

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Rental Prediction in Bicycle-Sharing System Using Recurrent Neural Network

Eric Hsueh-Chan Lu¹ and Zhan-Qing Lin²

¹Department of Geomatics, National Cheng Kung University, Tainan, Taiwan

²Department of Geomatics, National Cheng Kung University, Tainan, Taiwan

Corresponding author: Eric Hsueh-Chan Lu (e-mail: luhc@mail.ncku.edu.tw).

This research was supported by Ministry of Science and Technology, Taiwan, R.O.C. under grant no. MOST 108-2621-M-006-008-.

ABSTRACT As the rapid development of smart city and Internet of Things (IoT), related research issues have attracted much attention from industry and academia around the world, and Bicycle-Sharing System (BSS) is one of the thriving applications of smart transportation system. BSS is a system that allows users to rent the bicycle from any automatic rental station. If there're some stations that don't have enough bicycles or free places, then it is usually handled by dedicated vehicles to rebalance the bicycles. Thus, predicting the rental (i.e. the number of renting or returning bicycles) from users in the future is important to improve the service quality. This research uses Recurrent Neural Network (RNN) to predict the rental from users. The RNN consists of three parts: period, closeness, and general. Each of them represents the historical records in different time intervals in the past time respectively. After inputting the historical rental data into RNN and the training process, we can predict the bicycle rental in the coming day by inputting the rental records of the past time into RNN. Finally, we compare the effectiveness among this and the method of Poisson by real YouBike data and prove that our model outperforms it.

INDEX TERMS Bicycle-sharing system, rental prediction, recurrent neural network, deep learning.

I. INTRODUCTION

As the rapid development of smart city and Internet of Things (IoT), related research issues have attracted much attention from industry and academia around the world, and *Bicycle-Sharing System (BSS)* is one of the thriving applications of smart transportation system. BSS is a system that allows users to rent a bicycle from any automatic rental station, have a short travel around the city, and return it at any station. It aims to reduce urban traffic congestion, environmental pollution, and energy consumption by encouraging people to ride public bicycles as short-distance transportation and reduce the use of private vehicles. In addition to commuting, it can also be used for traveling or leisure and entertainment. There're many BSSs around the world, such as Vélib' in Paris, Citi Bike in New York, and YouBike in Taipei.

Most of the BSSs have *docks* at the automatic rental station to lock the bicycles, so users should rent and return the bicycles to the dock at the station belonging to the same BSS. Therefore, some of the stations in BSS may be lack or full of bicycles at peak hours such as the end of school hours during the day, making some people unable to rent or return the bicycles. Thus, the key point for success of BSS is having a

good quality of service for users, which means that the stations should have enough bicycles to be rented and free places for users to return the bicycles. If not, then they cannot satisfy all the users' needs. The problem is usually solved by dedicated vehicles to rebalance the bicycles in the middle of the night, that is, picking up the bicycles from the stations which have too many bicycles and unloading them to the stations having too few bicycles. However, there are still some problems with this method. First, the trucks are rebalancing in the middle of the night, so they cannot solve the unsatisfied demands during the day. Furthermore, even though we can monitor the remaining number of bicycles at each station in real time, it may be too late to assign vehicles to rebalance when the station has already become empty or full. As a result, we need a prediction method to predict the number of renting or returning bicycles from users in the future. Most of the existing solutions from the literature are statistic models [1, 3, 12, 13, 20, 21, 24, 25, 27], but they have a disadvantage: they need to assume the rental behavior from users in BSS to be random events. Besides, Poisson, one of the most commonly used statistical methods in the literature, is not accurate enough since it just calculates the averages in the past and does

not consider the spatial characteristics between different stations (including the POIs near the stations). Accordingly, we need a more accurate method, and that is why we apply neural network.

Due to the motivation mentioned above, we investigate the problem of solving users' unsatisfied demands in BSS. Our objective is to predict the rental behavior accurately before the trucks rebalance the bicycles so that the trucks can rebalance the bicycles based on the predicted results and thus minimize the user's unsatisfied demand. We propose a modified *Recurrent Neural Network (RNN)* to predict the number of renting or returning bicycles from users at each station in the future (we called it "*the rental from users*" briefly). The RNN consists of three parts: period, closeness, and general, and each of them represents the historical records in different time intervals in the past time respectively. We input the historical rental data into RNN, and after iterative training, we can obtain a set of optimal parameters of RNN. As a result, we can predict the bicycle rental in the coming day by inputting the rental records of the past time into the RNN.

The contributions of this paper are listed below:

- 1) We enhance and extent the RNN model from the literature, which predict the parking availability at that time, to be able to predict the rental from users for a long time in the future.
- 2) We consider both temporal and spatial characteristics of the stations in the model to improve the prediction accuracy.
- 3) Considering the periodicity of time, we use sine and cosine to model the temporal feature and design a "*POI feature vector*" to express the spatial characteristics of each station well.
- 4) We use the real data – YouBike data in Taipei – to conduct experiments. The experimental results show that our method can win the method of Poisson up to 15.38%.

The rest of this paper is organized as follows. First, we review the related works about BSS in Section II. Next, Section III is the problem statement for explaining the various terms in BSS. Then, the method we propose for rental prediction is explained in detail in Section IV. After that, we evaluate the performance of our method and compare the effectiveness between this and the other method in Section V. Finally, we conclude our findings and briefly discuss some future works in Section VI.

II. RELATED WORK

In this section, we reviewed the important literature related to Bicycle Bicycle-Sharing System (BSS). In order to avoid the problem of having too few or too many bicycles at some stations, which causes users' *unsatisfied demand*, we need to analyze and predict the bicycle rental behavior beforehand for the subsequent bicycle rebalancing problem. Therefore, we divide the literature into two topics: bicycle rental behavior analysis and bicycle rental behavior prediction.

