Predicting Parking Lot Availability by Graph-to-Sequence Model: A Case Study with SmartSantander

Yuya Sasaki
Osaka University, Japan
sasaki@ist.osaka-u.ac.jp

Junya Takayama
Osaka University, Japan
takayama.junya@ist.osaka-u.ac.jp

Juan Ramón Santana
Universidad de Cantabria, Spain
jrsantana@tlmat.unican.es

Shohei Yamasaki†
Nomura Research Institute, Ltd., Japan
s2-yamasaki@nri.co.jp

Tomoya Okuno
Osaka University, Japan
okuno.tomoya@ist.osaka-u.ac.jp

Makoto Onizuka
Osaka University, Japan
onizuka@ist.osaka-u.ac.jp

***Abstract*—Nowadays, so as to improve services and urban area livability, multiple smart city initiatives are being carried out throughout the world. SmartSantander is a smart city project in Santander, Spain, which has relied on wireless sensor network technologies to deploy heterogeneous sensors within the city to measure multiple parameters, including outdoor parking information. In this paper, we study the prediction of parking lot availability using historical data from more than 300 outdoor parking sensors with SmartSantander. We design a graph-to-sequence model to capture the periodical fluctuation and geographical proximity of parking lots. For developing and evaluating our model, we use a 3-year dataset of parking lot availability in the city of Santander. Our model achieves a high accuracy compared with existing sequence-to-sequence models, which is accurate enough to provide a parking information service in the city. We apply our model to a smartphone application to be widely used by citizens and tourists.**

***Index Terms*—Graph neural network, Internet of Things, Smart city, Spatio-temporal analysis**

I. Introduction

The Internet of Things (IoT) generates a large volume of real-time data. Smart city projects have emerged to manage their services more effectively based on IoT technologies. Santander, Spain, started its smart city project called SmartSantander[1], which envisioned a large-scale deployment of more than 12,000 IoT sensors such as traffic, environmental, and parking sensors [18], [19]. The aim of such deployment is to improve the city livability and support tourism in Santander by using ICT techniques, for example, route recommendation [21] and air quality analysis [7], [8], [20].

One of the services provided in Santander is guiding drivers to available parking lots. Thanks to the outdoor parking information gathered from the SmartSantander deployment, drivers are informed about available parking lots through two means. First, ten parking panels, deployed at the entrance to streets in the city center, show the number of available parking lots in the city center as well as each of the streets where they are deployed. Second, several applications provide this information remotely. Hence, drivers can plan whether to take their own cars or use public transportation services based on real-time parking information.

Fig. 1: Parking sensors in Santander. (top) locations of parking lots with cluster IDs, (bottom-left) a parking sensor with a car, (bottom-center) the parking sensor, and (bottom-right) a parking panel to guide drivers.

Figure 1 shows the deployment carried out in Santander, including the location of parking lots equipped with sensors, parking sensors, and the parking panels deployed in the city. Such services providing parking lot availability improve the efficiency and usefulness of the parking system.

Motivation. Despite the benefits provided by this deployment, SmartSantander is limited to offering real-time information to the users about available parking lots. A prediction service based on historical data can benefit the system deployed in Santander. Such prediction service would allow drivers

†This work was done when Shohei Yamasaki was a student at Osaka University

[1]https://www.smartsantander.eu/

to know whether there are available parking lots or not beforehand, even if the service has been accessed long before the arrival at the parking lot. Furthermore, SmartSantander can keep offering parking guiding services by predicting the number of available parking lots based on historical data, even if the service stops due to several issues, such as deployment maintenance or urban works.

There are two requirements to start a service that predicts parking availability in Santander: (1) predicting the number of available parking lots for both, the whole city center (i.e., all parking lots) and each of the streets; and (2) predicting their availability at multiple time steps *without* pre-defined time steps (e.g., in 15, 30, and 60 minutes, and more). To the best of our knowledge, there is no former work that predicts the availability of parking lots per street at several time steps.

Contribution. In this paper, we study the problem on the prediction of parking lot availability with SmartSantander. To this end, we design a graph-to-sequence neural network model that can predict parking lot availability at multiple time steps by using historical data from parking lots. Our graph-to-sequence model consists of encoder and decoder. For the encoder, we develop a graph neural network (GNN) encoder to capture the characteristics of parking lot availability: (1) temporal, as parking lot availability trends change periodically, and (2) spatial, as the availability of each parking lot is also affected by nearby parking lots. Our decoder is based on a recurrent neural network (RNN) that uses time information (e.g. a day) as well as historical parking lot availability data to learn the impact of holidays and weekdays.

We demonstrate that our model is able to accurately predict the number of available parking lots in periods of 15, 30, 60, and 120 minutes. Our model outperforms existing sequence-to-sequence models. We also develop a smartphone application with our prediction model to provide accurate predictions to citizens and tourists.

Reproducability. We open our source code in Github and parking data under requests[2].

II. Related work

We review the similar existing works. Since there are a large number of existing works on parking prediction, please refer a survey paper [14] that summarizes existing works related to smart parking systems.

Parking prediction. Due to the developments of smart city projects, nowadays many cities monitor the status of parking lots, such as Berlin [23], Barcelona [2] and Santander. There are many works that aim to predict parking lot availability. Tiedemann et al. [23] developed a system that predicts occupancy for parking spaces in Berlin, Germany, which collects data from roadside parking sensors. This system predicts occupancy by using machine learning methods combined with data threads. Caicedo et al. [2] developed a real-time availability forecast (RAF) algorithm for predicting real-time parking lot availability, located in Barcelona, Spain. Chen et al. [4] tackle the parking problem by aggregating parking lots in the same way as we do in our problem, in San Fransisco. They evaluated multiple models, such as ARIMA, linear regression, support vector regression, and feed-forward neural networks, and the neural network algorithm achieved the best performance.

