer linear network is used to extract the meta-info features, and the dense connection pattern is leveraged to further improve the propagation and reuse of features in the prediction process.
- The temporal correlations of two different scales, namely the 5-min temporal correlation and daily-wise temporal correlation, are captured by the dConvLSTM component. In addition, two different long-term prediction methods, namely the direct and iterative prediction methods, are investigated to further improve the prediction accuracy.
- Sufficient comparative experiments have been conducted to evaluate the effectiveness of the proposed prediction method. The results show that the dConvLSTM-DCN framework is capable of making both short-term predictions and long-term predictions, with a considerably high accuracy.

The rest of this paper is organized as follows: The section “Methodology” describes the detailed dConvLSTM-DCN-based prediction method. The section “Experimental results” presents the comparative experiment results and analysis. Finally, the section “Conclusion” concludes the paper.

Methodology

In this section, the novel dConvLSTM-DCN framework is introduced in detail and the proposed VPS availability prediction method is presented.

Data description and preprocessing

The data were collected from nine public parking lots (St1–St9) in Santa Monica, California, USA (longitude range [−118.499378, −118.49372], latitude range [34.019575, 34.01289]) from 7:00 on May 01, 2018 to 21:50 on June 11, 2018 [23]. The number of VPSs was collected every 5

min, so that a total of 11987 historical data were obtained in each parking lot.

The target zone is divided into $H \times W$ grids, each of which is $100\text{ m} \times 100\text{ m}$ (Fig. 1). After the division, each parking lot is distributed in a grid which has at most one parking lot. For the grid without parking lot, it can be regarded as a parking lot with zero VPS. Then, the number of VPSs in this zone at time t is denoted as

$$D_t = \begin{bmatrix} d_t^{(1,1)} & d_t^{(1,2)} & \dots & d_t^{(1,W)} \\ d_t^{(2,1)} & d_t^{(2,2)} & \dots & d_t^{(2,W)} \\ \vdots & \vdots & \ddots & \vdots \\ d_t^{(H,1)} & d_t^{(H,2)} & \dots & d_t^{(H,W)} \end{bmatrix}, \tag{1}$$

where each element in the matrix, denoted as $d_t^{(h,w)}$, $h \in [0, H]$, $w \in [0, W]$, is the number of VPSs in grid (h, w) .

All the data were scaled within the range of $[0,1]$ by the Min–Max normalization.

Next, the preliminary analysis of the spatial–temporal correlations of the historical VPS information is conducted. Due to space limitations, only the spatial–temporal correlation analysis results of the parking lot S7 are present. However, it should be noted that the spatial–temporal analysis method is applicable to all the other parking lots.

Temporal correlations

Figure 2a shows the dynamic temporal characteristics of number of VPSs of S7, where x -axis represents the number of time intervals and y -axis represents the number of VPSs. As can be seen from the figure, the number of VPSs varies with a strong regularity. Figure 2b shows the temporal autocorrelation of parking lot S7, where x -axis represents the time interval τ and y -axis represents the autocorrelation coefficient r which can be calculated as

$$r = \frac{1}{(T - \tau)} \sum_{t=1+\tau}^T \frac{d_t^{h,w}}{d_{t-\tau}^{h,w}}, \tag{2}$$

where T is the number of time intervals of the dataset and (h, w) is the two-dimensional index of the given grid. It can be observed from the figure that r is always greater than 0.6, indicating that the number of VPSs of a given parking lot is positively correlated in the time domain. This suggests that the number of VPSs can be effectively predicted through historical data. Moreover, the data in adjacent time periods are more correlated than those farther apart in time.

Spatial correlations

To analyze the spatial correlation between any two parking lots in the target zone, the correlation coefficient matrix is

calculated. Each element in this matrix is the Pearson correlation coefficient which can be used to measure the spatial correlation between the corresponding two parking lots, and is defined as follows:

$$\rho = \frac{\text{cov}(d^{(i,j)}, d^{(i',j')})}{\sigma_{d^{(i,j)}} \sigma_{d^{(i',j')}}}, \tag{3}$$

$\text{cov}(\cdot)$ is the covariance function, and σ represents the standard deviation operation. The spatial correlations among St1–St9 as shown in Fig. 2c can be clearly seen that there exists obvious spatial correlations between parking lots of different areas. This calculation can be employed to increase the accuracy of predictions.

The preliminary spatial–temporal analysis on historical number of VPSs verifies that the number of VPSs has a strong spatial–temporal correlation and is predictable.

Prediction model

Next, we present the dConvLSTM-DCN framework which consists of four components, namely the two ConvLSTM components, meta-info feature extraction component, and the DCN component (Fig. 3). The first captures the spatial–temporal correlations, the second is used to extract features in meta information, and the last is used to extract features from the output of the dual-ConvLSTM component.

Spatio-temporal correlations modeling

It is well known that although a Convolutional Neural Network (CNN) has a strong spatial modeling ability, a CNN cannot accurately extract the temporal correlations. Instead, LSTM [24] can establish the temporal correlations for time series data. It is natural to integrate CNN and LSTM together to form a three-layer ConvLSTM network to simultaneously capture the spatio-temporal correlations.

The ConvLSTM model is a special Recurrent Neural Network (RNN). First, like all other RNNs, a ConvLSTM allows previous outputs to be used as inputs while having hidden states. The connections between nodes form a directed or undirected graph along a temporal sequence, which is the most important characteristics of RNN. Moreover, different from traditional RNNs, ConvLSTM also has the salient feature of CNN, that is, the hidden layers include layers that perform convolutions. Figure 4 shows the internal structure of a ConvLSTM cell. A ConvLSTM cell is controlled by three gates, namely the forget gate f_g , the input gate i_g , and the output gate o_g . Specifically, whenever a new input arrives at a ConvLSTM cell, the information carried by the input will be stored in the memory cell if the input gate i_g is open. Similarly, information held in the current memory cell will be forgotten if the forget gate f_g is open. The output gate

Fig. 1 Distribution of the parking lots



o_g controls the value of the final output and determines the final hidden state. The operation of a ConvLSTM cell over the input \mathbf{D}_t at any given time point t can be defined as

$$\begin{aligned} i_g &= \sigma(\mathbf{w}_{di} \otimes \mathbf{D}_t + \mathbf{w}_{hi} \otimes \mathcal{H}_{t-1} + \mathbf{w}_{ci} \odot \mathcal{C}_{t-1} + b_i), \\ f_g &= \sigma(\mathbf{w}_{df} \otimes \mathbf{D}_t + \mathbf{w}_{hf} \otimes \mathcal{H}_{t-1} + \mathbf{w}_{cf} \odot \mathcal{C}_{t-1} + b_f), \\ \mathcal{C}_t &= i_g \odot \tanh(\mathbf{w}_{dc} \otimes \mathbf{D}_t + \mathbf{w}_{hc} \otimes \mathcal{H}_{t-1} + b_c) \\ &\quad + f_g \odot \mathcal{C}_{t-1}, \\ o_g &= \sigma(\mathbf{w}_{do} \otimes \mathbf{D}_t + \mathbf{w}_{ho} \otimes \mathcal{H}_{t-1} + \mathbf{w}_{co} \odot \mathcal{C}_t + b_o), \\ \mathcal{H}_t &= o_g \odot \tanh(\mathcal{C}_t), \end{aligned} \quad (4)$$

where $\sigma(\cdot)$ is the activation function, \otimes is the convolution operation, \odot is the dot product operation, $\tanh(\cdot)$ is a hyperbolic tangent function of the nonlinear excitation function, $\mathbf{w}_{(\cdot)}$ and $b_{(\cdot)}$ are weights and bias to be learned, respectively, and \mathcal{C}_t and \mathcal{H}_t are three-dimensional tensors stored in the memory cell and final hidden state at time step t , respectively.

Two ConvLSTM blocks are built to capture the recent and daily temporal correlations, respectively, as shown in Component 1 and 2 in Fig. 3. Component 1 (consists of $m \times$

k ConvLSTM cells) and Component 2 (consists of $n \times k$ ConvLSTM cells) are used to extract the spatial-temporal correlations among all parking lots based on m 5-min scale and n daily scale historical observations, respectively. For each prediction, m 5-min scale recent historical observations, denoted as

$$\mathbf{X}_{\text{recent}} = [\mathbf{D}_{t_c - (m-1)\delta}, \mathbf{D}_{t_c - (m-2)\delta}, \dots, \mathbf{D}_{t_c}]$$

are input into Component 1 where $\delta = 5$ min is the time step size for recent temporal correlations, and t_c is the current time. The obtained output $\mathbf{O}_r \in \mathbb{R}^{m \times H \times W}$ is written as

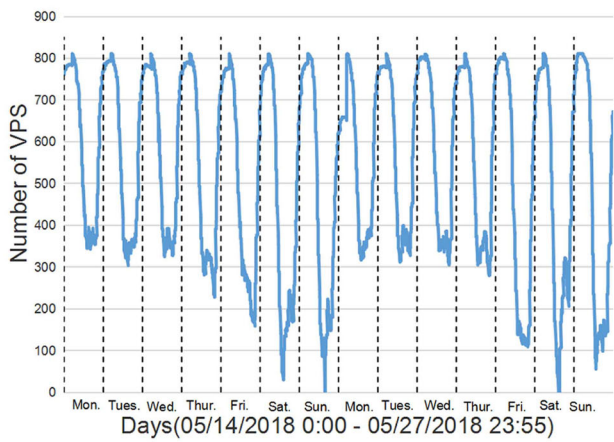
$$\mathbf{O}_r = \mathcal{H}_{t_c} \oplus \dots \oplus \mathcal{H}_{t_c - (m-2)\delta} \oplus \mathcal{H}_{t_c - (m-1)\delta},$$

where \oplus is the concatenation operation.