A. Bicycle Rental Behavior Analysis

The research of bicycle rental behavior analysis can be divided into three categories: 1) explore the factors that affect the usage of BSS, including weather, spatial relationship, temporal characteristics, topology, and rent fee, 2) investigate the strategic planning and recommend the location, quantity, or capacity of new bicycle stations, 3) explore the benefits of BSS, including traffic and health.

In the first part, we introduce the literature analyzing the factors that affect the bicycle rental behavior of users. First, Borgnat *et al.* [3] explained that the large dataset of BSS can be used to analyze the movements of using bicycles in the city, including the factor of weather. Bachand-Marleau, Lee, and El-Geneidy [2] discovered the influence of spatial relationship on the frequency of use. They found that the proximity of home to the docking stations would have the greatest effect on the probability of using BSS. Therefore, they also provided the strategic planning: increasing the number of stations in the residential neighborhoods to maximize the usage of BSS. Garcia-Palomares, Gutierrez, and Latorre [11] investigated the factor of spatial relationship as well. They proposed a GIS (Geographic Information System) based approach to calculate the spatial distribution of the potential demand for trips and discovered that the relationship between the location of the bicycle stations and potential demand (population, activities, and public transport stations) is the key factor for the success of BSS. Thus, they determined the optimal location and capacity of bicycle stations based on the characteristics of the demand. Rudloff and Lackner [23] applied Poisson model, negative binomial model, and hurdle model to model the overall demand and found the influence of weather and "full or empty neighboring stations" on the demand in BSS. In addition to weather and spatial relationship, Faghili-Imani *et al.* [7] still investigated the factor of temporal characteristics. They examined the effect of meteorological data, temporal characteristics, bicycle infrastructure, land use, and built environment attributes on the demand of BSS and found that each factor has different predictable effects on bicycle rental behavior. Based on this, they provided recommendations for new stations, for instance, increasing the number of bicycles stations or neighboring the bicycle facilities or restaurants to increase the usage of BSS. Frade and Ribeiro [8] explored two effects on the rental behavior: one is the distance between the start and the end point, and the other one is topography. They discovered that the stations are specially deployed near the steep slope in their study area so that the cyclists can ride a bicycle in one direction and use other transport modes (such as buses) for the opposite direction. Finally, apart from the network topology, Jurdak [15] also studied the impact of cost on the behavior of users in BSS. He found that the spatial topology of BSS is not strongly affecting the usage patterns, though the cost has higher impact on the willingness of short-term users.

Secondly, there're some articles investigating the strategic planning in BSS. O'Brien, Cheshire, and Batty [19] published

the first thesis that analyzes the data from 38 BSSs around the world (including YouBike and CityBike in Taiwan). They compared three kinds of characteristics (aggregate, spatial, and temporal) of the BSSs and proposed a system classification based on temporal characteristics. These features can provide suggestions for the location of new docking stations. Lin and Yang proposed a model in 2011 to determine the number and the locations of bicycle stations [17] and proposed a hub location inventory model with Chang in 2013 to additionally provide recommendations on the capacity of the station [18]. Lastly, after studying the factors affecting the bicycle rental behavior, the authors of [8] published the research for strategic planning in BSS in the next year, including locating the stations and defining the capacity of the stations [9].

Finally, there is some literature exploring the benefits of BSS. One is from Jäppinen, Toivonen, and Salonen [14]. They modeled a hypothetical BSS and quantified its spatial effect on public transport travel times. The other one is the research from Woodcock *et al.* They modeled the impacts of BSS in their study area on the health of its users [26].

B. Bicycle Rental Behavior Prediction

By analyzing the bicycle rental behavior of BSS, we can develop the algorithm for the bicycle rental behavior prediction. There're mainly four kinds of things to predict: 1) inventory level: the number of remaining bicycles in the future, 2) rental from users: the number of renting or returning bicycles from users in the future, 3) unsatisfied demand: how many users that can't rent or return bicycles, 4) position of bicycles: at which station each bicycle will be. Predicting these things are all for avoiding the unsatisfied demand. As for the methods, we can divide them into two categories: statistical model and data mining. Statistical model includes regression model, Poisson process, Markov chain, and random walk, and data mining includes pattern mining and deep learning.

Most of the studies use mathematic statistics model to predict. After the bicycle rental behavior analyzing study in the previous part, Borgnat *et al.* [3] used linear regression model considering the weather factor to predict the inventory level of BSS and further calculated the rental and unsatisfied demand from users. Alvarez-Valdes *et al.* [1] assumed the withdrawal and return rates as two independent and inhomogeneous Poisson processes, estimated the number of withdrawals and returns from the historical records, and computed the unsatisfied demand by Skellam distribution and conditional probability. The other studies using mathematic statistics model also apply Poisson process but additionally introduce the concept of Markov chain. Raviv and Kolka [20] proposed an inventory model, defined a function called User Dissatisfaction Function (UDF) and used Continuous-Time Markov Chain (CTMC) to represent the inventory of each station. Zhang *et al.* [27] applied the model from [20] but added other constrains that are closer to reality additionally. Regue and Recker [21] proposed two predictive models to

