Predicting Vacant Parking Space Availability Zone-Wisely: A Graph Based Spatio-Temporal Prediction Approach

Yajing Feng, Zhenzhou Tang, Senior Member, IEEE, Yingying Xu, and Qian Hu

Abstract—Intelligent parking guidance systems (IPGSs) based on vacant parking space (VPS) availability predictions are highly effective to alleviate the increasingly serious parking difficulties in metropolises. The spatial-temporal correlation analysis of VPS information of multiple parking lots in a region shows that not only the number of VPSs in each parking lot has a stable temporal correlation, but also there is obvious spatial correlation among different parking lots. Given this, this article takes full advantage of the space-time correlation and makes short-term (within 30 min) and long-term (over 30 min) predictions about the number of VPSs in multiple parking lots in an area parallelly. Specifically, this article develops a deep gated graph recurrent neural network (G$_2$RNN) model which has the ability to extract both spatial and temporal correlations concurrently between different parking lots. The advantages of the G$_2$RNN model are that: first, spatial and temporal correlations between different parking lots can be accurately obtained simultaneously, and second, there is no need to grid the raw data since it uses graph-structured data. For long-term predictions, two different approaches, namely the single-step and iterative predictions, are investigated. The performance of the G$_2$RNN based prediction method is extensively evaluated with practical data collected from eight public parking lots in Santa Monica. The results show that it can achieve considerably high accuracy in both short-term and long-term predictions.

Index Terms—Vacant parking space availability, deep learning, GCN, GRU.

I. INTRODUCTION

Parking is always a big problem for drivers in busy cities, as the growth rate of vehicles is much faster than that of parking spaces. Parking problem leads to cruising, i.e. vehicles looking for a parking space, causing long queues, congestion, and pollution [1], [2], [3], [4]. In order to alleviate the parking problem, many cities are actively carrying out the construction of intelligent parking guidance system (IPGS).

Currently, the most widely used IPGSs are cyber-physical systems [5] that come equipped with sensors, video surveillance, and electronic gate systems. These systems provide real-time information on the location of parking lots and the number of vacant parking spaces (VPSs), helping drivers to quickly locate available parking and alleviate parking problems. However, since the number of VPS is always changing, a great gap may appear between the number of VPSs known to the driver and the actual number when the driver arrives at the destination parking lot. Even worse, the driver may arrive at the destination parking lot with no parking space left.

The ongoing advancements in artificial intelligence and the Internet of Things [6] have led to increased interest in the VPS prediction technique, which utilizes real-time parking information to forecast future VPS availability. The VPS prediction technique can avoid such the above awkward situation and, on the other hand, can effectively direct drivers to parking lots with more VPSs. Therefore, it has become a hot research topic of IPGS. VPS prediction is a typical time series prediction problem. At present, the common methods can be mainly divided into three categories: the traditional prediction models, the machine learning models, and the deep learning models.

The traditional prediction models generally involve historical average models (HA) [7], moving average (MA) models [8], auto-regressive (AR) model [9], auto-regressive moving average (ARMA) model [10], and auto-regressive integrated moving average (ARIMA) model [11], etc. However, it should be noted that these methods require the input time series data to be stationary. Moreover, these methods can only extract linear relationship in the sequence, but not nonlinear relationship.

Comparatively, prediction models based on machine learning have been put forward one after another, and show better performance than traditional time series prediction methods. Combining the support vector regression (SVR) model with the fruit fly optimization algorithm (FOA), J. Fan et al. [12] proposed a new prediction model to predict the number of VPSs in a single parking lot. D. Chuan et al. [13] used the extreme gradient boosting (XGB) model to predict the short-team subway ridership. This study showed that XGB model outperformed...
the random forest model in prediction accuracy and efficiency. X. Ye [14] proposed a VPS prediction algorithm based on the gradient boosting decision tree (GBDT) and wavelet neural network (WNN). Habtemuchael et al. [15] built a k-nearest neighbor (KNN) model for short-time traffic flow prediction. Compared with traditional time series models, prediction methods based on machine learning do not require time series data to satisfy the stationarity assumption. However, the prediction accuracy of these machine learning models still needs to be improved, especially in the long-term predictions.

Consequently, prediction techniques based on deep learning models have been naturally proposed. Among them, the recurrent neural networks (RNNs), such as long short term memory (LSTM), gated recurrent unit (GRU), etc. have been widely. J. Fan [16] built an LSTM model to predict the VPS availability for a single parking lot. C. Zeng [17] proposed a stacked GRU-LSTM model for VPS prediction. Y. Du [18] used an RNN model for vehicle speed prediction.

However, the above models are only able to make single-point predictions. In other words, they only made use of the temporal correlation of the historical observations in the single target parking lot and can only make prediction for this single parking lot. Actually, it has been proved that there is spatial correlation between different parking lots, which can be exploited to not only improve prediction accuracy, but also make multi-point predictions, that is, concurrent predictions for multiple nearby parking lots [19], [20]. In [19] and [20], ConvLSTM was used to extract the spatio-temporal correlation between different parking lots. The results showed that the ConvLSTM based method can make both short-term and long-term predictions with high accuracy. However, due to the inherent requirement of convolutional neural networks, the historical data fed into the above models had to be reasonably gridded, which may be quite difficult to be satisfied in many practical scenarios. More importantly, the spatial-temporal correlation obtained by using gridded data is grid-level, but not the exact spatial-temporal correlation between parking lots. This may degrade the prediction accuracy.