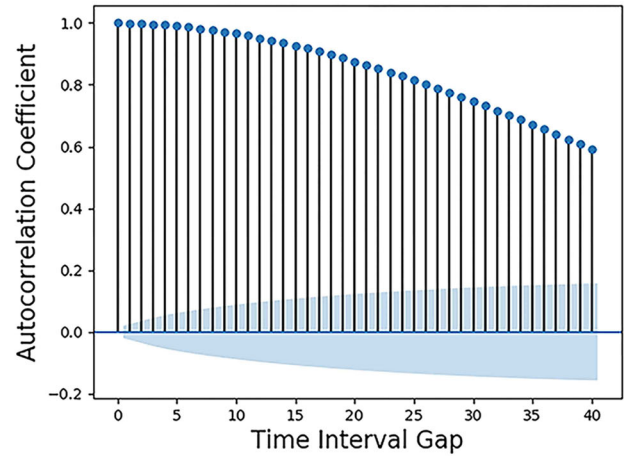
Moreover, n daily scale observations denoted as

$$\mathbf{X}_{\text{daily}} = [\mathbf{D}_{t_c - (n-1)\delta_d}, \mathbf{D}_{t_c - (n-2)\delta_d}, \dots, \mathbf{D}_{t_c}],$$

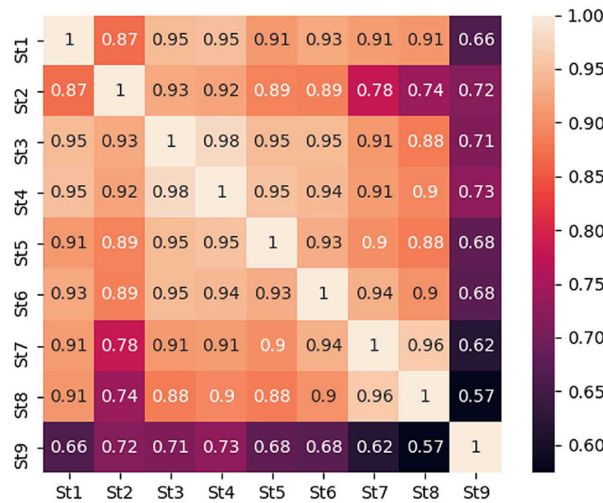
where $\delta_d = 24 \times 12 \times \delta$ is the daily scale time step size are input into Component 2 to capture the daily scale temporal



(a) Number of VPSs in time series.



(b) Temporal correlations of number of VPSs in time series.



(c) Spatial correlations (Pearson correlations) between different parking lots.

Fig. 2 The spatial and temporal dynamics of parking space

correlations. The corresponding output is

$$\mathbf{O}_d = \mathcal{H}_{t_c} \oplus \dots \oplus \mathcal{H}_{t_c-(n-2)\delta_d} \oplus \mathcal{H}_{t_c-(n-1)\delta_d},$$

and $\mathbf{O}_d \in \mathbb{R}^{n \times H \times W}$.

Meta-info feature extraction

It can be read from Fig. 2a that the number of VPSs shows an obvious daily periodicity, which implies that the information of hour of day and day of week is of clear significance to the predictions. In view of this, the meta-info feature extraction component is involved in the model. Concretely, the input time & date is extracted into four features. For instance,

the information of 7:15:00 05/11/2021 can be extracted as a four-dimensional meta-info vector \mathbf{m} which consists of the day_of_week (Friday), hour_of_day (7), minute_of_hour (15) and is_weekend (no).

Considering that there are $H \times W$ grids in each feature map, \mathbf{m} needs to be reshaped, such that the output of the component $\mathbf{O}_m \in \mathbb{R}^{H \times W}$. For this purpose, \mathbf{m} is first fed into a two-layer linear network and we obtain

$$\mathbf{o}_m = \sigma(\mathbf{w}_{m,2}\sigma(\mathbf{w}_{m,1}\mathbf{m} + b_{m,1}) + b_{m,2}), \tag{5}$$

where $\mathbf{w}_{m,i}, b_{m,i}, i = 1, 2$ are the weights and biases of the i th linear function, respectively, and $\mathbf{o}_m \in \mathbb{R}^{H \times W \times 1}$. Therefore, after a Reshape function, the final output of

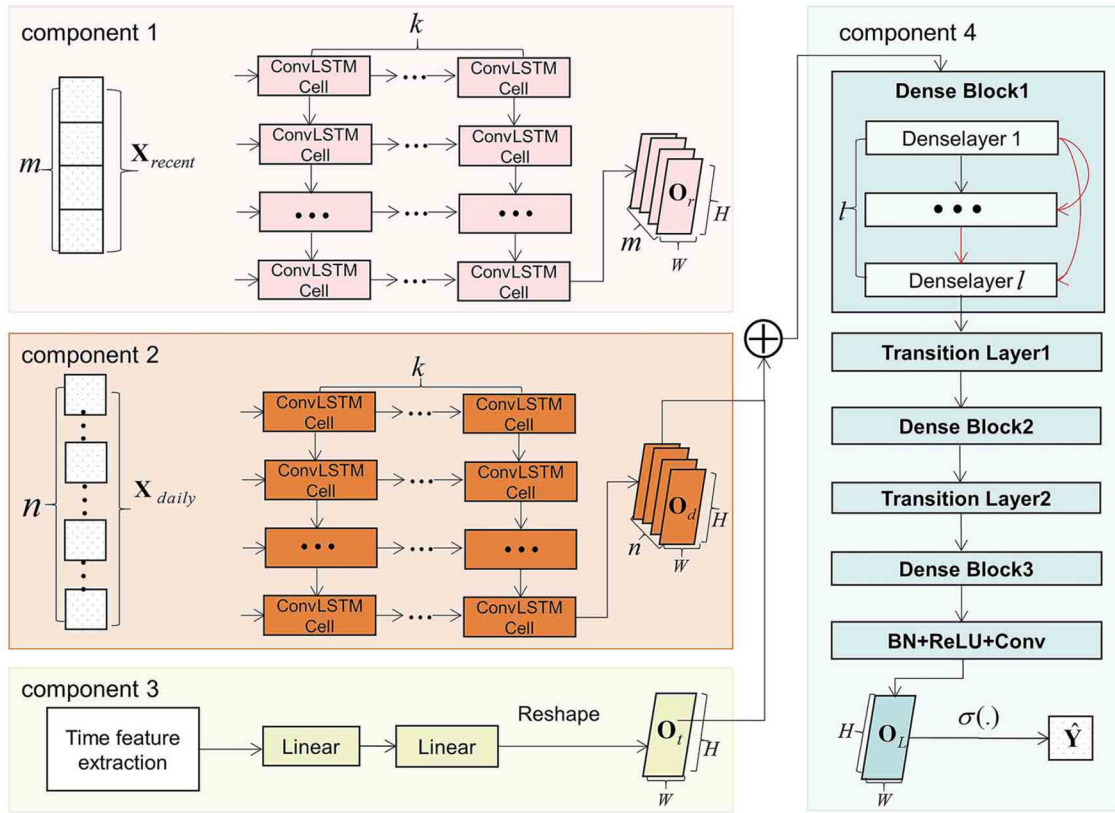


Fig. 3 Framework of the proposed model. There are four components in the model, i.e., the two ConvLSTM components for spatial-temporal feature extraction (“Spatio-temporal correlations modeling”), the meta-

info feature extraction component (“Meta-info feature extraction”), and the DCN component (“Densenet for feature learning”)

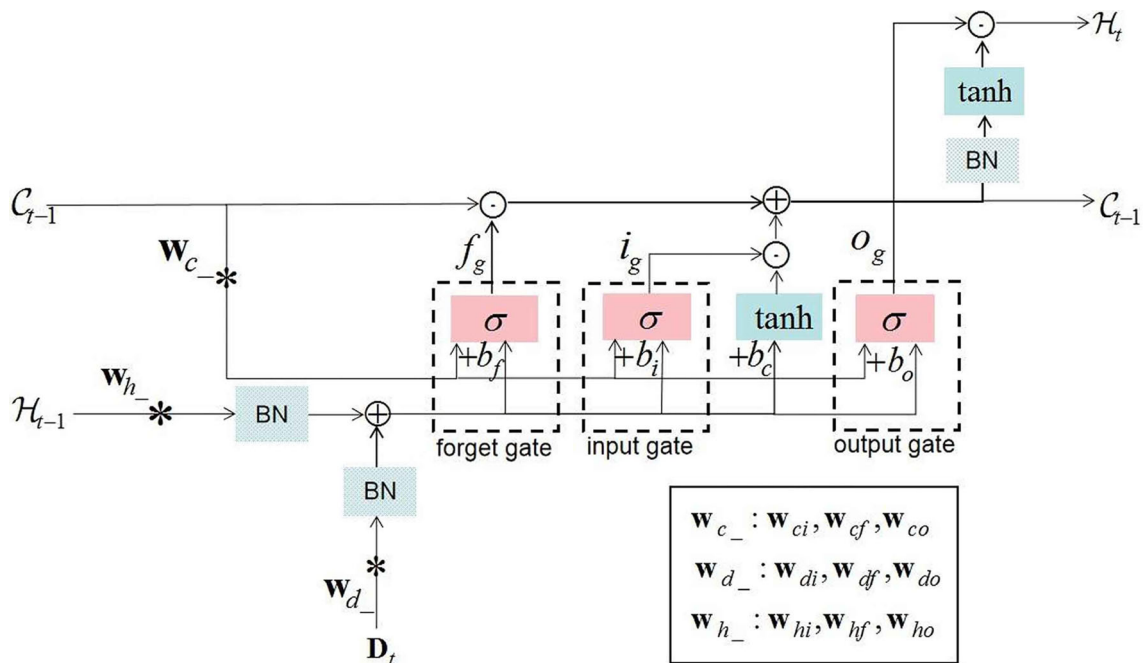


Fig. 4 Internal structure of a ConvLSTM cell

the meta-info feature extraction component is $\mathbf{O}_m, \mathbf{O}_m \in \mathbb{R}^{H \times W}$.