Several neural network-based prediction models have been proposed nowadays [3], [11], [22], [26][3]. For example, Shao et al. [22], Jomaa et al. [11], Xiao et al. [26] use LSTM, CNN, and GCN with GLU, respectively. Existing works predict the availability of parking lots at either a single step (i.e., current or near future) or pre-defined multiple steps. As our aim is to predict the availability of parking lots at non-predefined multiple-time steps simultaneously, existing algorithms are not applicable to our problem. The sequence-to-sequence models that we used as our baselines can be considered as the extension of existing works [11], [22], and we validated that our method achieves higher accuracy than them.

Some existing methods use not only historical parking data but also other data sources for predicting parking lot availability. For example, data sources include historical data generated by mobile phones (e.g., [15]), data extracted from vehicles equipped with GPS receivers (e.g., [16]), and information from web maps (e.g., [28], [31]). In this regard, our model could be improved if we include these extra data. However, such data is difficult to be accessed, and we must deal with privacy related aspects, as well as consider that not all users use such recent devices and services. So, they are not suitable in the situations of SmartSantander.

Deep learning on spatial and temporal data prediction. There are similar works with parking availability prediction such as predicting traffic density, pollution data, and the general availability of resources [10], [17]. These works are categorized into grid-based and graph-based predictions [10]. The grid-based prediction divides areas into equal-size grids and predicts the values for each grid (e.g., [30]). However, equal-size grids are not suitable for our problem because we aim to predicate the number of available parking lots per street, which cannot divide equal-size grids.

Our work belongs to the graph-based prediction which estimates attributes of nodes on graphs (e.g., [12], [24], [25], [27], [29], [32]). Nodes on graphs often represent sensors such as temperature and traffic volume sensors, and their measurements are attributes of the nodes. In our study, nodes represent parking clusters and their attributes are the parking lots availability, so the existing methods can be applied to parking predictions. To the best of our knowledge, there are no works that applied to parking lots availability yet.

III. Parking sensors with SmartSantander

As aforementioned, we develop our prediction model for parking sensors deployed in Santander. There are 323 parking sensors and each sensor corresponds to a single parking lot. All these sensors are connected to a wireless network.

[2]https://github.com/yuya-s/SatanderParking

[3]These works are not coupled with smart city projects.

Monitored parking lots are located in a special area in the city center, in which the drivers have to pay parking fee from 10:00 to 14:00 and 16:00 to 20:00 during weekdays and Saturday mornings. Parking time is restricted to two hours per vehicle, except for those citizens who live nearby and have special permission. Therefore, the status of parking lots changes frequently in these periods. Parking lots show a high occupancy for the full day, even overnight.

IV. Prediction model

In this section, we first explain the requirements and problem definition that we solve in this paper. Then, we explain the preprocessing for the parking data. Finally, we explain our graph-to-sequence neural network model and its training method.

A. Requirements and Problem Definition

As we described in Section 1, we have two requirements to start the prediction services in Santander. First, we need to predict the number of available parking lots for both, the whole city center (i.e., the entire parking area) and per street. We do not need to predict the status of each parking lot because the parking services in Santander provide street-based parking availability. Second, we need to predict the number of available parking lots at multiple time steps instead of a single time step. These time steps should not be defined in advance, because people may move from further areas to the city center by their cars and the distances are unsure.

We here formally define the problem we solve in this paper. We have the set P of historical parking lot data for each parking lot. $p_i \in P$ is a vector whose size is the number of parking lots and it consists of 0/1 values, where 1 and 0 represent whether each parking lot is available or not at time step i, respectively. We define that $\langle p_i, \ldots, p_j \rangle$ is a sequence of vectors from time step i to j.

Problem Definition: *Given the set of historical parking lots data P and locations of parking lots, we build a model to predict the parking lot availability. In this model, given a sequence of vectors $\langle p_{t-M}, \ldots, p_{t-1} \rangle$ as input, it outputs the number of available parking lots per street from time step t to t' (t' is not given in advance).*

B. Preprocessing

As preprocessing of parking lots data, we cluster the parking lots and construct a graph based on the closeness among parking lots .

Clustering. It is effective because nearby parking lots are likely to have similar behavior, as we do not have specific parking lots but specific areas where we can park our cars. We cluster the parking lots per street, considering that this is how SmartSantander provides the available parking lots to citizens. After this process, we obtain 27 different clusters. Figure 1(top) shows these clusters, where different colors represent each one of them. The number of parking lots differs for each of the clusters. For instance, clusters 2, 9, and 19 in Figure 1(top) include 5, 20, and 29 parking lots, respectively.

We normalize the values in the vector by dividing them by the number of parking lots in the cluster in order to reduce unnecessary effects of the clusters that include large numbers of parking lots. Finally, we have the set of vectors S that contains a vector s_i whose size is 27 and values are from 0 to 1 at time step i. We input $\langle s_{t-M}, \ldots, s_{t-1} \rangle$ to models, and the models output $\langle s_t, \ldots, s_{t'} \rangle$.