Comparatively, the graph convolutional network (GCN) using graph-structured data is more competitive for such application scenarios, since it extracts the spatial correlation exactly between parking lots without gridding the raw data. GCN-based models have been widely used, such as the global spatio-temporal network (GSTNet) [21], attention based spatial-temporal graph convolutional network (ASTGCN) [22], diffusion convolutional recurrent neural network (DCRNN) [23], [24], temporal multi graph convolutional network (T-MGCN) [25], etc. In general, most existing models separately extract temporal and spatial correlations by RNN and GCN, respectively, rather than parallelly. This results in insufficient feature extraction of hidden layer output, which negatively affects the prediction accuracy of the model.

Motivated by the above, this article first puts forward a deep gated graph recurrent neural network (G²RNN) model which is an integration of GCN and GRU. The most notable advantages of G²RNN are two-fold. First, different from most existing models that extract spatial and temporal correlations between different parking lots separately, G²RNN can do this in parallel and fully extract the features of the hidden layer output. Second, there is no need to grid the raw data since it uses graph-structured data, which determines that the model can get the precise spatio-temporal correlation between different parking lots.

Consequently, prediction techniques based on deep learning models have been naturally proposed. Among them, the recurrent neural networks (RNNs), such as long short term memory (LSTM), gated recurrent unit (GRU), etc. have been widely used. J. Fan [16] built an LSTM model to predict the VPS availability for a single parking lot. C. Zeng [17] proposed a stacked GRU-LSTM model for VPS prediction. Y. Du [18] used an RNN model for vehicle speed prediction.

However, the above models are only able to make single-point predictions. In other words, they only made use of the temporal correlation of the historical observations in the single target parking lot and can only make prediction for this single parking lot. Actually, it has been proved that there is spatial correlation between different parking lots, which can be exploited to not only improve prediction accuracy, but also make multi-point predictions, that is, concurrent predictions for multiple nearby parking lots [19], [20]. In [19] and [20], ConvLSTM was used to extract the spatio-temporal correlation between different parking lots. The results showed that the ConvLSTM based method can make both short-term and long-term predictions with high accuracy. However, due to the inherent requirement of convolutional neural networks, the historical data fed into the above models had to be reasonably gridded, which may be quite difficult to be satisfied in many practical scenarios. More importantly, the spatial-temporal correlation obtained by using gridded data is grid-level, but not the exact spatial-temporal correlation between parking lots. This may degrade the prediction accuracy.

Comparatively, the graph convolutional network (GCN) using graph-structured data is more competitive for such application scenarios, since it extracts the spatial correlation exactly between parking lots without gridding the raw data. GCN-based models have been widely used, such as the global spatio-temporal network (GSTNet) [21], attention based spatial-temporal graph convolutional network (ASTGCN) [22], diffusion convolutional recurrent neural network (DCRNN) [23], [24], temporal multi graph convolutional network (T-MGCN) [25], etc. In general, most existing models separately extract temporal and spatial correlations by RNN and GCN, respectively, rather than parallelly. This results in insufficient feature extraction of hidden layer output, which negatively affects the prediction accuracy of the model.

Motivated by the above, this article first puts forward a deep gated graph recurrent neural network (G²RNN) model which is an integration of GCN and GRU. The most notable advantages of G²RNN are two-fold. First, different from most existing models that extract spatial and temporal correlations between different parking lots separately, G²RNN can do this in parallel and fully extract the features of the hidden layer output. Second, there is no need to grid the raw data since it uses graph-structured data, which determines that the model can get the precise spatio-temporal correlation between different parking lots. The novel G²RNN model is then applied to make short- and long-term predictions of the VPS availability for multiple parking lots in an area. For long-term predictions, two different approaches, namely the single-step and iterative predictions, are investigated. Sufficient comparative experimental results show that the G²RNN model proposed in this article can achieve good prediction accuracy and has a good application prospect. The contributions of this article are shown below.

This work builds upon our previous publication [19] that was presented at VPPC 2021. However, this current work presents significant and novel contributions that go beyond our earlier work. To summarize, our new contributions are as follows:

- In [19], the spatial-temporal correlation between various parking lots was obtained using gridded data. However, this approach only provided correlation at the grid-level, which may result in lower prediction accuracy due to the lack of precise spatial-temporal correlation between the parking lots. In contrast, the current work converts the raw VPS data into graph-structured data, which can be fed into GCNs to extract the precise spatial-temporal correlation between the different parking lots. This approach is expected to improve prediction accuracy compared to the previous method.

- The most remarkable new contribution is that we developed the novel G²RNN model which is an integration of GCN and GRU to simultaneously make predictions for all parking lots within the target area. G²RNN uses graph-structured data so that it can extract the real spatio-temporal correlations in parallel behind the historical VPS information between different parking lots. The implementation of G²RNN model is available at: https://github.com/iNetWZU/G²RNN.

- Sufficient comparative experiments have been conducted to evaluate the effectiveness of the proposed prediction method. The results show that the G²RNN model outperforms the ConvLSTM-DCN model presented in [19] in terms of both accuracy and runtime, and can make both short-term predictions and long-term predictions with considerably high accuracy.