Densenet for feature learning

The outputs of the two ConvLSTMs and meta-info feature extraction component are concatenated together, denoted as $\mathbf{O}_0 = \mathbf{O}_r \oplus \mathbf{O}_d \oplus \mathbf{O}_t$, and fed into DCN. Due to its compact internal represents and reused feature redundancy, DCN may be a promising feature extractor [21]. Given this, DCN is used for feature learning in our work.

The DCN component, as shown in component 4 of Fig. 3, consists of six blocks, namely Dense Block 1, Transition Layer 1, Dense Block 2, Transition Layer 2, Dense Block 3, and a composite function of three consecutive operations, i.e., batch normalization (BN), a rectified linear activation (ReLU), and a 1×1 convolution (Conv), denoted as BN-ReLU-Conv(1×1). Each Dense Block consists of l Denselayers. Each Denselayer has the structure of BN-ReLU-Conv(1×1)-BN-ReLU-Conv(3×3) and is directly connected to all its subsequent layers. This means that Denselayer l receives the outputs of all preceding Denselayers, i.e., $\mathbf{O}_0, \mathbf{O}_1, \dots, \mathbf{O}_{l-1}$. Therefore, the output of the l^{th} Denselayer can be expressed as

$$\mathbf{O}_l = f_l(\mathbf{O}_0 \oplus \mathbf{O}_1 \oplus \dots \oplus \mathbf{O}_{l-1}), \tag{6}$$

where $f_l(\cdot)$ represents the composite function of BN-ReLU-Conv(1×1)-BN-ReLU-Conv(3×3). The transition layers do convolution and pooling. The transition layers used in our model are composed of a BN layer and a 1×1 Conv layer followed by a 2×2 average pooling layer.

Finally, after the sigmoid activation function, the final prediction result can be generated.

Prediction methods

We use the gridded two-dimensional historical VPS matrix sequence $[\mathbf{D}_{t_c-(m-1)\delta}, \mathbf{D}_{t_c-(m-2)\delta}, \dots, \mathbf{D}_{t_c}]$ to simultaneously predict the number of VPSs for all the parking lots in the target zone at time $t_c + h\delta$, denoted as $\hat{\mathbf{Y}}_{t_c+h\delta}$, where $h\delta$ represents the predicted time step. A prediction over the period of time that no more than 30 min, i.e., $h\delta \leq 30$ min, is defined as a short-term prediction. Otherwise, we call it a long-term prediction, as shown in Fig. 5.

Two fundamental prediction methods, namely the single-step direct and multi-step iterative prediction, are put forward to make short- and long-term predictions. Concretely, in single-step direct predictions, the result $\hat{\mathbf{Y}}_{t_c+h\delta}$ is directly predicted by using m historical observation data. While in multi-step iterative prediction methods, $\hat{\mathbf{Y}}_{t_c+\delta}$ is first pre-

dicted with

$$[\mathbf{D}_{t_c-(m-1)\delta}, \mathbf{D}_{t_c-(m-2)\delta}, \dots, \mathbf{D}_{t_c}]$$

by single-step direct predictions. Then, $\hat{\mathbf{Y}}_{t_c+2\delta}$ is predicted with $[\mathbf{D}_{t_c-(m-2)\delta}, \dots, \mathbf{D}_{t_c}, \mathbf{Y}_{t_c+\delta}]$. After that, $\mathbf{Y}_{t_c+2\delta}$ is further used to iteratively predict $\mathbf{Y}_{t_c+3\delta}$, and so on.

Experimental results

Experimental setup and evaluation indicators

The Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE) are adopted to measure the accuracy of the predicted values. Specifically, MAE is the average of the absolute errors between the predicted and real values. MAE can avoid the problem that the positive and negative errors cancel each other out, so it can accurately reflect the actual prediction errors. MAE can be calculated as follows:

$$\text{MAE} = \frac{1}{T} \sum_{t=1}^T \left| y_t^{(h,w)} - \hat{y}_t^{(h,w)} \right|, \tag{7}$$

where $y_t^{(h,w)}$ and $\hat{y}_t^{(h,w)}$ are the actual and predicted numbers of VPSs of grid (h, w) at time t , respectively, and T stands for the time step.

MAPE measures the percentage error of the prediction in relation to the actual values and usually expresses the accuracy as a ratio defined by the formula

$$\text{MAPE} = \frac{100\%}{T} \sum_{t=1}^T \frac{\left| y_t^{(h,w)} - \hat{y}_t^{(h,w)} \right|}{y_t^{(h,w)}}. \tag{8}$$

It should be stated that in some records, the actual numbers of VPSs of St3 and St5 are zeros, which leads to infinite MAPEs. In these cases, SMAPE (Symmetric MAPE) that is defined as

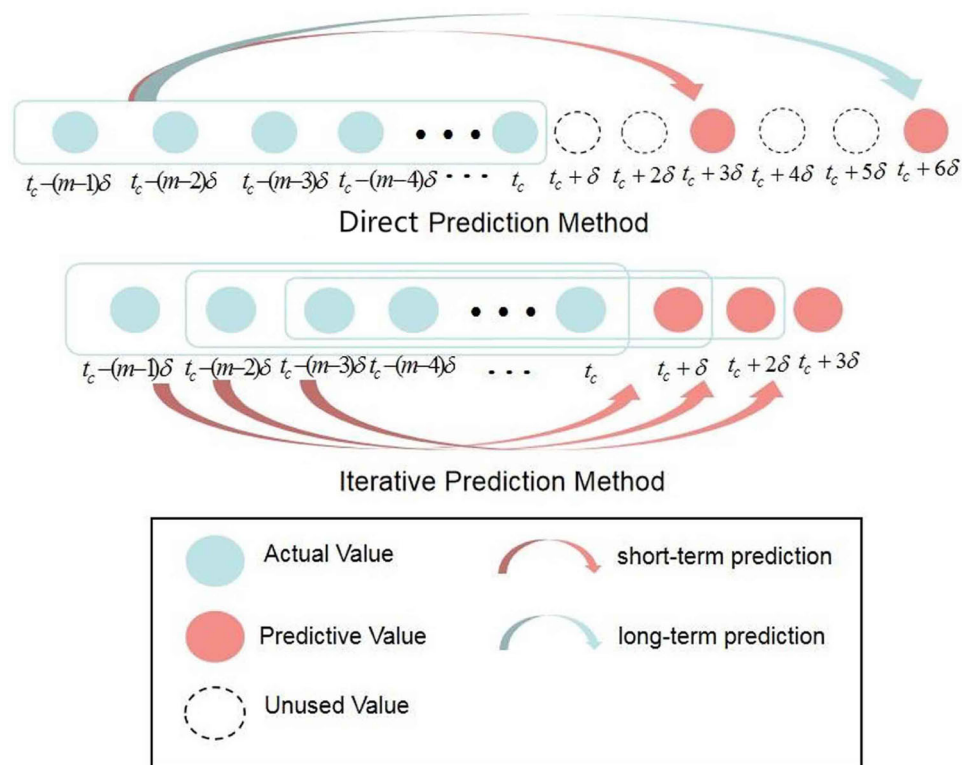
$$\text{SMAPE} = \frac{100\%}{T} \sum_{t=1}^T \frac{\left| y_t^{(h,w)} - \hat{y}_t^{(h,w)} \right|}{\left(\left| \hat{y}_t^{(h,w)} \right| + \left| y_t^{(h,w)} \right| \right) / 2} \tag{9}$$

is used instead of MAPE.

RMSE is the standard deviation of the prediction errors. Formally, it is defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{T} \sum_{t=1}^T \left(y_t^{(h,w)} - \hat{y}_t^{(h,w)} \right)^2}. \tag{10}$$

Fig. 5 Four prediction methods, i.e., short-term direct prediction, long-term direct prediction, short-term iterative prediction, and long-term iterative prediction methods. The red arrows represent the short-term predictions and the blue arrows represent long-term predictions



In summary, the smaller the values of these three indicators, the better the effect of the prediction model.