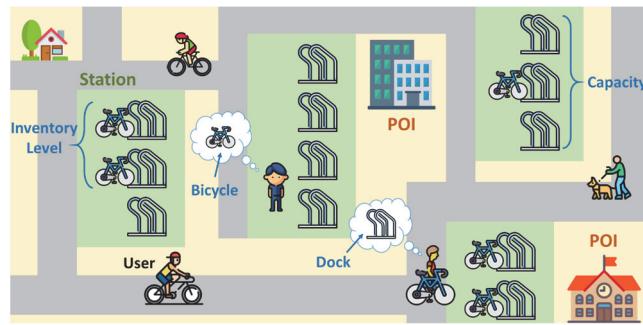


FIGURE 1. The scenario of BSS (including POI). Source: Icon made by Freepik & DinosoftLabs from www.flaticon.com.

predict the demand from users and the inventory of the stations respectively. Gast *et al.* [12] proposed a queuing theoretical time-inhomogeneous model, which models a station as a queue to estimate the inventory level of the station. Ghosh *et al.* [13] also regarded the customer arrivals at the stations as Poisson process and predicted the changing of the remaining number of bicycles by Markov Decision Process (MDP). Schuijbroek, Hampshire, and van Hoeve [24] modeled the stochastic demand by considering the inventory at each station to be a non-stationary queuing system with finite capacity. Instead of Poisson process and Markov chain, Tomaras *et al.* [25] used random walk to predict the bike demand changes at the stations.

The rest of the literature uses other kind of methods to predict. Cagliero *et al.* [6] proposed a pattern called Occupancy Monitoring Pattern (OMP) and used it in pattern mining to detect the situations of dock overload. Based on Artificial Neural Networks (ANN) and fuzzy logic, Caggiani and Ottomanelli [5] proposed a modular Decision Support System (DSS) to predict the demand. In 2018, they proposed a new comprehensive dynamic bicycle redistribution methodology with Camporeale and Szeto [4]. They clustered the stations in terms of spatio-temporal correlation and then predicted the number and position of bicycles by Nonlinear AutoRegressive neural network with eXogenous inputs (NARX) approach from Leontaritis and Billings [16]. Finally, Gao and Lee [10] proposed a new hybrid approach combining a Fuzzy C-Means (FCM)-based Genetic Algorithm (GA) with Back Propagation Network (BPN) to effectively predict the rental demand.

III. PROBLEM STATEMENT

In this section, we formally define the various terms in the problem we deal with. Our topic is to predict the rental from users in Bicycle-Sharing System (BSS), so we draw a picture to demonstrate the whole scenario of BSS (including the required POI information) in Fig. 1, and all of the notations about BSS and POI are summarized in Table I.

Definition 1. Bicycle: Bicycles in BSS are available for shared use to users on a short-term trip. The total number of bicycles in BSS is constant.

Definition 2. Dock: The dock is a special bicycle rack that locks the bicycle and is only released by computer control.

TABLE I
THE NOTATIONS ABOUT BSS AND POI

Notation	Description
F_r	The rental data
t	The time interval
s, S	A station, set of station
b, d	The number of bicycles, the number of docks
cap	The capacity of a station
id, na	The id and name of a station
lat, lng	The latitude and longitude of a station
poi, POI	A POI, set of POIs
ty, TY	A type of POI, set of POI types

After finishing riding the bicycle, the user should return it at the dock.

Definition 3. Rental Data: The rental data F_r is the number of renting or returning bicycles from users at certain station during a time interval t , and we also call them “*the rental from users*”.

Definition 4. Station: A station s is a place for renting or returning bicycles. Each $s \in S$ has various attributes: 1) $s.b$ is the remaining number of bicycles at a certain time point, and it is also referred to as *inventory level*, 2) $s.d$ is the remaining number of docks at a certain time point, 3) $s.cap$ is the total number of docks and is also called *capacity*. In normal case, for each time point, $s.b + s.d = s.cap$, 4) $s.id$ and $s.na$ is the unique ID and the name of s respectively, 5) $s.lat$ and $s.lng$ is the latitude and longitude of s respectively. Besides, each $s \in S$ has F_r in each t .

Definition 5. User: The users mean those who use BSS for riding the bicycles.

Definition 6. BSS: As mentioned in Section I, BSS is a system that allows users to rent a bicycle from any automatic rental station, have a short travel around the city, and return it at any station. The BSS network consists of all the stations, including bicycles and docks, and users.

Definition 7. POI: A POI (Point Of Interest) poi is a particular point, spot, or landmark recorded in electronic map. It can be a school, a restaurant, a company, or a park, and so on. All the POIs compose POI network. Each $poi \in POI$ has its own ID $poi.id$, name $poi.na$, type $poi.ty$ (such as food, mall, or business), and coordinate $poi.lat$ and $poi.lng$. There are usually various POIs around the station, which affect the station’s spatial characteristics.

Research Purpose: Given BSS network and POI network, our research purpose is to predict F_r in the future using $F_r, t, s.lat, s.lng$, and POI such that the RMSE between real F_r and predicted F_r is minimized.

IV. PROPOSED METHOD

In the following, we introduce our proposed method for rental prediction. We describe our method in three parts. First, we show the system framework of our method. Then, we introduce the required data for the prediction and explain how to convert the data into the input features for the prediction model. Finally, our proposed model is explained in detail.

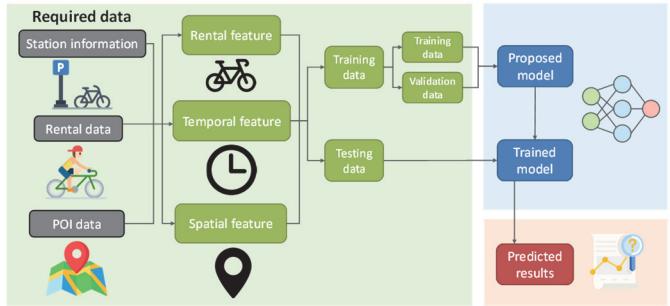


FIGURE 2. The framework of our research. Source: Icon made by Smashicons, Freepik, & dDara from www.flaticon.com.