The rest of this article is organized as follows: Section II introduces the proposed G²RNN model and the VPS availability prediction method based on it in detail. In Section III, the comparative experimental results and analysis are carried out. Finally, Section IV concludes this article.

II. METHODOLOGY

A. Problem Definition

VPS prediction is a typical time series prediction problem, i.e., given m historical observations on the number of VPSs of N(N ≥ 1) parking lots at time steps
is the number of steps, is predicted. Generally, predictions of \( h \) historical observations are used for \( h > h_\delta > 1 \), two approaches have been proposed: the direct prediction approach, and the iterative prediction approach.

\[
[t_c - (m - 1)\delta, t_c - (m - 2)\delta, \ldots, t_c], 
\]

denoted as:

\[
X_{t_c} = \left[ X_{t_c - (m-1)\delta}, X_{t_c - (m-2)\delta}, \ldots, X_{t_c} \right] = 
\begin{bmatrix}
X_{1, t_c - (m-1)\delta} & X_{1, t_c - (m-2)\delta} & \cdots & X_{1, t_c} \\
X_{2, t_c - (m-1)\delta} & X_{2, t_c - (m-2)\delta} & \cdots & X_{2, t_c} \\
\vdots & \vdots & \ddots & \vdots \\
X_{N, t_c - (m-1)\delta} & X_{N, t_c - (m-2)\delta} & \cdots & X_{N, t_c}
\end{bmatrix},
\]  

(1)

where \( X_t = [X_t^1, X_t^2, \ldots, X_t^N]^T \), \( X_{t,i} \), \( i = 1, \ldots, N \) is the number of VPSs of the \( i \)th parking lot at time \( t \), \( \delta \) is the step size and \( t_c \) is the current time. The number of VPSs at \( t_c + h\delta \) in the future, denoted as

\[
\hat{Y}_{t_c + h\delta} = [Y_{t_c + h\delta}^1, Y_{t_c + h\delta}^2, \ldots, Y_{t_c + h\delta}^N]^T,
\]  

(2)

where \( h \) is the number of steps, is predicted. Generally, predictions with \( N > 1 \) are multi-point predictions, and those with \( N = 1 \) are single-point predictions. Moreover, predictions with \( h\delta \leq 30 \) minutes are called short-term prediction, while those with \( h\delta > 30 \) minutes are called long-term predictions.

For predictions where \( h > 1 \), two approaches have been suggested, namely the single-step direct prediction and multi-step iterative prediction [16]. Specifically, in single-step direct prediction approach, \( m \) historical observations are used to directly predict \( \hat{Y}_{t_c + h\delta} \), as shown in the first part of Fig. 1. By contrast, in the multi-step iterative prediction approach, \( \hat{Y}_{t_c + \Delta} \) is firstly obtained with \( m \) historical observations \([X_{t_c -(m-1)\delta}, X_{t_c -(m-2)\delta}, \ldots, X_{t_c}]\). After that, \([X_{t_c -(m-2)\delta}, \ldots, X_{t_c}, \hat{Y}_{t_c + \Delta}]\) are input to predict \( \hat{Y}_{t_c + 2\Delta} \), \([X_{t_c -(m-3)\delta}, \ldots, X_{t_c}, \hat{Y}_{t_c + \Delta}, \hat{Y}_{t_c + 2\Delta}]\) are input to iteratively predict \( \hat{Y}_{t_c + 3\Delta} \), and so on, until \( \hat{Y}_{t_c + h\delta} \) is obtained, as shown in the second part of Fig. 1.

### B. Data Description & Preprocessing

The dataset come from 8 parking lots (St1-St8 in Fig. 2) in Santa Monica (longitude range is [−118.499378, −118.49372], latitude range is [34.019575, 34.01289]), California, USA [26]. Specifically, for each parking lot, the number of VPSs was collected every 5 minutes, from 05/11/2018 7:00 to 06/11/2018 23:55, so that each parking lot has 9108 items.

We performed the spatio-temporal correlations analysis between parking lots by the method in our previous work [19], [20]. Fig. 3(a) and (b) are the temporal correlation diagrams of parking lot St1, and Fig. 3(c) shows the spatial correlation between parking lots St1-St8. It can be seen from the figure that

Fig. 1. Four prediction methods, i.e. short-term direct prediction, long-term direct prediction, short-term iterative prediction and short-term direct prediction methods. The green arrows represent the short-term predictions and the blue arrows represent long-term predictions.

Fig. 2. Distribution of the parking lots.
Fig. 3. Spatial and temporal dynamics of parking space. (a) Number of VPS in time series. (b) Temporal correlations of number of VPS in time series. (c) Spatial correlations (Pearson correlations) between different parking lots.

there is obvious temporal correlation within each parking lot and spatial correlation between different parking lots.

The preprocessing of the raw dataset includes the following procedures:

1) Build a graph $G$ based on the geographic locations of the parking lots. $G = (V, E, A)$ where $V \in \mathbb{N}^N$ represents the set of nodes (i.e., parking lots), $N$ represents the number of nodes, $E \in \mathbb{R}^{N \times N}$ represents the set of edges; $A \in \mathbb{R}^{N \times N}$ represents the adjacency matrix whose element $A_{ij}$ can be determined by

$$A_{ij} = \begin{cases} d_{ij}, & d_{ij} \leq \varepsilon, \\ 0, & d_{ij} > \varepsilon, \end{cases}$$

where

$$d_{ij} = 2R \arcsin \sqrt{\sin^2(a) + \cos(lat_i) \cos(lat_j) \sin^2(b)},$$

$R$ is the radius of the earth; $lat_i$ and $long_i$ are the longitude and latitude of the parking lot $i$, respectively,

$$a = \left(\frac{lat_j - lat_i}{2}\right), \quad b = \left(\frac{long_j - long_i}{2}\right),$$

and $\varepsilon$ is the distance threshold with the typical value of 0.35.