The ADAM algorithm [25] is used as the gradient descent optimization algorithm, and MAE is used as the loss function. In this study, we use the grid search to determine the optimal values of the hyper-parameters. The fine-tuned hyper-parameter settings of the model are listed as follows: The learning rate is 0.01, the number of historical observations (m) is 10, the epoch size is 32, the batch size is 32, the number of the denselayers in each denseblock (l) is 4, and the number of the ConvLSTM layers (k) is 2. 60% of the data are selected as the training set, 20% are selected as the validation set, and the rest are used as the test set. The numbers of VPSs after 5, 15, 30, 45, and 60 min are predicted, correspondingly. To avoid contingency, each prediction task was independently repeated 50 times and the mean values were taken as the final results. The algorithms were implemented in JetBrains Pycharm Community Edition 2019.2.4 and the performance evaluations were conducted on a 64-bit server with Intel[®] Xeon E5-2620 CPU, 2.10GHz CPU, NVIDIA GeForce GTX 1080 Ti GPU, 64 GB RAM, and a Windows 10 operating system.

Results and analysis

To investigate the effect of prediction. The following four series of experiments were conducted:

- direct predictions with only 5-min scale historical observations;
- iterative predictions with only 5-min scale historical observations;
- direct predictions with both 5-min and daily scale historical observations;
- iterative predictions with both 5-min and daily scale historical observations.

Table 1 shows the detailed results of RMSE, MAE, and MAPE/ SMAPE of the predictions on number of VPSs in St1–St9. In addition, Fig. 6 shows the comparisons between the 5-, 15-, 30-, 45-, and 60-min predictions and actual values, respectively, where x -axis represents the time interval; y -axis represents the number of VPSs.

The results are threefold. First, it can be found that the direct prediction method always outperforms the iterative one, both in experiments with and without daily scale historical observations. Figure 7 summarizes the comparison results between the single-step direct and multi-step iterative predictions. The bar chart shows the number of times that the direct prediction method predicts more accurately than the iterative method in all 36 predictions (i.e., 15-/30-/45-/60-min predictions for 9 parking lots) under different evaluation indicators (i.e., RMSE, MAE, and MAPE). The reason is that the prediction error is accumulated during iterations and results in higher error. In view of this results, only the direct prediction method is considered in the fol-

Table 1 Performance evaluations in terms of MAE, RMSE, and MAPE/SMAPE

| | 5 min | 15 min | 30 min | 45 min | 60 min | 5 min | 15 min | 30 min | 45 min | 60 min | |
|------|--|--------|--------|--------|--------|---|--------|--------|--------|--------|-------|
| MAE | Direct, without daily scale observations | | | | | Iterative, without daily scale observations | | | | | |
| | st1 | 3.40 | 7.18 | 10.95 | 14.08 | 16.65 | – | 6.66 | 10.60 | 16.25 | 20.91 |
| | st2 | 4.01 | 6.86 | 11.04 | 14.33 | 16.95 | – | 7.78 | 13.40 | 22.23 | 26.54 |
| | st3 | 3.70 | 5.62 | 8.62 | 10.99 | 12.44 | – | 6.61 | 11.04 | 17.50 | 18.79 |
| | st4 | 4.00 | 6.15 | 9.27 | 12.03 | 14.46 | – | 7.16 | 12.84 | 19.24 | 24.67 |
| | st5 | 5.23 | 6.77 | 10.50 | 14.06 | 17.25 | – | 8.99 | 14.16 | 21.81 | 25.77 |
| | st6 | 5.46 | 9.59 | 15.85 | 21.12 | 24.38 | – | 10.93 | 20.29 | 31.22 | 37.56 |
| | st7 | 5.38 | 7.92 | 12.02 | 16.13 | 19.30 | – | 9.36 | 16.12 | 22.29 | 25.32 |
| | st8 | 7.46 | 9.16 | 13.31 | 17.10 | 20.50 | – | 10.05 | 16.17 | 26.55 | 32.56 |
| | st9 | 3.51 | 4.09 | 5.58 | 6.95 | 7.94 | – | 4.75 | 7.50 | 10.77 | 16.98 |
| | Direct, with daily scale observations | | | | | Iterative, with daily scale observations | | | | | |
| | st1 | 3.11 | 5.11 | 7.54 | 9.69 | 11.77 | – | 6.38 | 11.41 | 17.67 | 20.90 |
| | st2 | 3.93 | 6.72 | 10.47 | 13.47 | 16.16 | – | 7.56 | 14.36 | 20.60 | 25.82 |
| | st3 | 3.63 | 5.51 | 7.54 | 9.38 | 11.14 | – | 6.49 | 10.51 | 14.77 | 18.36 |
| | st4 | 3.79 | 5.85 | 8.47 | 10.75 | 12.65 | – | 7.01 | 12.07 | 17.86 | 21.77 |
| | st5 | 4.64 | 7.24 | 11.59 | 16.37 | 20.87 | – | 8.66 | 14.28 | 18.67 | 23.92 |
| | st6 | 6.05 | 8.79 | 13.15 | 17.62 | 21.37 | – | 11.06 | 21.03 | 31.29 | 42.13 |
| | st7 | 5.08 | 7.69 | 11.21 | 14.58 | 17.69 | – | 8.82 | 16.37 | 21.57 | 28.10 |
| st8 | 5.53 | 8.41 | 12.23 | 15.67 | 18.96 | – | 9.25 | 15.89 | 23.85 | 30.24 | |
| st9 | 2.46 | 3.86 | 5.28 | 7.11 | 8.00 | – | 4.57 | 7.59 | 10.02 | 12.72 | |
| RMSE | Direct, without daily scale observations | | | | | Iterative, without daily scale observations | | | | | |
| | st1 | 4.62 | 7.18 | 10.95 | 14.08 | 16.65 | – | 8.74 | 14.07 | 21.80 | 28.26 |
| | st2 | 5.54 | 9.61 | 15.47 | 20.12 | 24.00 | – | 11.17 | 19.24 | 32.22 | 37.38 |
| | st3 | 5.06 | 7.88 | 12.07 | 15.68 | 18.22 | – | 9.33 | 15.42 | 25.62 | 29.05 |
| | st4 | 5.48 | 8.51 | 12.94 | 17.18 | 21.04 | – | 9.79 | 17.29 | 26.60 | 34.51 |
| | st5 | 14.15 | 19.99 | 27.71 | 33.62 | 38.85 | – | 24.26 | 31.70 | 40.23 | 46.77 |
| | st6 | 9.91 | 16.13 | 23.38 | 28.86 | 32.50 | – | 17.49 | 28.14 | 42.31 | 50.65 |
| | st7 | 7.57 | 11.14 | 16.91 | 23.14 | 28.67 | – | 12.90 | 22.06 | 31.65 | 36.86 |
| | st8 | 10.04 | 12.76 | 18.94 | 24.39 | 30.17 | – | 13.89 | 23.28 | 38.62 | 47.80 |
| | st9 | 4.60 | 5.41 | 7.29 | 8.99 | 10.37 | – | 6.15 | 9.57 | 13.91 | 21.41 |
| | Direct, with daily scale observations | | | | | Iterative, with daily scale observations | | | | | |
| | st1 | 4.38 | 6.97 | 10.36 | 13.64 | 16.93 | – | 8.45 | 15.16 | 23.50 | 28.20 |
| | st2 | 5.39 | 9.27 | 14.42 | 18.88 | 22.87 | – | 10.45 | 20.86 | 27.69 | 35.58 |
| | st3 | 5.03 | 7.64 | 10.40 | 12.84 | 15.49 | – | 8.91 | 14.56 | 21.14 | 27.30 |
| | st4 | 5.14 | 8.09 | 11.67 | 14.88 | 17.94 | – | 9.41 | 16.12 | 24.12 | 30.04 |
| | st5 | 12.43 | 20.39 | 29.17 | 37.94 | 46.14 | – | 21.84 | 30.53 | 36.15 | 43.58 |
| | st6 | 9.97 | 13.13 | 17.55 | 23.52 | 29.24 | – | 17.39 | 30.50 | 41.79 | 56.12 |
| | st7 | 6.74 | 10.68 | 15.16 | 19.72 | 24.60 | – | 12.08 | 22.77 | 30.73 | 39.75 |
| st8 | 7.56 | 11.77 | 17.19 | 21.84 | 26.32 | – | 12.92 | 22.67 | 34.46 | 43.70 | |
| st9 | 3.39 | 5.16 | 5.28 | 9.49 | 11.07 | – | 5.96 | 9.79 | 13.12 | 16.10 | |