Graph construction. We build a graph whose vertices are clusters of parking lots and two vertices have an edge if the parking lots represented by the two vertices are closer than a given spatial threshold. We assume that parking lots represented by the two vertices are affected each other if both vertices have edges. So as to locate each cluster, we use the centroid of the locations of parking lots that belong to that cluster. Two vertices have edges if the Euclidean distance between them is a given threshold.

C. Graph-to-Sequence model

Our graph-to-sequence model consists of a *GNN encoder* and an *RNN decoder*. The GNN encoder maps the input to a vector of a fixed dimensionality by graph neural networks, and the RNN decoder receives and decodes it to generate the sequence of predicted parking lots availability.

GNN encoder model. Our GNN encoder model captures the temporal perspective by 1D-convolution and the spatial perspective by graph convolution, which captures local trends of nearby parking lots instead of global trends.

Figures 2 and 3 show our GNN encoder model and the input data for this model, respectively. Our GNN encoder model combines gated graph neural networks (GGNN for short) [13] with gated convolutional neural network (GCNN for short) [6]. The GGNN takes graph convolution to capture spatial perspective and the GCNN takes 1D-convolution to capture temporal perspective. The propagation model in the GGNN conducts graph convolution that learns about the effect from neighbor clusters (i.e., vertices connected by edges). Consecutive GCNNs after the GGNN emphasizes the characteristics by gradually reducing the size of the vector. Our GNN encoder has three input parameters, parking a_i, hidden state h_i, and set of edges E of the graph. a_i is a vector whose values are the number of available parking lots of cluster i at M steps, and h_i is a vector that is extended from a_i and the extended area has zero values.

In the following, we describe our GNN encoder model in detail. $[;]$ and $\otimes$ denote concatenation and element-wise multiplication, respectively .

Our GNN encoder model first applies GCNNs to a_i and h_i, respectively. The GCNNs have two 1D-convolutional layers; the top 1D-CNN computes the importance via a sigmoid function and the bottom one computes values themselves. This is the same structures as the original GCNNs. a'_i and h'_i are outputs of the GCNNs, which are calculated as follows:

$$a'_i = (a_i * \Gamma_{a,f} + b_{a,f}) \otimes \sigma(a_i * \Gamma_{a,g} + b_{a,g}) \quad (1)$$

$$h' = (h_i * \Gamma_{h,f} + b_{h,f}) \otimes \sigma(h_i * \Gamma_{h,g} + b_{h,g}) \quad (2)$$

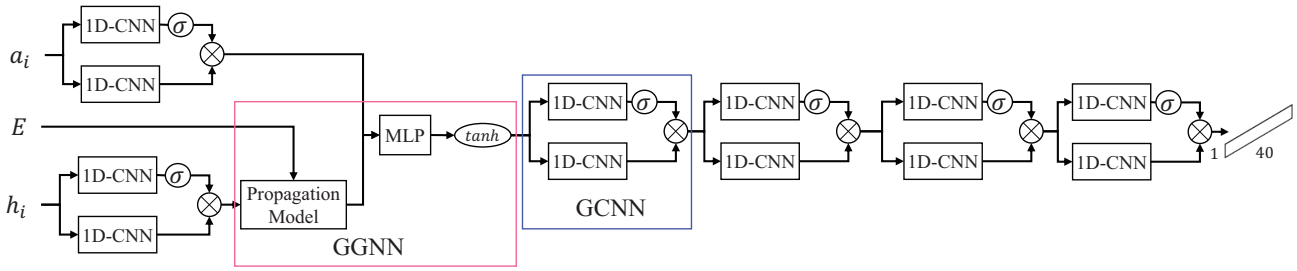


Fig. 2: GNN encoder model. Our GNN encoder consists of one GGNN and six GCNN layers.

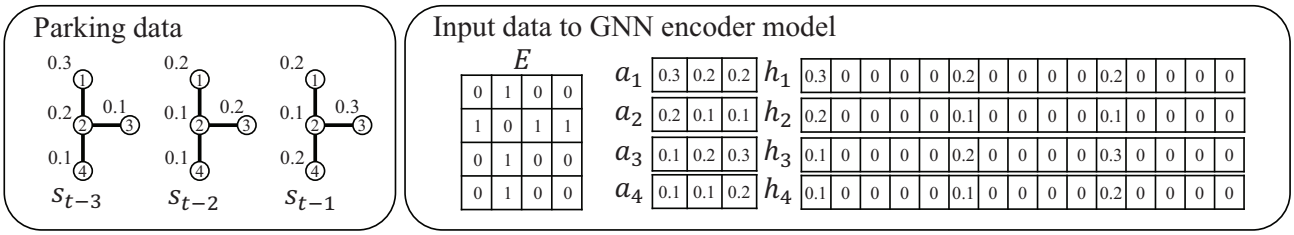


Fig. 3: Input data to the GNN encoder model

where σ denotes sigmoid function. $\Gamma_{a,f}, \Gamma_{a,g}$ and $\Gamma_{h,f}, \Gamma_{h,g}$ are 1-dimensional convolution operators for annotations and hidden states, respectively. $b_{a,f}, b_{a,g}$ and $b_{h,f}, b_{h,g}$ are correspondence biases.