2) Use the min-max normalization method to normalize all the data to the range $[0,1]$;

3) Use the sliding window method to intercept the dataset, and reshape the data into the form of (1).

4) Split the preprocessed dataset into training set and test set.

C. $G^2$RNN

In this section, we describe the structure of the $G^2$RNN model which is an integration of GCNs and GRU in detail. The overview of the $G^2$RNN model is depicted in Fig. 4, where the left part shows the prediction process of the model, and the right part shows the specific structure of the $G^2$RNN model.

As depicted in Fig. 4, the model has an input layer which receives the historical number of VPSs of multiple parking lots in an area ($X_t$ given by (1)) and the adjacency matrix of the corresponding graph model ($A$ calculated by (3)). The inputs
are fed into the spatial-temporal processing layer consisting of $m$ $G^2$RNN cells.

A $G^2$RNN cell is a synthesis of graph convolutional neural network and gated recurrent unit, rather than a simple concatenation of them. Specifically, $G^2$RNN introduces a two-layer GCN ($G_t$) after the input $X_t$ and the previous output $H_{t-1}$, respectively, to extract the spatial correlation between different parking lots. The structure of the two-layer GCN is shown in Fig. 5. The first convolutional layer applies a linear transformation to each node’s feature vector based on the feature vectors of its neighbors. Specifically, the feature vector of each node is updated by aggregating the feature vectors of its neighbors and applying a linear transformation to the resulting vector. This operation is applied to all nodes in the graph, resulting in a new set of feature vectors for each node. The operation of the first layer is

$$o_1 = \tilde{D}^{-\frac{1}{2}}A\tilde{D}^{-\frac{1}{2}}XW_0,$$

where $A$ is the adjacency matrix of the graph $G$, $W_0$ is the weight vector, $\tilde{D}$ is the degree matrix in the form of diagonal matrix where $\tilde{D}_{ii} = \sum_j A_{ij}$.

After the first convolutional layer, a ReLU activation function is applied to the feature vectors of each node. This function introduces non-linearity into the network, allowing it to learn more complex representations of the graph data and make accurate predictions.

The second convolutional layer applies a similar operation as the first convolutional layer to the updated feature vectors. Specifically, the feature vector of each node is updated by aggregating the feature vectors of its neighbors, but this time based on the updated feature vectors from the previous layer. This operation is also applied to all nodes in the graph, resulting in a new set of feature vectors for each node. After the second convolutional layer, the sigmoid activation function $\sigma(\cdot)$ is applied to the feature vectors of each node. The detail operation is as follows:

$$G'(X_t, A) = \sigma\left(\tilde{D}^{-\frac{1}{2}}A\tilde{D}^{-\frac{1}{2}}\text{ReLU}(o_1)W_1\right).$$

On the other hand, the gated recurrent component of $G^2$RNN is leveraged to capture the temporal dependencies between sequential data. Difference from the traditional GRU that takes $X_t$ as the direct input, the $G^2$RNN cell takes the output of the GCNs as the input of the gated recurrent component, so that it can leverage the ability of the GCNs to capture spatial information in the graph-structured data and then use the GRU to capture the temporal dependencies in the data.

It also includes an update gate $z_t$ and a reset gate $r_t$. The update gate determines the influence of the previous hidden state $H_{t-1}$ on the new hidden state. The equation for the update gate is as follows:

$$z_t = \sigma(W_{xz}G'(X_t, A) + W_{hz}G(H_{t-1}, A) + b_z),$$

where $W_{xz}$ and $W_{hz}$ are the learnable weight matrices, and $b_z$ is the bias. The larger the $z_t$ value is, the more $H_{t-1}$ is brought in.

The reset gate determines how much of the previous hidden state $(H_{t-1})$ to forget. The smaller the $r_t$ value is, the more $H_{t-1}$ is ignored. The equation for the reset gate is as follows:

$$r_t = \sigma(W_{xr}G'(X_t, A) + W_{hr}G(H_{t-1}, A) + b_r),$$

where $W_{xr}$ and $W_{hr}$ are the learnable weight matrices, and $b_r$ is the bias.

Using these gates, the gated recurrent component can update its memory vector as follows:

$$\tilde{H}_t = \tanh(W_{xz}G'(X_t, A) + W_{xh}G(r_t \circ (H_{t-1}, A)),$$

$$H_t = z_t \odot H_{t-1} + (1 - z_t) \odot \tilde{H}_t,$$

where $\odot$ is the dot product operation, $\tanh(\cdot)$ represents the hyperbolic tangent function, $W_{xh}, W_{hh}$ are the learnable weights.