A. System Framework

Our system framework is illustrated in Fig. 2. The whole framework comprises three parts: input features, model, and output. First, we need to extract the features we need from the data we have obtained. Then, we will divide the data into training and testing data, where the training data will be further split the validation data out. Next, the training and validation data are fed into the neural network model for training. We refer to the model in the literature from Rong *et al.* [22], and the structure of our model will be introduced in the end of this section. After the training process, we can put the testing data into the trained model, and it will output the predicted results.

B. Feature Extraction

Before the step to the prediction, we should prepare the required data first. We have three kinds of required data: station information, rental data, and POI data.

- 1) Station information: It contains basic information about each station $s \in S$ in the specific BSS, including $s.id$, $s.na$, $s.cap$, $s.lat$, and $s.lng$. The meanings for each notation are described in Section III.
- 2) Rental data: The rental data (F_r) is the main data for predicting the rental from users. It is the historical data of all the stations in the specific BSS and gives us the clues for predicting future trends. For each station $s \in S$, it contains F_r for all the study time zones, and each one records the number in a specific time interval t . The minimum time interval of data from the literature is 30 minutes, so in our case, t is 30 minutes, and in a whole day, it will record the number between 00:00 to 00:30, 00:30 to 01:00, until 23:30 to 24:00.
- 3) POI data: It is the POI data in the study area. Each $poi \in POI$ contains $poi.na$, $poi.ty$, $poi.lat$, and $poi.lng$.

In the following, we explain how to convert the data into the input features, which will be fed into the model. Our input features consist of three main parts: rental feature, temporal feature, and spatial feature. Finally, we briefly describe what kinds of features used in the literature are.

1) RENTAL FEATURE

Rental feature is rental data F_r . As for its time and space information (in which time interval and which station), it is described in temporal and spatial features.

2) TEMPORAL FEATURE

Temporal feature indicates what time the information is in. Here we assume that the time of F_r is in time interval t_i , and we set two kinds of temporal features: week feature and time feature. For the week feature, it represents which day of the week t_i is. It is worthwhile to mention that, instead of setting it as 1 to 7, we consider the cyclic characteristics of the week (that is, after the whole week from Monday to Sunday, the week begins on Monday again) and split it into sine and cosine. If you set it as 1 to 7 (Monday to Sunday), there will be a problem: in neural network, the closer number will be regarded as the stronger correlation, so 1 is close to 2 and far from 7, but it is incorrect. Monday is followed by Sunday. Therefore, we should use sine and cosine to split the week w (1 to 7) into $\sin(2\pi \cdot w/7)$ and $\cos(2\pi \cdot w/7)$, denoted as \sin_w and \cos_w , so (\sin_w, \cos_w) implies the coordinate in the Cartesian coordinate system in the Euclidean plane, corresponding to the seven points on the circumference that dividing the unit circle into seven equal parts. Fig. 3 shows the concept of the coordinate.

As for the time feature, if a day is divided into n parts, then the length of each time interval t_i is $1/n$ day. As mentioned in the beginning of this section, the minimum time interval of data in the thesis we refer to is 30 minutes, so we set n as 48, and then the length of each t_i will be 30 minutes. Similar to the week data, the time data should also be split into sine and cosine, denoted as \sin_t and \cos_t .

3) SPATIAL FEATURE

Spatial feature implies which station the information belongs to, so the most intuitive way to express is the latitude and longitude of the station: $s.lat$ and $s.lng$, and we adopt different kinds of normalization on them. However, we think that its spatial characteristics are more important. Because the POIs near the station would affect the spatial characteristics of the station, such as being full of POIs in food type or locating near school, which result in different rental behaviors, we design a “POI feature vector” (denoted as V_{poi}) that represents the spatial characteristics of the station, and inspecting the POIs near the station (we set the threshold as 100 meters) is for this purpose. Given k POI types, each station s has a V_{poi} containing k elements, which represent the effect level of the corresponding type to s . The more POIs in certain type near s

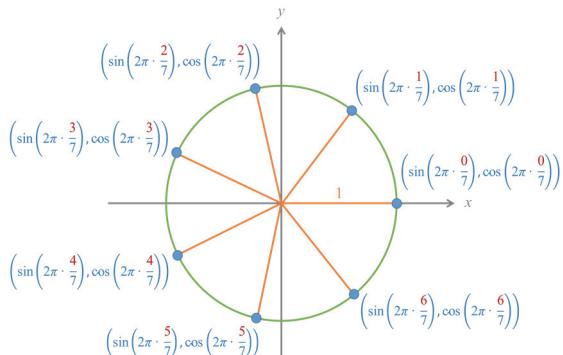


FIGURE 3. The concept of the coordinates on the unit circle.

or the closer the POI in certain type to s is, the larger the corresponding number will be. The following explains how to calculate V_{poi} of any station s :

- 1) Collect the POI data in the study area. Each $poi \in POI$ contains $poi.na$, $poi.ty$, $poi.lat$, and $poi.lng$.
- 2) Find all the POIs within 100 meters from s , and then count the number of POIs for each $ty \in TY$, where TY is the set of all POI types.
- 3) In order to take the effect of the distance into account, for each $ty \in TY$, sum up the reciprocals of the distances between s and each POI in type ty . Because the reciprocal is a small number, we multiply it by 100 to make it bigger than 1.
- 4) Due to the discrepancy among the types (the values of common types such as food are usually higher than those of uncommon types such as entertainment), we take nature log of all the elements of the vector. Note that in order to avoid the error of $\ln(0)$, we should add 1 to the numbers first before taking the log.