Table 1 continued

| | 5 min | 15 min | 30 min | 45 min | 60 min | 5 min | 15 min | 30 min | 45 min | 60 min |
|------|--|--------|--------|--------|--------|---|--------|--------|--------|--------|
| MAPE | Direct, without daily scale observations | | | | | Iterative, without daily scale observations | | | | |
| st1 | 4.26 | 6.42 | 9.57 | 12.74 | 15.46 | – | 7.16 | 10.80 | 14.73 | 18.51 |
| st2 | 1.06 | 1.84 | 2.97 | 3.88 | 4.62 | – | 2.12 | 3.66 | 6.18 | 7.07 |
| st3 | 5.81 | 7.89 | 10.15 | 11.57 | 13.10 | – | 9.97 | 13.85 | 18.06 | 18.82 |
| st4 | 1.27 | 1.98 | 2.97 | 3.91 | 4.72 | – | 2.28 | 4.00 | 5.84 | 7.49 |
| st5 | 4.10 | 5.10 | 5.25 | 5.53 | 6.34 | – | 9.24 | 9.85 | 11.97 | 12.07 |
| st6 | 1.74 | 2.97 | 4.91 | 6.73 | 7.98 | – | 3.48 | 6.39 | 10.47 | 12.05 |
| st7 | 2.01 | 3.08 | 4.57 | 6.08 | 7.49 | – | 3.54 | 5.92 | 7.50 | 9.23 |
| st8 | 1.82 | 2.51 | 3.58 | 4.43 | 5.16 | – | 2.64 | 4.09 | 6.06 | 7.34 |
| st9 | 2.87 | 3.46 | 4.67 | 5.65 | 6.45 | – | 3.92 | 6.21 | 8.92 | 12.90 |
| | Direct, with daily scale observations | | | | | Iterative, with daily scale observations | | | | |
| st1 | 5.14 | 6.78 | 9.85 | 12.66 | 15.90 | – | 7.94 | 12.98 | 19.89 | 20.33 |
| st2 | 1.05 | 1.80 | 2.83 | 3.75 | 4.60 | – | 2.03 | 3.92 | 5.38 | 6.76 |
| st3 | 6.17 | 8.03 | 10.07 | 10.96 | 12.90 | – | 8.81 | 12.60 | 14.88 | 17.62 |
| st4 | 1.23 | 1.89 | 2.73 | 3.56 | 4.29 | – | 2.23 | 3.81 | 5.68 | 6.78 |
| st5 | 3.90 | 4.43 | 5.19 | 6.33 | 6.80 | – | 5.07 | 5.89 | 7.44 | 8.60 |
| st6 | 1.95 | 2.86 | 4.71 | 6.80 | 8.61 | – | 3.62 | 6.82 | 10.42 | 14.11 |
| st7 | 1.89 | 3.04 | 4.41 | 5.78 | 7.28 | – | 3.40 | 6.37 | 8.28 | 10.25 |
| st8 | 1.53 | 2.37 | 3.39 | 4.22 | 5.01 | – | 2.51 | 3.99 | 5.59 | 6.89 |
| st9 | 2.02 | 3.25 | 4.51 | 5.87 | 6.58 | – | 3.72 | 6.19 | 8.91 | 10.41 |

lowing analysis. Second, for the short-term predictions, the predicted values and the actual values of each parking lot are very close for all the parking lots. However, for the long-term predictions, there appear small gaps. It is reasonable, since long-term predictions are made on the basis of historical observations with much weaker temporal correlations compared with short-term predictions. Third, the daily scale historical observations are beneficial in most prediction tasks. As shown in Fig. 8, for RMSE, MAE, and MAPE, the direct predictions with daily scale historical observations predicted better on 35, 32, and 33 out of 45 tasks, respectively, than those without daily scale historical observations, and in iterative predictions, predictions with daily scale historical observations outperformed those without scale historical observations 25, 25, and 28 times (totally 36 tasks), respectively.

Example 1 To illustrate how well the proposed model can predict the actual number of VPSs, in Table 2, we list the detailed comparisons between the real and predicted numbers of VPSs of St7 (from 14:30 to 15:30, 06/05/2018) output by the proposed dConvLSTM-DCN. We also calculate the corresponding MAEs, MAPEs, and RMSEs. It can be observed that the predicted values output by dConvLSTM-DCN are quit close to the real ones. The mean absolute error (MAE) of 5-min prediction is only 4.31, i.e., 1.54% in percentage term (MAPE), with the standard deviation of the prediction

errors 5.05 (RMSE). Even for 60-min long-term predictions, the MAE is 12.92 (4.63% for MAPE) and the RMSE is 13.96.

Example 2 We test our model on other two parking lots, i.e., the parking lot at Civic Center (Longitude: -118.48997 , Latitude: 34.01158) and Lot 1 North (Longitude: -118.497361 , Latitude: 34.010806), in Santa Monica, California, USA. The time range of the dataset of each parking lot is from 05/09/2018 17:00 to 06/11/2018 21:50. The settings of the hyper-parameters are remain unchanged. For the parking lot at Civic Center, the values of MAPEs of 5-, 15-, 30-, 45-, and 60-min predictions are 1.07%, 1.3%, 2.28%, 3.04%, and 3.95%, respectively, and the values of RMSEs are 7.73, 9.68, 17.63, 23.20, and 30.53, respectively. And for Lot 1 North, the values of MAPEs are 2.11%, 3.72%, 6.34%, 8.14%, and 9.77%, respectively, and the values of RMSEs are 11.02, 16.60, 26.16, 33.52, and 40.12, respectively.

The prediction performance of the dConvLSTM-DCN model was compared with the ConvLSTM-DCN model proposed in [22] and six other mainstream ML/DL models developed in our previous work [17], namely an LSTM model, a gated recurrent units neural network (GRU-NN) model, a stacked autoencoder (SAE) model, a support vector regression (SVR) model, a back propagation neural network (BPNN) model, and a k-nearest neighbor (KNN) model. The values of the hyper-parameters of the comparison models are fine-tuned by grid search. Specifically, the values

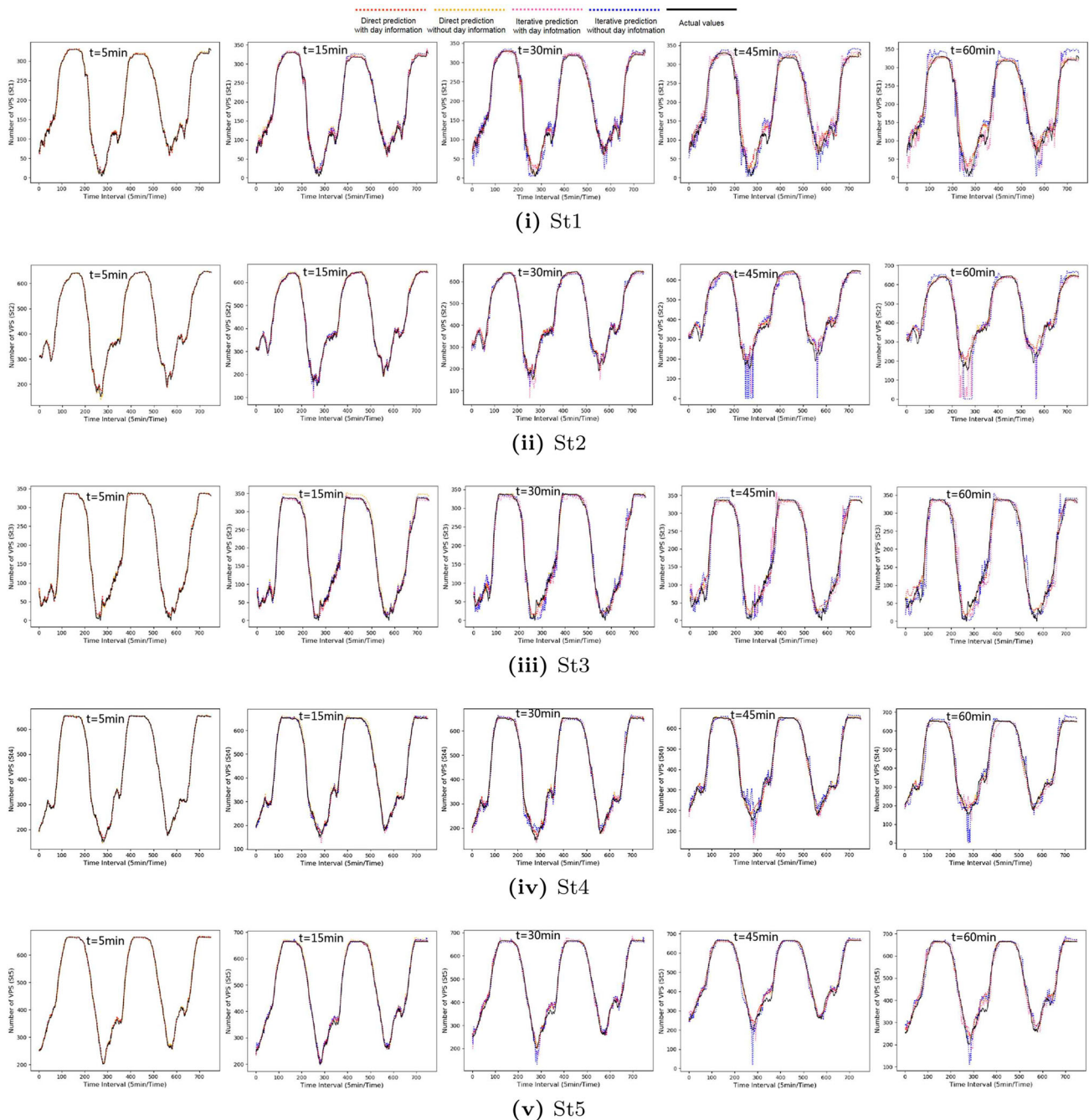


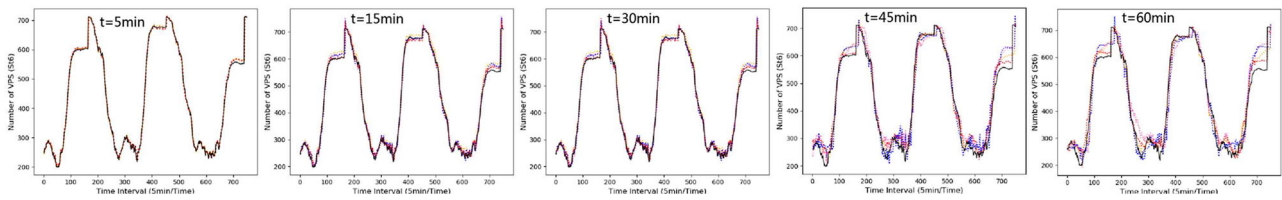
Fig. 6 Comparisons between the 5-, 15-, 30-, 45-, and 60-min predictions made by dConvLSTM-DCN and actual values. Four prediction methods are involved, i.e., direct prediction with day information, direct

prediction without day information, iterative prediction with day information, and iterative prediction without day information