### III. Experiments and Results

#### A. Experimental Setup

1) Performance Indicators: Three evaluation metrics were selected to quantify the performance of the model.

a) Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{T} \sum_{t=1}^{T} |Y^*_t - \hat{Y}_t|,$$

where $Y^*_t$ denotes the true value and $\hat{Y}_t$ denotes the predicted value, and $T$ denotes the time step.

b) Root Mean Square Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{T} \sum_{t=1}^{T} (Y^*_t - \hat{Y}_t)^2}.$$
Fig. 6. Comparisons between the 5-, 15-, 30-, 45- and 60-minute predictions made by $G^2$RNN and actual values. The red curves represent the predictions, and the black curves represent the ground truth values.
TABLE II

| Indicators | Packing Int. | 5min | 15min | 30min | 60min |
|------------|--------------|------|-------|-------|-------|
| CNN-GRU, dense prediction | 0.30 | 0.01 | 0.01 | 0.01 | 0.01 |
| GCN-GRU, sparse prediction | 0.30 | 0.01 | 0.01 | 0.01 | 0.01 |

TABLE III

| Models | Indicators | 5min | 15min | 30min | 60min |
|--------|------------|------|-------|-------|-------|
| G2RNN | RMSE | 9.10 | 10.10 | 18.84 | 22.84 |
| MAE | 1.94 | 2.95 | 9.58 | 11.96 | 14.81 |
| NMAE | 1.89 | 2.83 | 4.27 | 5.39 | 6.90 |
| GCN+GRU | RMSE | 18.10 | 18.76 | 31.18 | 34.57 | 38.26 |
| MAE | 14.93 | 13.81 | 22.06 | 23.31 | 26.92 |
| NMAE | 5.07 | 5.23 | 8.01 | 8.74 | 10.47 |
| ConvLSTM-DCN | RMSE | 5.75 | 11.14 | 16.91 | 23.14 | 28.67 |
| MAE | 5.38 | 7.92 | 12.02 | 16.13 | 19.30 |
| NMAE | 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 |
| NMAE | 1.87 | 3.74 | 6.09 | 8.24 | 10.39 |
| GRU-NN | RMSE | 8.53 | 11.57 | 19.40 | 27.85 | 36.92 |
| MAE | 4.01 | 8.02 | 13.79 | 19.45 | 26.93 |
| NMAE | 1.92 | 3.72 | 6.36 | 8.69 | 11.01 |
| SAE | RMSE | 7.24 | 12.67 | 18.94 | 27.36 | 37.70 |
| MAE | 5.02 | 8.92 | 14.04 | 18.67 | 26.58 |
| NMAE | 2.71 | 4.25 | 6.84 | 9.39 | 11.18 |
| SVR | RMSE | 15.25 | 14.41 | 20.08 | 28.69 | 38.14 |
| MAE | 10.12 | 10.44 | 14.94 | 21.58 | 28.68 |
| NMAE | 3.50 | 5.40 | 6.75 | 9.15 | 11.66 |
| BPNN | RMSE | 5.81 | 11.63 | 19.60 | 28.57 | 37.91 |
| MAE | 3.98 | 8.17 | 14.22 | 20.81 | 27.82 |
| NMAE | 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 |
| NMAE | 2.12 | 3.42 | 7.24 | 9.74 | 12.00 |

c) Mean Absolute Percentage Error (MAPE):  
\[ \text{MAPE} = \frac{100\%}{T} \sum_{t=1}^{T} \frac{|Y_t^i - Y_t^i|}{Y_t^i}. \]  

(11)

In the case that MAPE is not computable (i.e., \( Y_t^i = 0 \)), SMAPE (Symmetric MAPE) is used instead of MAPE. SMAPE can be obtained by  
\[ \text{SMAPE} = \frac{100\%}{T} \sum_{t=1}^{T} \frac{|Y_t^i - \hat{Y}_t^i|}{\left( |Y_t^i| + |\hat{Y}_t^i| \right)/2}. \]  

(12)

Generally speaking, the smaller the values of the above indicators, the better the prediction performance of the model.

2) Baseline Models: We compared the performance of the G2RNN model with the following models: ConvLSTM-DCN [19], GCN+GRU, and six other mainstream ML/DL models developed in our previous work [16], 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, a stacked autoencoder (SAE) model, a support vector regression (SVR) model, and a back propagation neural network (BPNN) model.
model, and a k-nearest neighbor (KNN) model. Among them, the ConvLSTM-DCN and GCN+GRU models are multi-point prediction models that can make predictions for multiple parking lots in parallel, while the others can only predict the number of VPSs for one parking lot.

The values of the hyper-parameters of all models were fine-tuned by the grid search method. Specifically, Adam optimizer is used in $G^2RNN$, the training epoch size is 500, the batch size is 32, and the learning rate is 0.001. The hyper-parameter settings for the GCN+GRU model are the same as those of the $G^2RNN$ model. The settings of the ConvLSTM-DCN model and other six models can be found in [19] and [16], respectively.

**B. Experimental Setup**

In the experiment, the Santa Monica dataset is divided into training and data sets in a 4:1 ratio. The model is fed with $m = 12$ historical observations, and the step size $\delta = 5$ min. The task are to predict numbers of VPSs after 5, 15, 30, 45 and 60 minutes. The experimental results were the average of 50 independent experiments. The experiments were conducted on a 64-bit server, the experimental environment is shown in Table I.