Here gives an example for calculating a POI feature vector, and the scenario is shown in Fig. 4. Assume that there are only three kinds of types: $TY = \{\text{food}, \text{park}, \text{sport}\}$, and then V_{poi} should have three elements, corresponding to these three types respectively. In Fig. 4, we can know that there are two POIs in type food and one POI in type sport near station s . For each type $ty \in TY$, sum up the reciprocals of the distances between s and each POI in type ty and multiply them by 100, and V_{poi} will be like this:

$$\left[\frac{1}{50} \times 100 + \frac{1}{10} \times 100, 0, \frac{1}{100} \times 100 \right] = [12, 0, 1]$$

Then, add 1 to the numbers and take the nature log:

$$[\ln(12+1), \ln(0+1), \ln(1+1)] \approx [2.56, 0, 0.69]$$

This is the POI feature vector of s : the effect level of type “food” to s is 2.56, “park” is 0, and “sport” is 0.69.

4) FEATURES FROM LITERATURE

Compared to the features from the literature, our features are very different from theirs. In the literature, they use five kinds of features: mobility features, meteorological features, POI-related features, holiday event features, and time t_i . The last feature t_i represents that the rest of the features are all record the data at time t_i . Mobility features contain the data of visitors located in the certain parking lot area; meteorological features contain the data of temperature, weather and wind speed; POI-

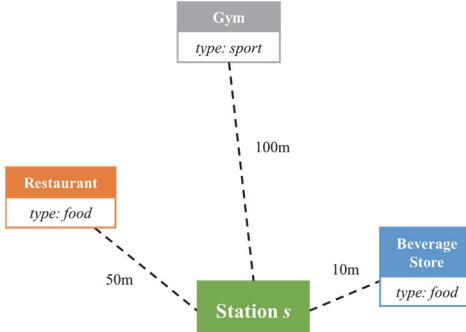


FIGURE 4. An example for calculating the POI feature vector.

related features indicate the type of POI where the certain parking lot is located; holiday event features record whether time t_i is a weekday or weekend. They utilize these features to predict the real-time parking availability of parking lots throughout the city. They have done some simple experiments to prove that the features and parking availability are related, so that's why they choose these features to do the prediction.

C. Prediction Model

In order to predict the rental from users in the future, we apply RNN (Recurrent Neural Network) to solve this task. RNN is a model that has been successfully used for applying to sequence learning tasks, and more specifically, we use LSTM (Long Short-Term Memory), which is an architecture of RNN that can model the temporal properties well and has been widely employed in many fields. We also introduce our model first, and then we briefly describe the model structure from the literature.

1) MODEL STRUCTURE

Using the features mentioned above, we can input them to the model and get the predicted rental in the future. Our model is shown in Fig. 5. We mentioned that the input features contain rental, temporal, and spatial feature, and we compose them into a vector called X_{ti} , representing the rental from users at one of the stations s in time interval t_i . An $X_{ti} = [F_r, \sin_w, \cos_w, \sin_{ti}, \cos_{ti}, s.lat, s.lng, V_{pol}]$ is a unit of a feature. However, an X_{ti} only contains a time interval with thirty minutes, which is not enough for us to predict for a long time in the future such as six hours. Therefore, we concatenate several X_{ti} to form a new vector: X^*_{ti} . In our case, we want to predict the rental within the next six hours, so assuming that t is the time interval of the next thirty minutes, $X^*_{ti} = [X_t, X_{t+1}, \dots, X_{t+11}]$, where the time unit is also thirty minutes.

Next, we will input them into the model according to X^*_{ti} in different t_i . The model consists of three main components which models the temporal *period*, *closeness*, and *general* influence respectively. Period component is the features in periodic time. The period is p , and the length of the period sequence is lp , so the period component contains the features at time $t-lp:p, t-(lp-1)p, \dots, t-p$. Because the bicycle behavior has the periodic change in weeks, we set p as one week and lp as 2, so the features are $X^*_{t-2weeks}$ and $X^*_{t-1week}$, containing 2 time

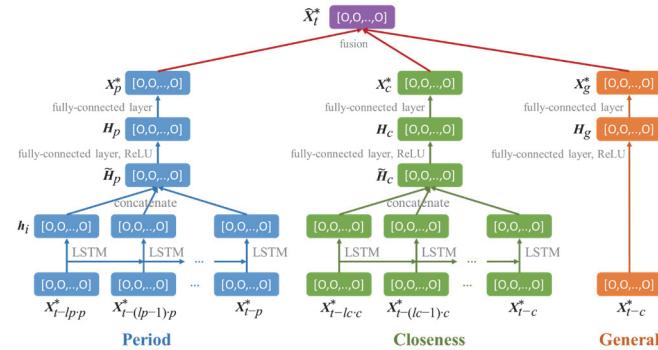


FIGURE 5. The proposed model.