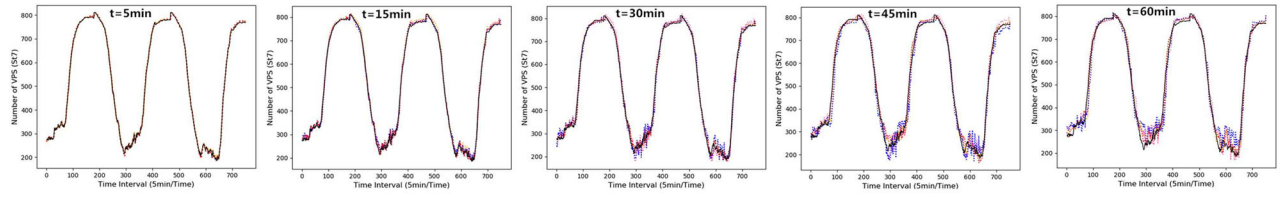
of the hyper-parameters of the ConvLSTM-DCN model are just as the same as those of the dConvLSTM-DCN model, except that the number of the ConvLSTM layers (k) of the ConvLSTM-DCN model is 3. And the hyper-parameter settings of the other six comparison models can be found in [17]. Different from dConvLSTM-DCN and ConvLSTM-DCN that can make zone-wise predictions, the other models

can only predicted the number of VPSs for one parking lot. In this comparative experiment, St7 is considered as the target.

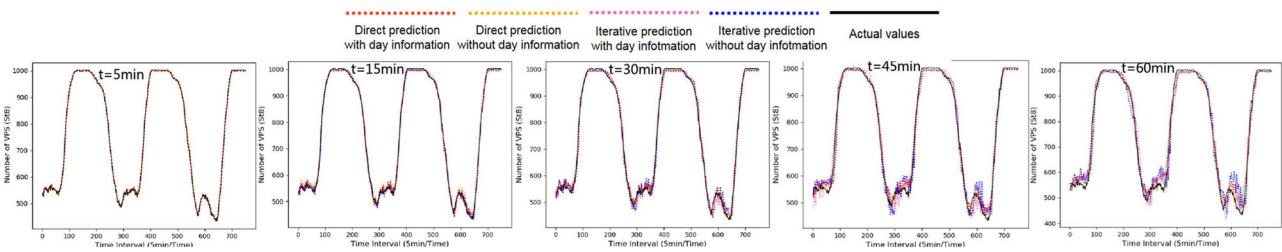
Table 3 and Fig. 9 show the comparisons results on prediction errors. It can be observed that although the accuracy of the dConvLSTM-DCN is slightly lower than that of the LSTM model in 5-min prediction (MAPE, for example, a tiny gap of 0.02%), our method wins in all the other predic-



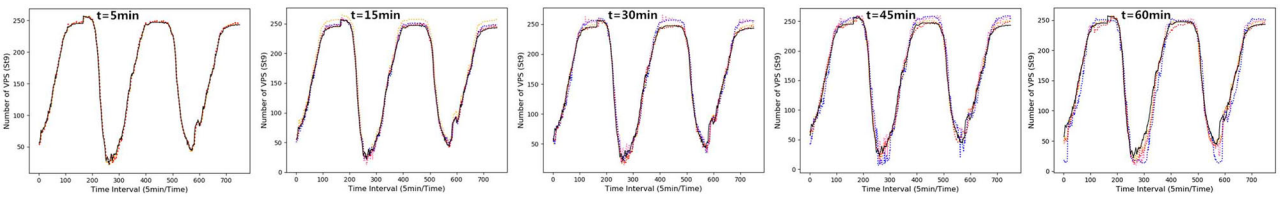
(vi) St6



(vii) St7



(viii) St8



(ix) St9

Fig. 6 continued

Fig. 7 Number of times that the direct prediction method predicts better than the iterative method in 36 predictions (i.e., 15-/30-/45-/60-min predictions for 9 parking lots) under different evaluation indicators (i.e., RMSE, MAE, and MAPE)

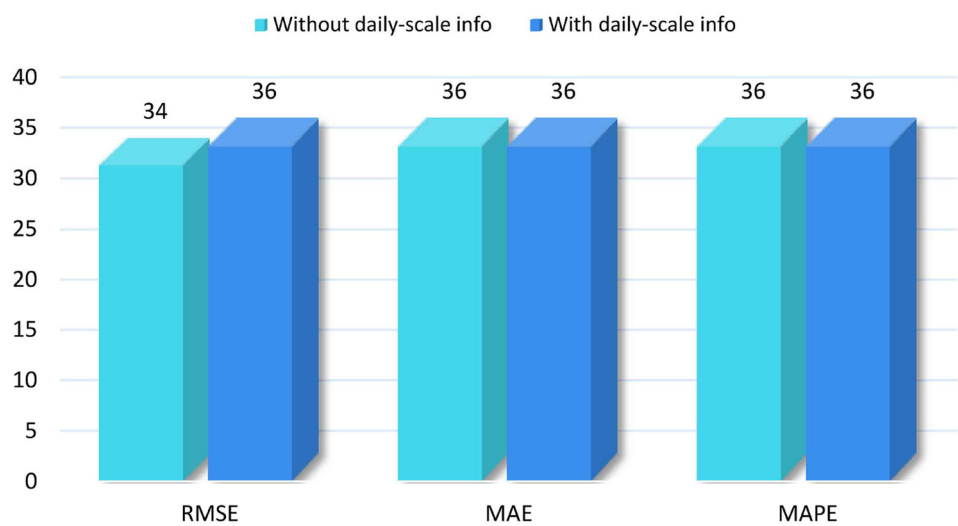
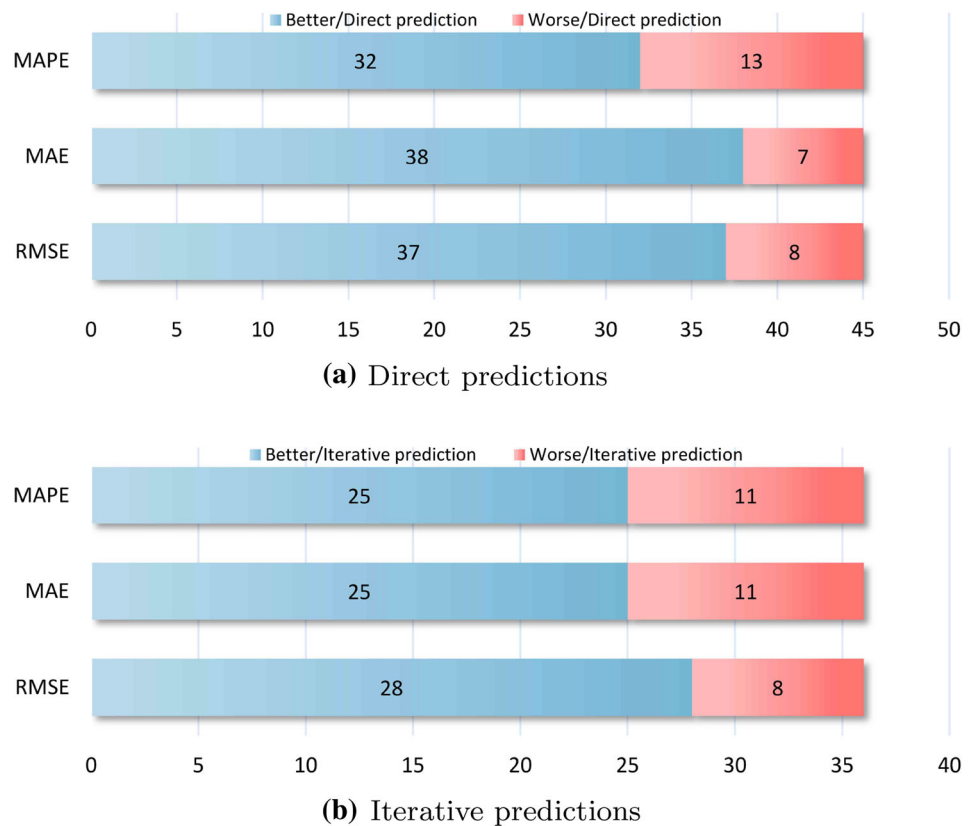


Fig. 8 Number of times that the predictions with daily scale information do better/worse than those without daily scale information under different evaluation indicators (i.e., RMSE, MAE, and MAPE)



tion tasks, namely the 15-min, 30-min, 45-min, and 60-min prediction tasks. Moreover, the longer the predicting term is, the greater advantage the dConvLSTM-DCN framework achieves.