The GGNN aggregates hidden states of neighbors at each time step (Eq. (4)), and then updates the hidden states like GRU (Eqs. (5)–(8)) [5]. Then, an output of message passing is applied to MLP layer (Eq. (9)). The propagation model aggregates and updates repeatedly by the following equations:

$$
\begin{aligned}
h(0) &= h'_i && (3)\\
X_{(j)} &= E[h_{(j-1)}W_1 + b_1; \ldots; h_{(j-1)}W_M + b_M] && (4)\\
r_{(j)} &= \sigma(X_{(j)}W_r + h_{(j-1)}U_r) && (5)\\
z_{(j)} &= \sigma(X_{(j)}W_z + h_{(j-1)}U_z) && (6)\\
\widehat{h_{(j)}} &= \tanh(X_{(j)}W_h + (h_{(j-1)} \otimes r_{(j)}U_h)) && (7)\\
h_{(j)} &= (1 - z_{(j)}) \otimes h_{(j-1)} + z_{(j)} \otimes \widehat{h_{(j)}} && (8)
\end{aligned}
$$

where j denotes the time of iterations and X, r, z denote the message, reset gate, and update gate, respectively. $W_1, \ldots, W_M, W_r, U_r, W_z, U_z, W_h, U_h$ and $b_1, \ldots, b_M$ are learnable parameters. The parameters $W_1, \ldots, W_M$, $b_1, \ldots, b_M$ are for message passing.

We apply an output model to the I times-propagated hidden state $h_{(I)}$. An output O_g of the GGNN is calculated as follows:

$$O_g = \tanh([h_{(I)}; a'_i]W_O + b_O) \quad (9)$$

where W_O, b_O are learnable parameters. We then repeatedly apply four GCNNs to O_g that equations are the same as Eq. (1).

Our GNN can capture spatial perspective for every parking lot because we input the data for each single parking lot one by one along with the graph that represents the closeness of parking lots. Therefore, our GNN model captures the local spatial perspective effectively.

RNN Decoder. Our RNN decoder employs a multilayered Long Short-Term Memory (LSTM) [9] to output the sequence of predicted parking lots availability without pre-defined time steps. We use time information d_t as input for the decoder in order to capture the periodical fluctuation. d_t includes four values that represent the time, day, month, and weekday. Due to the use of time information in the decoder, our model effectively handles periodical fluctuation, such as weekdays and weekends.

In our RNN decoder, each function is computed by the same equations of basic LSTM [9], with the exception that we use a ReLU function for embedding the inputs, and a sigmoid function for the output. In more concretely, our decoder model consists of the follow equations:

$$
\begin{aligned}
\hat{s}_t &= ReLU(W_s s_t + b_s) && (10)\\
\hat{d}_t &= ReLU(W_d d_t + b_d) && (11)\\
f_t &= \sigma(W_f[h_t; \hat{s}_t; \hat{d}_t] + b_f)\\
l_t &= \sigma(W_i[h_t; \hat{s}_t; \hat{d}_t] + b_l)\\
\tilde{C}_t &= \tanh(W_C[h_t, \hat{s}_t, \hat{d}_t] + b_C)\\
C_{t+1} &= f_t \otimes C_t + l_t\tilde{C}_t\\
o_t &= \sigma(W_o[h_t; \hat{s}_t; \hat{d}_t] + b_o)\\
h_{t+1} &= o_t \otimes \tanh C_{t+1}\\
s_{t+1} &= \sigma(W_{s'}C_{t+1} + b_{s'}) && (12)
\end{aligned}
$$

These equations are the same for basic LSTM cells except for Eqs. (10)–(12). Eqs. (10) and (11) are for embedding, and Eq. (12) is for output. Both h_t and C_t are initially set to the output of encoder.

D. Training

We use the mean absolute error (MAE) as the measure to quantify errors for our training data. Hence, we minimize the MAE measure in the objective function over training data. Our loss function is defined as follows:

$$Loss = \frac{1}{N' \cdot |s|} \sum_{i=1}^{N'} \sum_{j=1}^{|s|} abs(s'_{i,j} - s_{i,j}) \tag{13}$$

where, $|s|$ denotes the size of vector s (i.e., 27), and $s'_{i,j}$ and $s_{i,j}$ denote the predicted and measured values of cluster j at time step i, respectively. N' is a predefined parameter for model training, and we set N' to a random value.

V. Experiment

In this section, we evaluate our graph-to-sequence model to predict parking lot availability from the Santander dataset.

A. Setting

We provide an overview of our experimental setup, including the dataset, competitors and parameters.

Dataset. We use a dataset with three years of parking data from Santander. The number of parking lots covered in this dataset is 323, with data from 29th April 2014 to 29th January 2017. We sample the data each 15 minutes. Figure 1 (top) shows the location of parking lots.

We divide the dataset into a training set from April 29th, 2014, to December 29th, 2016; a validation set from December 29th, 2016, to January 19th, 2017; and a test set from January 19th, 2017, to January 29th, 2017. We here note that test data is small compared with train and validation data because we assume that SmartSantander often retrains models to provide accurate predictions.

Competitors. We evaluate our graph-to-sequence model compared with two sequence-to-sequence models and two traditional auto-regressive (AR) models.

In sequence-to-sequence models, we use two types of encoders RNN and CNN with the same decoder of our model. The RNN encoder consists of LSTM as same as the decoder. The CNN encoder convolutes a matrix containing $\langle s_{t-M}, \cdots, s_{t-1} \rangle$ to a fixed-size vector. We describe the detailed architectures of RNN and CNN encoders in the appendix.

In AR models, we use ARIMA and SARIMA [1], two of the most widely used methods for time series forecasting. ARIMA model assumes that the future value of a time series is a linear combination of historical values. SARIMA model takes into account seasonality as well. We note that there are no existing neural network models to directly apply our problem setting. We extend the sequence-to-sequence models CNN and RNN for parking lots predictions [11], [22].