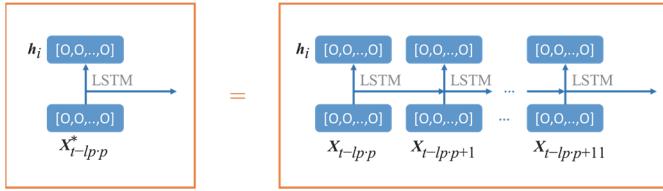


FIGURE 6. Further explanation about LSTM from X^*_{ti} .

intervals. The features are fed into the LSTM and output the hidden state \mathbf{h}_i , where i is the corresponding time of the input feature. We further explain the LSTM from X^*_{ti} in Fig. 6. Take $X^*_{t-lp:p}$ for example, since an X^*_{ti} contains 12 X_{ti} , it contains 12 LSTMs in sequence, rather than just one LSTM. After that, the \mathbf{h}_i are concatenated as $\tilde{\mathbf{H}}_p$, and then $\tilde{\mathbf{H}}_p$ is fed into the fully-connected layer (followed by an activation function: ReLU) to output \mathbf{H}_p . Lastly, \mathbf{H}_p is fed into the fully-connected layer to output X^*_{ti} , which is the predicted rental from users in time interval t to $t+11$ (that is, the next six hours in the future) from the period component. Closeness component is similar to period one. It is the features in recent time. The period is c , and the length of the closeness sequence is lc , so the closeness component contains the features at time $t-lc:c, t-(lc-1)c, \dots, t-c$. We set c as six hours (because an X^*_{ti} contains the time interval with six hours) and lc as 5, so the features are $X^*_{t-30hrs}, X^*_{t-24hrs}, \dots, X^*_{t-6hrs}$. The network structure of closeness component also contains LSTM and two fully-connected layers, which outputs X^*_{ti} . As for general component, it is the features at the present time, meaning that it only contains the feature X^*_{t-6hrs} . The feature is fed into two full-connected layers directly, rather than being fed into LSTM first, and output X^*_{ti} . Finally, the three outputs from each component are merged by fusion to output the final predicted rental from users in time interval t to $t+11$, denoted by \hat{X}^*_{ti} . Note that this X does not contain features other than the rental value. The formula is shown below:

$$\hat{X}^*_{ti} = V_p \circ X^*_{ti} + V_c \circ X^*_{ti} + V_g \circ X^*_{ti}$$

where \circ is the element-wise multiplication, and V_p , V_c , and V_g are learnable 12-element vectors that neural network will learn how to adjust the importance of X^*_{ti} , X^*_{ti} , and X^*_{ti} . Since the output value is numerical data, the loss function we use is MSE.

2) MODEL STRUCTURE FROM LITERATURE

The goal of the literature is to predict the real-time parking availability of parking lots throughout the city. They proposed a model called “Du-Parking”, and Fig. 7 shows the whole model.

Although the model they proposed is complete, but the model still exists something inapplicable for the problem we solve, so we modify many parts of the model to fit our problem. In the following paragraphs, we will describe the differences between our model and the model from the literature.

- Unlike our input features, theirs do not contain the information they want to predict. Their goal is to predict the parking availability, but their input features just contain mobility features, meteorological features, and so on. Because of this, the symbol of their output is

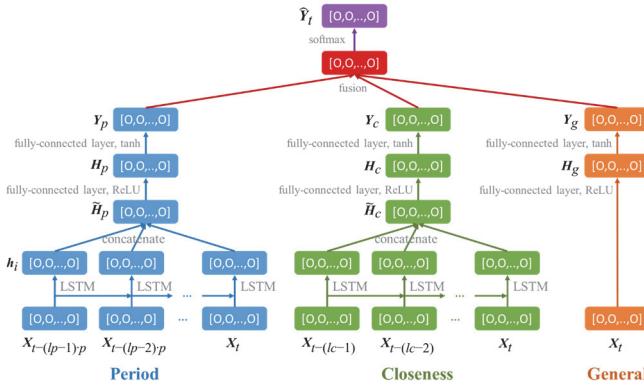
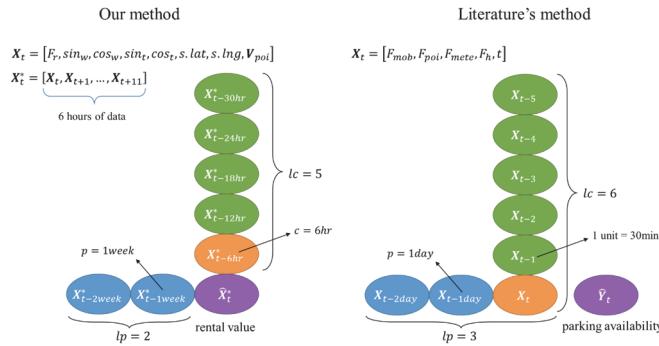


FIGURE 7. The model from literature.

different from input. It is set as \mathbf{Y} . Moreover, the rental from users is a data within a “time interval”, while the parking availability is a data at a “time point”, so each t in our paper represents a 30-minute time interval, while that in the literature represents a single time point.

- 2) They just predict the parking availability at present time, that is, t , so they do not put several X_{ti} into a vector. In addition, because the input features do not contain the data to be predicted, the input data can also contain the data at time t , which makes lp and lc one more than our method. Accordingly, the setting parameters of their model are different from ours. In our model, closeness component contains the features $\mathbf{X}^*_{t-30hrs}$, $\mathbf{X}^*_{t-24hrs}$, ..., \mathbf{X}^*_{t-6hrs} , and lc is 5. Theirs also contains previous five time points, but one time point is thirty minutes, so their features are $\mathbf{X}_{t-2hrs30mins}$, \mathbf{X}_{t-2hrs} , ..., $\mathbf{X}_{t-30mins}$, and \mathbf{X}_t . Note that they contain the feature at present time t , so lc is 6. It does not need to assign the period c here because they see the period as the minimum time interval of data, which is thirty minutes, so their closeness component is $\mathbf{X}_{t-(lc-1)}$, $\mathbf{X}_{t-(lc-2)}$, ..., \mathbf{X}_t . As for the period part, the period p is not one week but one day. Finally, the general component is the features at the present time, meaning that it only contains the feature \mathbf{X}_t . Fig. 8 shows the differences between our method and literature’s method.
- 3) The remaining parts are some small differences. First, in the second fully-connected layers in each component, they have the tanh activation function because the output