T Tests were used to calculate the p values to investigate whether the prediction errors of dConvLSTM-DCN and the compared models have statistically significant differences. The results are listed in Table 4. It can be seen that all the p values are smaller than 0.05, except the P values between dConvLSTM-DCN and ConvLSTM-DCN in 60-min predictions and between dConvLSTM-DCN and GRU in 15-min predictions, which demonstrates significant differences between them in a statistical sense.

We also compared the training and prediction times of the proposed model with and without GPU, as well as with other models, i.e., ConvLSTM-DCN and LSTM. The experiments were performed on a 64-bit server with an Intel® Xeon E5-2620 CPU and an NVIDIA GeForce GTX 1080 TI GPU. For training time, the experiments were run 30 times independently, and we took the averages as the final results. And for prediction time, the values were averaged from 10,000 independent predictions. The results are shown in Table 5. We can find that with GPUs, the training time required by our model is only 18.66% of that without GPUs. Henceforth, we suggest the use of GPUs due to their significant speed when compared to CPUs. However, considering the much

higher cost of GPUs, CPUs are also acceptable due to their cost savings.

On the other hand, although the LSTM model trains and predicts faster due to a much simpler structure, it can only extract the temporal features of the VPS time series in a single parking lot. By contrast, the ConvLSTM-DCN and dConvLSTM-DCN models extract not only temporal but also spatial features. Besides, they are able to deal with multiple parking lots in an area simultaneously. Therefore, though they consume more time on training and predictions, they can much better learn the spatio-temporal features of the VPS time series and make predictions with much higher accuracy. The dConvLSTM-DCN model uses two similar stacked ConvLSTM structures to capture the spatio-temporal features of two different scales, i.e., the recent historical and daily observations. Naturally, it is more complex than the ConvLSTM-DCN model and more time-consuming; nevertheless, it achieves better prediction performance.

Conclusion

In this paper, we predict the VPS availabilities for all parking lots within a given zone using a hybrid DL model. First, through data analysis, we find that there are strong temporal autocorrelations within the historical VPS data of a given

Table 2 Comparisons between the real and predicted numbers of VPSs of St7 (from 14:30 to 15:30, 06/05/2018)

| Time Points | Real Values | Predicted values | | | | |
|-------------|-------------|------------------|--------|--------|--------|--------|
| | | 5 min | 15 min | 30 min | 45 min | 60 min |
| 14:30 | 277 | 273 | 272 | 282 | 293 | 290 |
| 14:35 | 277 | 276 | 273 | 287 | 295 | 287 |
| 14:40 | 278 | 277 | 273 | 287 | 295 | 286 |
| 14:45 | 272 | 278 | 284 | 289 | 290 | 280 |
| 14:50 | 284 | 275 | 285 | 285 | 286 | 269 |
| 14:55 | 281 | 282 | 284 | 296 | 284 | 268 |
| 15:00 | 285 | 282 | 285 | 295 | 284 | 270 |
| 15:05 | 284 | 286 | 288 | 297 | 286 | 275 |
| 15:10 | 281 | 286 | 284 | 296 | 289 | 287 |
| 15:15 | 275 | 283 | 297 | 294 | 290 | 290 |
| 15:20 | 284 | 278 | 297 | 283 | 293 | 295 |
| 15:25 | 279 | 282 | 300 | 296 | 295 | 297 |
| 15:30 | 274 | 281 | 298 | 288 | 297 | 301 |
| MAE | | 4.31 | 9.00 | 11.23 | 11.38 | 12.92 |
| MAPE | | 1.54% | 3.25% | 4.04% | 4.11% | 4.63% |
| RMSE | | 5.05 | 12.14 | 12.59 | 13.49 | 13.96 |

Table 3 Comparisons of prediction errors

| Models | Indicators | 5 min | 15 min | 30 min | 45 min | 60 min |
|--------------|------------|-------------|--------------|--------------|--------------|--------------|
| dConvLSTM-DN | RMSE | 6.74 | 10.68 | 15.16 | 19.72 | 24.60 |
| | MAE | 5.08 | 7.69 | 11.21 | 14.58 | 17.69 |
| | MAPE (%) | 1.89 | 3.04 | 4.41 | 5.78 | 7.28 |
| ConvLSTM-DN | RMSE | 7.57 | 11.14 | 16.91 | 23.14 | 28.67 |
| | MAE | 5.38 | 7.92 | 12.02 | 16.13 | 19.30 |
| | MAPE (%) | 2.01 | 3.08 | 4.57 | 6.08 | 7.49 |
| LSTM | RMSE | 5.77 | 11.35 | 18.82 | 27.11 | 36.13 |
| | MAE | 3.96 | 7.96 | 13.47 | 19.38 | 25.78 |
| | MAPE (%) | 1.87 | 3.74 | 6.09 | 8.24 | 10.39 |
| GRU-NN | RMSE | 5.83 | 11.57 | 19.40 | 27.85 | 36.92 |
| | MAE | 4.01 | 8.02 | 13.79 | 19.45 | 26.93 |
| | MAPE (%) | 1.92 | 3.72 | 6.36 | 8.69 | 11.61 |
| SAE | RMSE | 7.24 | 12.67 | 19.84 | 27.73 | 36.70 |
| | MAE | 5.02 | 8.92 | 14.04 | 18.67 | 26.58 |
| | MAPE (%) | 2.71 | 4.25 | 6.84 | 9.39 | 11.88 |
| SVR | RMSE | 12.55 | 14.41 | 20.08 | 28.69 | 38.14 |
| | MAE | 10.12 | 10.44 | 14.94 | 21.58 | 28.68 |
| | MAPE (%) | 3.50 | 5.40 | 6.75 | 9.15 | 11.66 |
| BPNN | RMSE | 5.81 | 11.63 | 19.62 | 28.37 | 37.91 |
| | MAE | 3.98 | 8.17 | 14.22 | 20.81 | 27.82 |
| | MAPE (%) | 1.88 | 3.79 | 6.34 | 8.88 | 11.38 |
| KNN | RMSE | 6.66 | 12.99 | 21.19 | 29.87 | 39.20 |
| | MAE | 4.59 | 8.90 | 14.21 | 19.50 | 24.47 |
| | MAPE (%) | 2.12 | 4.32 | 7.24 | 9.74 | 12.00 |

The values in bold indicate the best performances

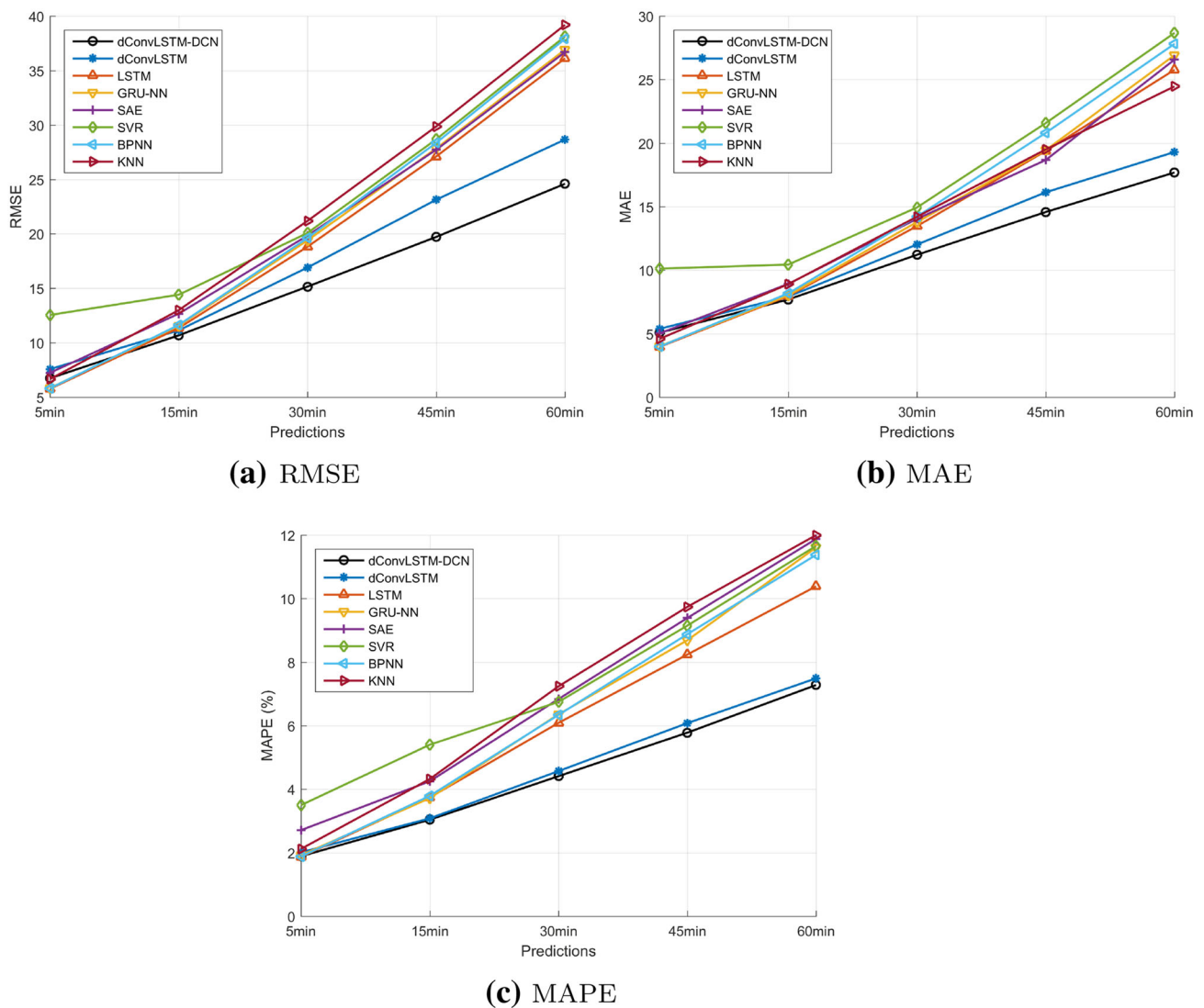


Fig. 9 Comparisons of the predictions of the number of VPSs in St7 between dConvLSTM-DCN and comparison models under different evaluation indicators (i.e., RMSE, MAE, and MAPE)