Parameter. The input size of historical data M is 48. That is, we use parking data for 12 hours (i.e., $48 \cdot 15$ min) as historical data. We predict the parking data in 15 min to 2 hours. During our preliminary experiments, we tested different input sizes for the model: 4, 24, 48, and 96 as M, from which the chosen input size achieved high accuracy on average. In a graph, two vertices have edges if the Euclidean distance between them is 95 meters, which is decided based on the length of streets. To reduce overfitting, we apply dropout with a probability of 0.3 to the encoding.

We explain the architecture parameter of our GNN encoder. In the 1D-convolutional layers for a_i, the number of filters, filter size, and stride length are 1, 2, and 1, respectively. In the 1D-convolutional layers for h_i, the number of filters, filter size, and stride length are 2, 10, and 5, respectively. Four consecutive 1D-CNNs after the GGNN have 4, 115, 60, and 65 as the filter sizes, 2, 5, 5, and $null$ as the stride length, and 5, 10, 20, and 40 as the number of filters, respectively. We use $null$ as the stride length, as the filter is of the same size as the matrix size. We also set the number I of propagation in the GGNN to five. Finally, our GNN encoder outputs a vector whose size is 40, which is the input for our decoder model.

When we train the model, we set the size of the batch and epoch as 512 and 50, respectively. Then, in the test phase, we choose the same parameters that provide the highest accuracy in average during the validation phase.

B. Experimental results

Comparing our models with baselines. We first compare the prediction performance for our model and the four baselines. We evaluate MAE in 15, 30, 60, and 120 minutes time steps. Table I shows the MAE of the whole city for each time step and model.

TABLE I: MAE of available parking lots in the entire parking area. The bold font indicates the best accuracy.

	15 min	30 min	60 min	120 min
ARIMA	15.31	14.95	15.415	15.26
SARIMA	11.79	12.04	10.97	11.49
RNN	10.95	11.18	11.16	12.13
CNN	4.166	4.173	5.102	7.466
GNN	**2.511**	**3.301**	**4.608**	**7.242**

The GNN and CNN encoders achieve the best and second-best accuracy for all time steps, respectively. The accuracy of RNN encoder is almost the same accuracy of the SARIMA model. The GNN and CNN encoders capture the spatial perspective effectively, while RNN encoder and SARIMA are able to capture the seasonality effect, but they cannot capture spatial perspectives well. From these results, we can confirm that spatial perspectives are effective in parking prediction. In addition, the parking lot prediction becomes more accurate when using nearby parking lots than the entire parking area because the GNN encoder achieves higher accuracy than the CNN encoder.

In our graph-to-sequence and sequence-to-sequence models, MAE increases as time steps increase, while in AR models the MAE value remains constant. This is due to the fact that graph and sequence-to-sequence models accumulate errors in the previous steps, thus the error becomes larger as time steps increase.

TABLE II: MAE of available parking lots per street. The numbers within parentheses at Cluster ID denote the number of parking lots in clusters. The bold font indicates the best accuracy.

Cluster ID	RNN				CNN				GNN			
	15 min	30 min	60 min	120 min	15 min	30 min	60 min	120 min	15 min	30 min	60 min	120 min
0 (10)	0.749	0.734	0.769	0.769	0.468	0.497	0.540	**0.606**	**0.293**	**0.392**	**0.502**	0.640
1 (15)	0.803	0.816	0.822	0.843	0.552	0.556	0.579	0.647	**0.301**	**0.367**	**0.472**	**0.576**
2 (5)	0.404	0.392	0.416	0.429	0.088	0.095	0.117	0.155	**0.069**	**0.081**	**0.110**	**0.139**
3 (13)	0.911	0.899	0.911	0.904	0.669	0.717	0.773	0.844	**0.243**	**0.319**	**0.444**	**0.606**
4 (6)	0.265	0.291	0.302	0.301	0.126	0.148	0.175	0.231	**0.104**	**0.123**	**0.171**	**0.223**
5 (6)	1.015	0.978	0.968	0.975	0.513	0.530	0.587	0.632	**0.247**	**0.337**	**0.448**	**0.591**
6 (14)	0.806	0.786	0.791	0.803	0.579	0.645	0.675	0.806	**0.284**	**0.379**	**0.495**	**0.673**
7 (15)	0.913	0.936	0.966	1.042	0.510	0.561	0.599	0.738	**0.241**	**0.342**	**0.465**	**0.629**
8 (6)	0.631	0.703	0.765	0.847	0.415	0.539	0.640	0.769	**0.243**	**0.368**	**0.524**	**0.673**
9 (20)	1.550	1.585	1.611	1.666	0.589	0.653	0.715	0.817	**0.244**	**0.358**	**0.521**	**0.696**
10 (12)	0.560	0.543	0.549	0.540	0.279	0.288	0.342	0.449	**0.131**	**0.184**	**0.244**	**0.327**
11 (6)	0.422	0.405	0.401	0.417	0.240	0.261	0.281	0.321	**0.088**	**0.121**	**0.168**	**0.251**
12 (10)	0.982	0.974	0.953	0.966	0.722	0.733	0.756	0.808	**0.301**	**0.400**	**0.539**	**0.699**
13 (5)	0.804	0.846	0.874	0.948	0.178	0.204	**0.256**	0.371	**0.127**	**0.179**	0.265	**0.370**
14 (18)	1.976	2.214	2.331	2.372	0.669	0.708	0.894	1.115	**0.248**	**0.361**	**0.554**	**0.806**
15 (10)	0.744	0.838	1.011	1.175	0.245	0.278	0.363	0.457	**0.131**	**0.170**	**0.252**	**0.339**
16 (10)	0.806	0.886	0.912	1.001	0.314	0.331	0.399	0.490	**0.126**	**0.187**	**0.275**	**0.402**
17 (11)	0.819	0.821	0.839	0.860	0.502	0.528	0.594	0.711	**0.229**	**0.309**	**0.399**	**0.586**
18 (16)	1.203	1.197	1.193	1.167	0.701	0.725	0.758	0.821	**0.297**	**0.391**	**0.505**	**0.658**
19 (29)	1.276	1.317	1.397	1.494	0.940	0.940	1.053	**1.221**	**0.759**	**0.890**	**1.038**	1.238
20 (7)	0.581	0.623	0.787	0.892	0.157	0.182	0.278	0.389	**0.066**	**0.108**	**0.212**	**0.351**
21 (12)	0.796	0.824	0.834	0.850	0.524	0.526	0.593	0.680	**0.185**	**0.259**	**0.367**	**0.508**
22 (10)	0.882	0.929	0.980	1.108	0.835	0.818	**0.929**	**1.088**	**0.580**	**0.737**	0.944	1.211
23 (10)	1.510	1.444	1.226	1.154	0.395	0.448	0.547	0.650	**0.191**	**0.295**	**0.429**	**0.572**
24 (18)	1.352	1.165	1.004	1.036	0.572	0.653	0.736	0.907	**0.403**	**0.546**	**0.694**	**0.879**
25 (17)	3.172	3.200	3.145	3.120	0.859	0.889	1.107	1.504	**0.346**	**0.509**	**0.816**	**1.258**
26 (12)	0.925	0.956	0.985	1.057	0.567	0.628	**0.702**	**0.833**	**0.417**	**0.549**	0.707	0.929