**C. Multi-Point Predictions**

We first compared the accuracy of the multi-point predictions made by $G^2RNN$, ConvLSTM-DCN and GCN+GRU, respectively, for different prediction methods, namely the single-step direct predictions and multi-step iterative predictions. Table II shows the results of 5-, 15-, 30-, 45-, and 60-min predictions of the number of VPSs of St1$\sim$St8 in Santa Monica. The best of these results are marked in bold red.

| Indicators | Parking lot | 5min | 15min | 30min | 45min | 60min |
|-----------|------------|------|-------|-------|-------|-------|
| ConvLSTM-DCN | hz1 | 3.56 | 3.99 | 5.72 | 8.54 | 9.13 |
| | hz2 | 5.13 | 5.23 | 12.89 | 11.88 | 12.29 |
| | hz3 | 10.21 | 7.44 | 12.89 | 19.60 | 25.78 |
| | hz4 | 5.05 | 4.69 | 8.51 | 16.30 | 20.17 |
| | hz5 | 5.78 | 5.95 | 8.11 | 10.99 | 13.00 |
| GCN+GRU | hz1 | 9.80 | 9.95 | 10.46 | 10.91 | 11.68 |
| | hz2 | 9.46 | 9.95 | 9.97 | 10.25 | 11.60 |
| | hz3 | 10.20 | 12.14 | 20.53 | 24.56 | 28.84 |
| | hz4 | 11.34 | 12.47 | 15.14 | 22.98 | 25.23 |
| | hz5 | 13.00 | 13.28 | 15.05 | 15.46 | 17.25 |
| MAE | hz1 | 2.20 | 3.21 | 5.65 | 12.52 | 13.72 |
| | hz2 | 2.23 | 3.62 | 6.79 | 12.91 | 14.13 |
| | hz3 | 3.48 | 7.44 | 11.14 | 15.27 | 19.97 |
| | hz4 | 2.27 | 4.86 | 10.75 | 18.28 | 20.95 |
| | hz5 | 5.77 | 6.78 | 8.95 | 13.49 | 14.25 |
| G2RNN | hz1 | 1.72 | 12.38 | 12.81 | 13.89 | 14.47 |
| | hz2 | 12.38 | 12.66 | 13.43 | 13.86 | 15.96 |
| | hz3 | 14.63 | 18.43 | 28.46 | 32.58 | 36.71 |
| | hz4 | 16.32 | 17.92 | 23.12 | 34.07 | 37.99 |
| | hz5 | 18.40 | 18.83 | 21.54 | 23.96 | 25.30 |
| RMSE | hz1 | 2.85 | 4.39 | 6.78 | 14.54 | 15.43 |
| | hz2 | 3.12 | 4.94 | 8.57 | 16.17 | 16.38 |
| | hz3 | 4.64 | 9.60 | 15.20 | 22.21 | 39.36 |
| | hz4 | 3.63 | 6.55 | 13.35 | 24.17 | 25.80 |
| | hz5 | 7.30 | 8.36 | 11.08 | 16.57 | 16.73 |

The results are three-fold. First, it can be seen in Table II that the MAE, MAPE, and RMSE of the $G^2RNN$ model are smaller than those of the comparison models, for both direct and iterative predictions. In terms of MAE, MAPE, and RMSE, the direct predictions with $G^2RNN$ model performed better on 34, 34 and 29 out of 40 tasks, respectively, than those with ConvLSTM-DCN model. In iterative predictions, predictions made by $G^2RNN$ model outperformed those by ConvLSTM-DCN model 26, 26 and 28 times (totally 32 tasks), respectively. Second, long-term predictions are always less accurate than short-term ones. 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. Nevertheless, the experimental results show that the performance of $G^2RNN$ degrades much more slowly than comparison models and is more stable. Third, for $G^2RNN$, direct predictions are always better than iterative ways. The results demonstrated that the direct prediction method performed better on 27, 29, and 27 out
of 32 tasks (i.e. 15-/30-/45-/60-min predictions for 8 parking lots under three evaluation indicators, i.e. RMSE, MAE and MAPE), respectively. Therefore, in the following experiments, the former approach was adopted unless otherwise stated.

The effectiveness of G$^2$RNN is also confirmed in Fig. 6. We can find that the prediction curves (red curves) fit well with the ground truth ones (black curves), which verifies that the prediction model based on G$^2$RNN achieves fairly high prediction accuracy.

D. Single-Point Predictions

Next, the performance of single-point prediction was evaluated. The prediction method based on G$^2$RNN was compared with the ConvLSTM-DCN model, GCN+GRU model and six single-point prediction models, namely an LSTM model, a GRU-NN model, an SAE model, an SVR model, a BPNN model, and a KNN model. In this comparative experiment, St7 was considered as the target. It can be observed from Table III that although the prediction accuracy of the G$^2$RNN in 5 min is slightly lower than that of the LSTM model in 5-min prediction (i.e. MAPE, with a sight difference of 0.02%), our method outperforms all the other prediction tasks in 15-, 30-, 45-, and 60-min prediction tasks.

T-Test was used to calculate the p-values to verify whether there was statistical difference between the performance of G$^2$RNN and the comparison models. The results are listed in Table IV, which show that the observed differences in Table III are statistically significant for all case except the one with regard to the difference between G$^2$RNN and ConvLSTM-DCN in 5-minute prediction (with a p-value of 0.0828).