Table 4 *P* values between dConvLSTM-DCN and comparison models calculated by *T*-test

| Comparison Models | <i>P</i> -values | | | | |
|-------------------|------------------|----------|----------|----------|----------|
| | 5 min | 15 min | 30 min | 45 min | 60 min |
| ConvLSTM | 6.01E−11 | 4.03E−95 | 2.35E−03 | 2.27E−03 | 7.73E−01 |
| LSTM | 2.91E−05 | 3.45E−28 | 2.34E−30 | 4.19E−07 | 7.06E−47 |
| GRU | 5.76E−17 | 1.79E−01 | 2.64E−45 | 1.78E−05 | 7.26E−34 |
| SAE | 5.85E−03 | 2.65E−05 | 1.45E−14 | 1.50E−02 | 4.05E−27 |
| SVR | 5.26E−98 | 5.00E−23 | 6.81E−30 | 7.16E−18 | 3.51E−50 |
| BPNN | 2.03E−16 | 3.80E−02 | 1.02E−21 | 3.69E−14 | 1.49E−45 |
| KNN | 1.04E−05 | 5.11E−06 | 7.43E−18 | 2.66E−05 | 4.03E−14 |

Table 5 Runtime comparisons among different models

| Models | Training time (s/epoch) | Prediction time (μ s/Prediction) |
|--------------------|-------------------------|---------------------------------------|
| dConvLSTM-DCN | 254.89 | 15.05 |
| dConvLSTM-DCN(GPU) | 47.56 | 3.86 |
| ConvLSTM_DCN(GPU) | 28.36 | 1.62 |
| LSTM(GPU) | 1.19 | 0.93 |

parking lot. And there also exists apparent spatial correlations between different parking lots. In view of this, we develop the dConvLSTM-DCN framework which consists of two parallel ConvLSTM components and a DCN to fully utilize the spatial-temporal correlations in historical VPS data to predict the VPS availability in 5–60 min of all parking lots within the target zone. Both the 5-min scale and daily-wise historical observations are fed into the dConvLSTM model to capture the recent and daily spatial-temporal correlations, and the information on hour of day and day of week are as well used to promote the prediction accuracy. Moreover, two prediction methods, namely, the single-step direct and multi-step iterative prediction methods, are proposed to make long-term predictions.

Sufficient experiments have been conducted using real-world data collected from 9 public parking lots in Santa Monica, California, USA. The experiment results show that our model can achieved considerably high accuracy with MAPEs of lower than 5% in short-term predictions and lower than 8% in long-term predictions, respectively. We also compared the dConvLSTM-DCN-based prediction method with the ConvLSTM-DCN-based method proposed in [22] and the sufficiently fine-tuned LSTM-based method presented in [17]. The results demonstrate that the dConvLSTM-DCN framework is superior to others, especially in long-term predictions.

Acknowledgements This study was funded by fundamental scientific research project of Wenzhou city (Grant number G20190021) and Natural Science Foundation of Zhejiang Province (Grant number LZ20F010008).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Shoup DC (2006) Cruising for parking. *Transport Policy* 13(6):479–486
- Monteiro FV, Ioannou P (2018) On-street parking prediction using real-time data. In: 2018 21st International conference on intelligent transportation systems (ITSC), pp 2478–2483. <https://doi.org/10.1109/ITSC.2018.8569921>
- Giuffrè T, Siniscalchi SM, Tesoriere G (2012) A novel architecture of parking management for smart cities. *Proc Soc Behav Sci* 53:16–28. <https://doi.org/10.1016/j.sbspro.2012.09.856>
- Hong TP, Soh AC, Jaafar H, Ishak AJ (2013) Real-time monitoring system for parking space management services. In: 2013 IEEE conference on systems, process and control (ICSPC), pp 149–153
- Yang CF, Ju YH, Hsieh CY, Lin CY, Tsai MH, Chang HL (2017) iParking—a real-time parking space monitoring and guiding system. *Veh Commun* 9:301–305
- Amato G, Carrara F, Falchi F, Gennaro C, Meghini C, Vairo C (2017) Deep learning for decentralized parking lot occupancy detection. *Expert Syst Appl* 72:327–334
- Hassija V, Saxena V, Chamola V, Yu FR (2020) A parking slot allocation framework based on virtual voting and adaptive pricing algorithm. *IEEE Trans Veh Technol* 69(6):5945–5957. <https://doi.org/10.1109/TVT.2020.2979637>
- Nie Y, Yang W, Chen Z, Lu N, Huang L, Huang H (2021) Public curb parking demand estimation with poi distribution. *IEEE Trans Intell Transp Syst*. <https://doi.org/10.1109/TITS.2020.3046841>
- Caliskan M, Barthels A, Scheuermann B, Mauve M (2007) Predicting parking lot occupancy in vehicular ad hoc networks. In: 2007 IEEE 65th vehicular technology conference—VTC2007-Spring, pp 277–281. <https://doi.org/10.1109/VETECS.2007.69>
- Xiao J, Lou Y, Frisby J (2018) How likely am i to find parking?—a practical model-based framework for predicting parking availability. *Transp Res Part B Methodol* 112:19–39. <https://doi.org/10.1016/j.trb.2018.04.001>
- Caicedo F, Blazquez C, Miranda P (2012) Prediction of parking space availability in real time. *Expert Syst Appl* 39(8):7281–7290. <https://doi.org/10.1016/j.eswa.2012.01.091>
- Yu F, Guo J, Zhu X, Shi G (2015) Real time prediction of unoccupied parking space using time series model. In: 2015 International conference on transportation information and safety (ICTIS), pp 370–374. <https://doi.org/10.1109/ICTIS.2015.7232145>
- Zheng Y, Rajasegarar S, Leckie C (2015) Parking availability prediction for sensor-enabled car parks in smart cities. In: 2015 IEEE tenth international conference on intelligent sensors, sensor networks and information processing (ISSNIP), pp 1–6. <https://doi.org/10.1109/ISSNIP.2015.7106902>
- Ji Y, Tang D, Blythe P, Guo W, Wang W (2015) Short-term forecasting of available parking space using wavelet neural network model. *IET Intell Transport Syst* 9(2):202–209. <https://doi.org/10.1049/iet-its.2013.0184>
- Badii C, Nesi P, Paoli I (2018) Predicting available parking slots on critical and regular services by exploiting a range of open data. *IEEE Access* 6:44059–44071. <https://doi.org/10.1109/ACCESS.2018.2864157>

16. Fan J, Hu Q, Tang Z (2018) Predicting vacant parking space availability: an svr method with fruit fly optimisation. *IET Intell Transport Syst* 12(10):1414–1420. <https://doi.org/10.1049/iet-its.2018.5031>
17. Fan J, Hu Q, Xu Y, Tang Z (2020) Predicting the vacant parking space availability: an lstm approach. *IEEE Intell Transp Syst Mag.* <https://doi.org/10.1109/MITS.2020.3014131>
18. Rong Y, Xu Z, Yan R, Ma X (2018) Du-parking: spatio-temporal big data tells you realtime parking availability. *Assoc Comput Mach.* <https://doi.org/10.1145/3219819.3219876>
19. Rajabioun T, Ioannou PA (2015) On-street and off-street parking availability prediction using multivariate spatiotemporal models. *IEEE Trans Intell Transp Syst* 16(5):2913–2924. <https://doi.org/10.1109/TITS.2015.2428705>
20. Shi X, Chen Z, Wang H, Yeung DY, Wong Wk, Woo Wc (2015) Convolutional lstm network: a machine learning approach for precipitation nowcasting. In: *Proceedings of the 28th international conference on neural information processing systems—vol 1*, MIT Press, Cambridge, MA, USA, NIPS'15, pp 802–810
21. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp 2261–2269. <https://doi.org/10.1109/CVPR.2017.243>
22. Feng Y, Tang Z, Xu Y, Krishnamoorthy S, Hu Q (2021) Predicting vacant parking space availability zone-wisely: a densely connected ConvLSTM approach. In: *IEEE vehicular power and propulsion 2021*
23. (2020) Santa monica open data portal. <https://data.smgov.net/>. Accessed July 2018
24. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
25. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: *3rd International conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, conference track proceedings*

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.