FIGURE 8. The differences between our method and literature's method.
(a) Our method. (b) Literature's method.

value in literature is the probability, and thus they need to limit the range of the output into 0 to 1. Similarly, since their prediction is a probability, they need to add the softmax activation function after the output layer. Next, since their output value is categorical data, the loss function they use is cross-entropy. Finally, our number of neurons in each layer is different from the literature due to the different size of input and output. Basing on our input and output size, we can calculate the numbers of neurons in the fully connected layers \mathbf{H}_p , \mathbf{H}_c , and \mathbf{H}_g by min-max normalization. For instance, the numbers of neurons of $\tilde{\mathbf{H}}_p$, \mathbf{H}_p , and \mathbf{Y}_p in the literature are 96, 16, and 3. As for our $\tilde{\mathbf{H}}_p$ and \mathbf{X}_p^* , the numbers of neurons are 408 and 12, so the number of neurons of our \mathbf{H}_p is $12 + (16 - 3) \times (408 - 12) / (96 - 3) \approx 67$.

V. EXPERIMENTAL EVALUATION

This section describes the experiments and analyzes the results of our method. We divide this section into four parts. First, we give a detailed description of our experimental data and settings. Then, we evaluate the performance of our method under various parameters, compare the effectiveness between this and the other method, and carry on the spatial analysis of the result respectively. All the experiments are implemented on a computer with an Intel Core i5-7500 CPU 3.40 GHz and 8 GB memory under Microsoft Windows 10.

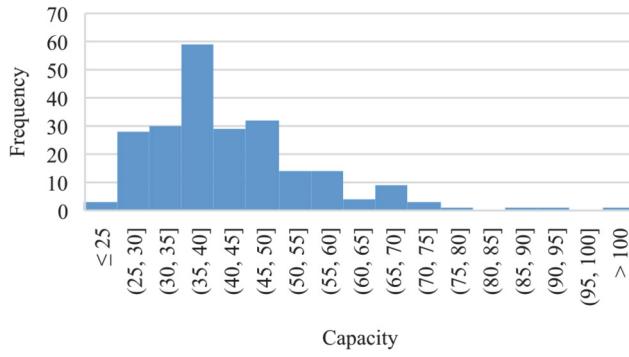
A. Experimental Data and Setting

We obtained three kinds of data in total, which corresponds to the three kinds of information mentioned in Section IV: station information, rental data, and POI data. The information of the experimental data is listed in Table II. The first two kinds of data are the BSS data, and we applied YouBike data from the Taipei City Government; the last one is the POI data in Taipei City, and we obtain it from Google Places API.

- 1) Station information: It is extracted from the real-time data of YouBike in Taipei. The real time data is in CSV format, and it contains many attributes: station ID, station capacity, station inventory level, time, the remaining number of docks, station name, the district where the station locates, the latitude and longitude of the station, and the location of the station. From the data at the same time point, we can easily extract the station information we need.
- 2) Rental data: The rental data is in ACCDB format. It records when each bicycle was rented from which station and when it was returned from which station. In

TABLE II
THE DESCRIPTION OF EXPERIMENTAL DATA

Required Data	Source	Obtained Data	Format	Data Preprocessing Implementation
Station Information	YouBike in Taipei	Real-Time Data	CSV	-
Rental Data	YouBike in Taipei	Rental Data	ACCDB	Java JDK 9 using Eclipse
POI Data	Google Places API	POI Data	JSON	Python 3.5 using Jupyter

**FIGURE 9. Histogram of the stations' capacity.**

order to arrange the rental data in the interval of every thirty minutes, we have to preprocess the data first. We implement the data preprocessing in Java JDK 9 using Eclipse. The time interval of the historical rental data contains entire thirteen weeks, ranging from October 2nd, 2016 to December 31st, 2016. In the arranging process, we found some stations with problems (such as missing rental data) and excluded them. There are remaining 229 stations with complete and usable information in the end, and Fig. 9 shows the histogram of the stations' capacity. After finishing the preprocessing of the rental data, we have total $48 \times 7 \times 13 \times 229 = 1,000,272$ records.

- 3) POI feature: We obtain the POI data in Taipei City from Google Place API. It is in the format of JSON, and it records quite detailed POI information. It contains address, latitude and the longitude, URL of the icon, ID, name, opening hours, URL of the photos, rating, reviews, types, and so on, and Fig. 10 shows an example of POI data. We implement the POI data preprocessing in Python 3.5 using Jupyter. Originally there are 56,380

```

object {3}
  ▶ html_attributions [0]
  ▶ result (18)
    ▶ address_components [6]
      adr_address : <span class="street-address">No. 23-4, Lane 313, Section 2, Zhonghua Road</span>, <span class="locality">Zhongzheng District</span>, <span class="region">Taipei City</span>
      <span class="country-name">Taiwan</span> <span class="postal-code">>100</span>
      formatted_address : No. 23-4, Lane 313, Section 2, Zhonghua Road, Zhongzheng District, Taipei City, Taiwan 100
    ▶ geometry {2}
      ▶ location {2}
        lat : 25.0297459
        lng : 121.5065601
      ▶ viewport {2}
        icon : https://maps.gstatic.com/mapfiles/place_api/icons/restaurant-71.png
        id : 7126586c0bf607282dd2f49bc2d75af1c84031
        name : 秋月日式炸猪排
    ▶ opening_hours {3}
    ▶ photos [10]
      place_id : ChIJ_seOrupQjQRJlqy4mDQ93HE
      rating : 3.9
      reference : CmRRAAAAKimungIZz_pmsgUruruU2eEPHC_62Vp81L1m7rNktlENCp7Q-RErlyN-uxExdIXnnKKV4asSPYg7DYXuoyVU250nfQz6ayVn1lqeE9q157HldTR2xJjPjHT19XhiP7xpBPEhB2x-nc7Lh343pkGcmofggHQ5xjls-OIR2njKzyA407Xm0Y8A
    ▶ reviews [5]
      scope : GOOGLE
    ▶ types [4]
      0 : restaurant
      1 : food
      2 : point_of_interest
      3 : establishment
      url : https://maps.google.com/?cid=8204499917272817857
  
```

FIGURE 10. An example of original POI data.