E. Runtime Comparisons

Table V gives the runtime costs for different models in CPU/GPU conditions, respectively, which are implemented in a 64-bit server. The average training time of 30 independent experiments was used as the final training time, and the average prediction time of 10,000 independent predictions was used as the final prediction time. Compared with the dConvLSTM-DCN and ConvLSTM-DCN models which have complex internal structures, G$^2$RNN can achieve shorter training and prediction time due to its simple internal structure. In the case of using GPU, the training time of the G$^2$RNN model is only 56.77% of that without GPU. Henceforth, GPUs are recommended due to the significant speed boost compared to CPUs. However, considering the much higher cost of GPUs, CPUs are also acceptable.

F. Case Study on Parking Lots in Hangzhou

To assess the potential transferability of the model, we utilized an additional dataset collected from five parking lots (Hz1-Hz5 in Fig. 7) in Hangzhou, China. Similarly to the previous dataset, the number of VPSs was recorded every five minutes for each parking lot, from 11/14/2022 0:40 to 11/27/2022 19:45, resulting in 3974 data points for each parking lot. We also evaluated the performance of three different models - G$^2$RNN, ConvLSTM-DCN, and GCN+GRU - and compared their results. Table VI summarizes our findings (the best of these results are marked in bold red), indicating that G$^2$RNN achieved the best performance when compared to the other two models. Specifically, based on our evaluation metrics of MAE, MAPE, and RMSE, the G$^2$RNN model outperformed the other models in 13, 16, and 12 out of 25 tasks, respectively.

IV. CONCLUSION AND DISCUSSIONS

There exists strong and obvious spatio-temporal correlations between the number of VPSs of different parking lots within an area. In view of this, this article proposes a novel deep learning model, named G$^2$RNN, to simultaneously predict the numbers of VPSs for multiple parking lots by fully leveraging the inherent spatio-temporal correlations between them. By introducing the GCN operations into GRU cells, the G$^2$RNN model can extract precise spatial and temporal correlations between different parking lots in parallel and make multi-point predictions with input graph-structured historical observations. The G$^2$RNN model was applied to eight public parking lots in Saint Monica and the numbers of VPSs in 5, 15, 30, 45 and 60 minutes were predicted. Moreover, two long-term prediction approaches, i.e., single-step direct prediction and multi-step iterative prediction approaches, were investigated. The results show that the proposed model can make multi-point predictions with considerably high accuracy. Moreover, we also compared the G$^2$RNN model with other fine-tuned models. The results demonstrate that the G$^2$RNN model is superior to others, especially in long-term predictions.

Discussion 1: Our current work only focuses on predicting VPS availability and not occupancy of parking spaces. However, considering both the predictions of VPS availability and occupancy of parking spaces can provide several benefits.

First, improved accuracy: Combining VPS availability and occupancy predictions can help to provide a more accurate picture of the parking situation. By analyzing the occupancy data in conjunction with the VPS predictions, it may be possible to identify trends and patterns in parking behavior that can inform more accurate predictions of VPS availability.

Second, enhanced user experience: By knowing not just whether a parking spot is available, but also how many other spots are occupied in the surrounding area, drivers can make more informed decisions about where to park and potentially avoid congested areas.

Third, better resource utilization: By considering both VPS availability and occupancy, it may be possible to optimize parking lot usage and minimize wasted space. For example, if the occupancy data shows that certain areas of the parking lot are consistently underutilized, this information can be used to guide drivers to those areas and increase overall parking efficiency.

It is a promising direction of our future work.

Discussion 2: It is important to consider the ability of our model to handle changes in parking lot infrastructure, regulations or usage patterns.

One potential approach to addressing this issue in future work is to explore the use of transfer learning techniques, which can
enable the model to adapt to new parking lot configurations, regulatory changes and diverse usage patterns using smaller amounts of new data. Additionally, ongoing data collection and analysis can help to identify patterns and trends in parking lot usage that can inform updates to the model over time. By taking a proactive approach to addressing this aspect, the model can be designed to be more robust and adaptable to changes in the parking landscape. It may also be useful to explore the feasibility of incorporating external factors, such as weather or local events, that may impact parking availability in the area.