We evaluate the prediction performance of GNN, CNN, and RNN encoders for the number of available parking lots per cluster. Table II shows the MAEs for each cluster. In the table, the numbers in brackets denote the number of parking lots in each of the clusters. Since the MAEs of ARIMA and SARIMA are larger than those of our models (see Table I), we do not show these results.

The predicted errors are quite small even for the 120 minutes time step. Since absolute errors are mostly less than one, each model often accurately predicts the number of available parking lots. The GNN and CNN encoders outperform the RNN encoder model for all clusters at each time step as they are able to capture the spatial perspective more precisely than the RNN models. Comparing the GNN encoder with the CNN encoder, the GNN encoder achieves higher accuracy than the CNN encoder in most cases.

From these results, we can see that our graph-to-sequence model accurately predicts the number of available parking lots in the entire parking area and outperform baselines.

Impact of time steps. Our sequence-to-sequence model can predict accurately when we predict near time step from the current (e.g., 15 min). As time steps become further from the current, the accuracy decreases. Therefore, we evaluate accuracy on time steps.

Figure 4 shows MAE varying with time steps for our encoder models. From these results, the accuracy deteriorates gradually and almost linearly step by step for all models. If we predict further time steps than 120 minutes, we can estimate the prediction performance from this tendency. GNN always achieves the best performance among models.

Error analysis at each time step. We finally analyze the transitions of the difference between measured and predicted values in GNN, CNN, and RNN encoders. Figure 5 shows the difference between measured and predicted values on RNN, CNN, and GNN encoders. Dashed lines represent measured values, while solid lines represent predicted values, and the same colors represent the same time period. So, if dashed and solid lines with the same color are vertically close, the prediction is accurate. We note that each solid line with each color indicates predicted values for eight steps (i.e., for 2 hours), and two neighbor solid lines with different colors are not overlapped.

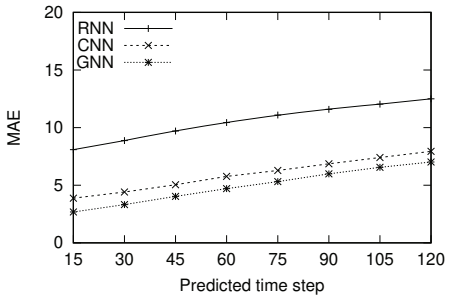
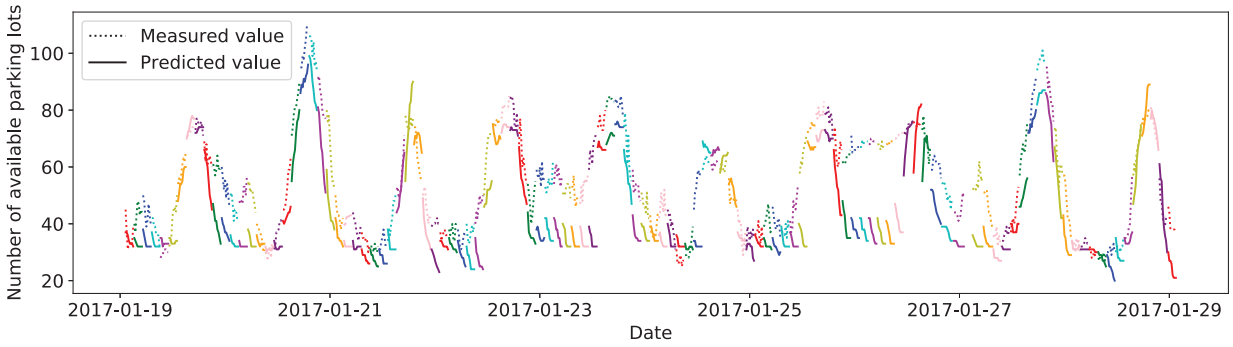
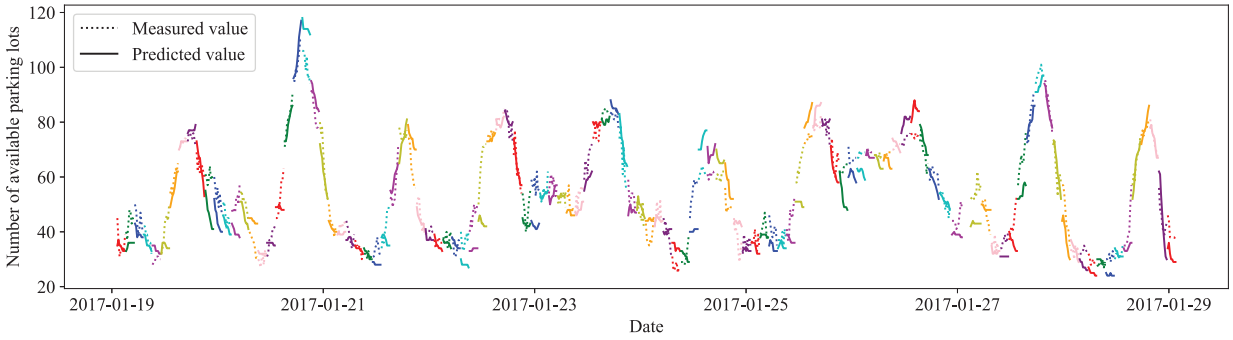


Fig. 4: Impact of time steps

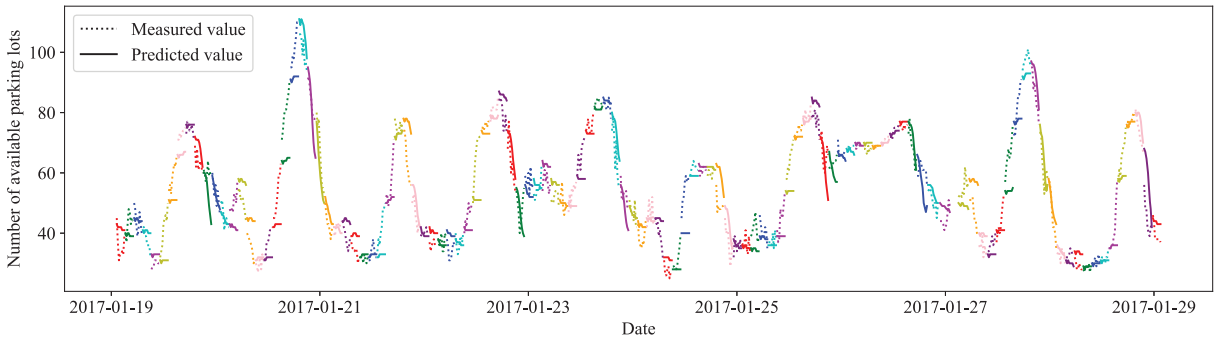
From the results, we can see that the predicted values generally have a similar trend as measured values. Figure 5(a) shows that the RNN encoder predictions often underestimate available parking lots from measured values and have a large difference between predicted and measured values. In our use case, underestimated predictions are inadequate for citizens as they would be guided to streets with occupied parking lots. Figure 5(b) shows that predicted values in the CNN encoder are closer to measured values than that in the RNN encoder. However, the transition sometimes follows an opposite direction, for example, green and blue lines around 2017-01-26. In the case of the CNN encoder, we can notice from the figures that it cannot capture temporal perspective at some points. Finally, Figure 5(c) shows that predicted values



(a) RNN encoder



(b) CNN encoder



(c) GNN encoder

Fig. 5: Prediction performance at each time step. Each solid line represents two hours prediction. The solid and dashed lines with the same colors represent the same time periods.

in the GNN encoder are also close to measured values. The GNN encoder does not present opposite transitions. However, it consecutively outputs the same predicted values in several points. Hence, we can observe that our GNN encoder is conservative.

VI. Smartphone Application

We apply our graph-to-sequence model to a smartphone application in order to provide accurate parking predictions to citizens and tourists. Figure 6 shows a screenshot of our smartphone application. The application shows how many parking lots are currently available and will be available in 15–120 minutes.

We developed our application using the Flutter framework for deploying it on both iOS and Android platforms. We run prediction models on a server, and then smartphones access the server for obtaining and displaying the prediction results. The server accesses the real-time parking information and the historical ones within 24 hours through the SmartSantander APIs. This smartphone application provides an intuitive way to view the current and future availability of parking lots.

VII. Conclusion

Our research focused on predicting parking availability using the SmartSantander IoT deployment. To achieve this, we developed a graph-to-sequence neural network model that

Fig. 6: Screenshot of our smartphone application for parking prediction with SmartSantander

forecasts the number of available parking spots in the entire city center and on each street across multiple time intervals. Our model was trained on a three years dataset of real outdoor parking data from SmartSantander, and we found that it accurately predicted parking availability with high precision, making it suitable for a parking prediction service in the city.