REFERENCES

[1] D. C. Shoup, “Cruising for parking,” Transport Policy, vol. 13, no. 6, pp. 479–486, 2006.
[2] D. Shoup, “Free parking or free markets,” Access Mag., vol. 1, pp. 28–35, 2011.
[3] T. Giuffrè, S. M. Siniscalchi, and G. Tesoriere, “A novel architecture of parking management for smart cities,” Procedia - Social Behav. Sci., vol. 53, pp. 16–28, Oct. 2012.
[4] F. Caicedo, “Real-time parking information management to reduce search time, vehicle displacement and emissions,” Transp. Res. Part D: Transport Environ., vol. 15, no. 4, pp. 228–234, 2010.
[5] R. Atat, L. Liu, J. Wu, G. Li, C. Ye, and Y. Yang, “Big data meet cyber-physical systems: A panoramic survey,” IEEE Access, vol. 6, pp. 73603–73636, 2018.
[6] H. Song, J. Bai, Y. Yi, J. Wu, and L. Liu, “Artificial intelligence enabled Internet of Things: Network architecture and spectrum access,” IEEE Comput. Intell. Mag., vol. 15, no. 1, pp. 44–51, Feb. 2020.
[7] Y. Kamarianakis and V. Voutou, “Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches,” Transp. Res. Rec. J. Transp. Res. Board, vol. 1857, no. 1, pp. 74–84, 2003.
[8] H. T. Akaike, “A new look at the statistical model identification,” IEEE Trans. Autom. Control, vol. 19, no. 6, pp. 716–723, 1974.
[9] A. Shakeel, T. Tanaka, and K. Kitajo, “Time-series prediction of the oscillatory phase of EEG signals using the least mean square algorithm-based AR model,” Appl. Sci., vol. 10, no. 10, 2020, Art. no. 3616.
[10] M. Laner, P. Svoboda, and M. Rupp, “Parsimonious fitting of long-range dependent network traffic using ARMA models,” IEEE Commun. Lett., vol. 17, no. 12, pp. 2368–2371, Dec. 2013.
[11] G. E. P. Box and G. M. Jenkins, “Time series analysis forecasting and control,” J. Time Ser. Anal., vol. 3, no. 3228, 1970.
[12] J. Fan, Q. Hu, and Z. Tang, “Predicting vacant parking space availability: An SVR method with fruit fly optimisation,” IET Intell. Transport Syst., vol. 12, no. 10, pp. 1414–1420, 2018.
[13] D. Chuan, W. Donggen, M. Xiaoeli, and L. Haying, “Predicting short-term subway ridership and prioritizing its influential factors using gradient boosting decision trees,” Sustainability, vol. 8, no. 11, 2016, Art. no. 1100.
[14] X. Ye, J. Wang, T. Wang, X. Yan, Q. Ye, and J. Chen, “Short-term prediction of available parking space based on machine learning approaches,” IEEE Access, vol. 8, pp. 174530–174541, 2020.
[15] F. G. Habtemichael and M. Cetin, “Short-term traffic flow rate forecasting based on identifying similar traffic patterns,” Transp. Res. Part C: Emerg. Technol., vol. 10, pp. 61–78, 2016.
[16] J. Fan, Q. Hu, Y. Xu, and Z. Tang, “Predicting vacant parking space availability: A long-term short memory approach,” IEEE Intell. Transp. Syst. Mag., vol. 14, no. 2, pp. 129–143, Mar./Apr. 2022.
[17] C. Zeng, C. Ma, K. Wang, and Z. Cui, “Parking occupancy prediction method based on multi factors and stacked GRU-LSTM,” IEEE Access, vol. 10, pp. 47361–47370, 2022.
[18] F. De et al., “The vehicle’s velocity prediction methods based on RNN and LSTM neural network,” in Proc. IEEE Chin. Control Decis. Conf., 2020, pp. 99–102.
[19] Y. Feng, Z. Tang, Y. Xu, S. Krishnamoorthy, and Q. Hu, “Predicting vacant parking space availability zone-wisely: A densely connected ConvLSTM model,” in Proc. IEEE Veh. Power Propulsion Conf., 2021, pp. 1–6.
[20] Y. Feng, Y. Xu, Q. Hu, S. Krishnamoorthy, and Z. Tang, “Predicting vacant parking space availability zone-wisely: A hybrid deep learning approach,” Complex Intell. Syst., vol. 8, pp. 4145–4161, 2022.
[21] S. Fang, Q. Zhang, G. Meng, S. Xiang, and C. Pan, “GSTNet: Global spatial-temporal network for traffic flow prediction,” in Proc. 28th Int. Joint Conf. Artif. Intell., 2019, pp. 2286–2293.

Qian Hu received the B.S. degree in communication engineering from the Nanjing Institute of Posts and Telecommunications, Nanjing, China, in 2001, and the M.S. degree in circuits and systems from Zhejiang University, Hangzhou, China, in 2005. She is currently an Associate Professor with the College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou, China. Her research interests are in the areas of machine learning, intelligent systems and AI-enabled wireless networks.

Yajing Feng received the B.S. degree in information and computing science from the Ningbo University of Technology, Ningbo, China in 2019, and the M.S. degree in computer science and technology from Wenzhou University, Wenzhou, China, in 2022. Her research interests include artificial intelligence, machine learning, and deep learning.

Zhenzhong Tang (Senior Member, IEEE) received the B.S. and M.S. degrees in communication engineering from Zhejiang University, Hangzhou, China, in 2001 and 2004, respectively, and the Ph.D. degree in communication engineering from the Dalain University of Technology, Dalain, China, in 2015. In March 2004, he joined Wenzhou University, Wenzhou, China, where he is currently a Full Professor with the College of Computer Science and Artificial Intelligence, and the Director with the Wenzhou Key Laboratory for Intelligent Networking and the Director with the Innovation Research Center for Intelligent Networking. Wenzhou University. From February 2015 to August 2015, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA, USA. From November 2015 to June 2018, he was a Postdoctoral Research Fellow in Zhejiang University. His research interests include machine learning, intelligent systems, and wireless networks.

Yiying Xu received the Ph.D. degree in communication engineering from Zhejiang University, Hangzhou, China, in 2017. From 2015 to 2016, she was a Visiting Research Scholar with the School of Electrical and Computer Engineering, University of California, Davis, CA, USA. She is currently an Associate Researcher with the School of Electronic Information, Wuhan University, Wuhan, China. She was the recipient of the Excellence Award of Excellent Ph.D. Thesis of China Electronics Society in 2017, Chu Kochen Scholarship at Zhejiang University in 2016, and Ph.D. Researcher National Scholarship in 2015. Her research interests include 5G communications, WLAN, and energy efficient communication techniques in next generation wireless communication systems.