As part of our future work, we plan to develop online learning methods that can capture real-time trends, particularly in response to significant changes such as those resulting from road and street renovations. Additionally, we aim to improve prediction performance by incorporating real-time data from user feedback through user interfaces. Lastly, we intend to leverage additional features such as traffic volume and weather conditions, which are available through SmartSantander, to further enhance the accuracy of our parking predictions.

ACKNOWLEDGEMENT

This research is partially supported by the Grant-in-Aid for Scientific Research JP20H00584 and JP22H03700. We thank to SmartSantander developer group.

REFERENCES

[1] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[2] F. Caicedo, C. Blazquez, and P. Miranda, "Prediction of parking space availability in real time," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7281–7290, 2012.

[3] A. Camero, J. Toutouh, D. H. Stolfi, and E. Alba, "Evolutionary deep learning for car park occupancy prediction in smart cities," in *LION*, 2018, pp. 386–401.

[4] X. Chen, "Parking occupancy prediction and pattern analysis," *Stanford univ. Tech. Rep.*, 2014.

[5] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv*, 2014.

[6] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *ICML*, 2017, pp. 933–941.

[7] K. Harada, Y. Sasaki, and M. Onizuka, "Miscela: Discovering correlated attribute patterns in time series sensor data," in *MDM*, 2019, pp. 72–80.

[8] ——, "Miscela: discovering simultaneous and time-delayed correlated attribute patterns," *Distributed and Parallel Databases*, vol. 39, pp. 637–664, 2021.

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

[10] R. Jiang, D. Yin, Z. Wang, Y. Wang, J. Deng, H. Liu, Z. Cai, J. Deng, X. Song, and R. Shibasaki, "Dl-traff: Survey and benchmark of deep learning models for urban traffic prediction," in *CIKM*, 2021, pp. 4515–4525.

[11] H. S. Jomaa, J. Grabocka, L. Schmidt-Thieme, and A. Borek, "A hybrid convolutional approach for parking availability prediction," in *IJCNN*, 2019, pp. 1–8.

[12] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *ICLR*, 2018.

[13] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," *arXiv*, 2015.

[14] T. Lin, H. Rivano, and F. Le Mouël, "A survey of smart parking solutions," *IEEE T-ITS*, vol. 18, no. 12, pp. 3229–3253, 2017.

[15] A. Nandugudi, T. Ki, C. Nuessle, and G. Challen, "Pocketparker: Pocketsourcing parking lot availability," in *UbiComp*, 2014, pp. 963–973.

[16] C. Pflügler, T. Köhn, M. Schreieck, M. Wiesche, and H. Krcmar, "Predicting the availability of parking spaces with publicly available data," *Informatik*, 2016.

[17] L. Rottkamp and M. Schubert, "A time-inhomogeneous markov model for resource availability under sparse observations," in *SIGSPATIAL*, 2018, pp. 460–463.

[18] L. Sanchez, L. Muñoz, J. A. Galache, P. Sotres, J. R. Santana, V. Gutierrez, R. Ramdhany, A. Gluhak, S. Krco, E. Theodoridis *et al.*, "Smartsantander: Iot experimentation over a smart city testbed," *Computer Networks*, vol. 61, pp. 217–238, 2014.

[19] Y. Sasaki, "A survey on iot big data analytic systems: Current and future," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1024–1036, 2021.

[20] Y. Sasaki, K. Hori, D. Nishihara, S. Ohashi, Y. Wakuta, K. Harada, M. Onizuka, Y. Arase, S. Shimojo, H. Hongdi *et al.*, "Smart city data analysis via visualization of correlated attribute patterns," in *EDBT*, 2021, pp. 650–653.

[21] Y. Sasaki, Y. Ishikawa, Y. Fujiwara, and M. Onizuka, "Sequenced route query with semantic hierarchy," in *EDBT*, 2018, pp. 37–48.

[22] W. Shao, Y. Zhang, B. Guo, K. Qin, J. Chan, and F. D. Salim, "Parking availability prediction with long short term memory model," in *GPC*, 2018, pp. 124–137.

[23] T. Tiedemann, T. Vögele, M. M. Krell, J. H. Metzen, and F. Kirchner, "Concept of a data thread based parking space occupancy prediction in a berlin pilot region," in *Workshops at AAAI*, 2015.

[24] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *KDD*, 2020, pp. 753–763.

[25] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *IJCAI*, 2019, pp. 1907–1913.

[26] X. Xiao, Z. Jin, Y. Hui, Y. Xu, and W. Shao, "Hybrid spatial–temporal graph convolutional networks for on-street parking availability prediction," *Remote Sensing*, vol. 13, no. 16, p. 3338, 2021.

[27] D. Xu, W. Cheng, D. Luo, X. Liu, and X. Zhang, "Spatio-temporal attentive RNN for node classification in temporal attributed graphs," in *IJCAI*, 2019, pp. 3947–3953.

[28] H. Yang, C. Liu, Y. Zhuang, W. Sun, K. Murthy, Z. Pu, and Y. Wang, "Truck parking pattern aggregation and availability prediction by deep learning," *IEEE T-ITS*, 2021.

[29] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *IJCAI*, 2018, pp. 3634–3640.

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

[31] D. Zhao, C. Ju, G. Zhu, J. Ning, D. Luo, D. Zhang, and H. Ma, "Mepark: Using meters as sensors for citywide on-street parking availability prediction," *IEEE T-ITS*, 2021.

[32] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *AAAI*, 2020, pp. 1234–1241.