POIs, and each POI may contain several types. We found that almost every POI has two meaningless types, “point_of_interest” and “establishment”, so after we filtered them out, there are 39,842 meaningful POIs remaining. Next, we only need the POIs within 100 meters of the 229 stations, and thus, after filtering them out, there are 4,270 POIs left, including 84 kinds of types which appear 8,052 times in total. Since the 84 types are too many, we combine them into 10 types manually, and we list the mapping from old to new type in Table III, including the final number of occurrences of new types. However, some of the POIs may become having many same types. Taking Fig. 10 for instance, the original types are restaurant and food, but they correspond to the same new type: food; therefore, the duplicate types should only be counted once. As a result, we can calculate the POI feature vector for each station according to these 10 new types.

As for the experiment settings, the first one we need to assign is the number of neurons. Since we want to predict the rental in the future six hours, X_t^* contains 12 X_{ti} . The reason why we set the prediction period as next six hours is because it is enough to provide the prediction for the following rebalancing procedure. Each X_t^* contains 12 X_{ti} , and each X_{ti} contains 17 elements ($F_r, sin_w, cos_w, sin_{ti}, cos_{ti}, s.lat, s.lng$, and V_{poi}), so the number of neurons of X_t^* is $17 \times 12 = 204$. The number of neurons of the hidden layer \mathbf{h}_i is the same. $\tilde{\mathbf{H}}_p$ is concatenated from two \mathbf{h}_i , so it has 408 neurons. \mathbf{H}_p has 67 neurons, which is mentioned at the end of Secton 4.3.2. The output contains 12 time intervals, so the number of neurons of

TABLE III
 THE NEW TYPES AND THEIR CORRESPONDING OLD TYPES

new type (#)	old type	new type (#)	old type	new type (#)	old type
food (1623)	food restaurant cafe		gas_station bicycle_store hardware_store car_rental locksmith electrician plumber car_wash movie_rental	transportation (391)	transit_station bus_station parking subway_station
	store convenience_store home_goods_store clothing_store bakery hair_care	mall (cont.)			school local_government_office police post_office university embassy fire_station courthouse
	grocery_or_supermarket car_repair electronics_store beauty_salon furniture_store pharmacy travel_agency jewelry_store shoe_store meal_takeaway florist department_store car_dealer		real_estate_agency general_contractor insurance_agency accounting moving_company storage lawyer premise funeral_home cemetery	government (198)	park place_of_worship library church museum art_gallery aquarium painter hindu_temple
	shopping_mall meal_delivery book_store liquor_store laundry pet_store	business (571)	health dentist doctor hospital veterinary_care physiotherapist	park (152) apartment (65)	bar spa night_club movie_theater gym stadium
		hospital (472)			

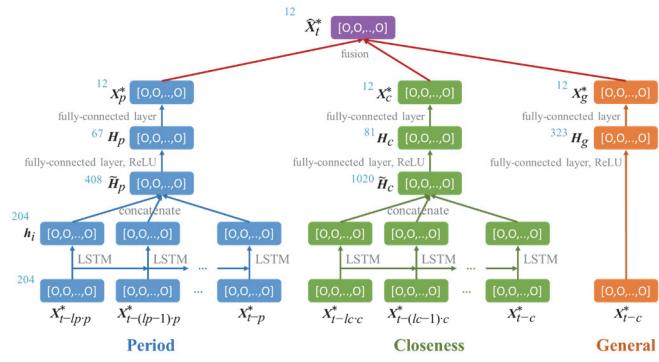


FIGURE 11. The number of neurons of each layer.

the output is 12. The rest of the setting are the same concept, and thus, we directly list the number of neurons of each layer in Fig. 11. The other thing we need to set is the training and testing ratio. We have totally 13 weeks of the historical rental data, and we take first 10 weeks for training and 3 weeks for testing. Because we have the period component, which contains the data far from two weeks ago, our first \hat{X}_t^* should start at the first time interval (00:00 to 06:00) of the third week, and the next \hat{X}_t^* will be from 06:00 to 12:00, and so on. We implement the neural network in Python 3.5 with Keras using Jupyter. For the training process, we set the maximum number of iteration as 1,000 epochs, where an epoch is an iteration over the entire training data. In order to prevent the model from overfitting the training data, we split 10% of training data as validation data and set patience as 20, which is the number of epochs with no improvement of validation's loss after which the training process will be stopped. Besides, the optimizer we use is Adam, the same as the literature. We illustrate the training process in Fig. 12. Finally, we set the error as the error of renting/returning “ratio”, instead of the error of the “number” of renting/returning because the same error has greater impact on the stations with less capacity and less impact on the stations with larger capacity. As a result, we define the error as follow:

$$\text{error} = \frac{\text{predicted rental} - \text{true rental}}{\text{capacity}}$$

Then, the RMSE is calculated using the defined error. Because of this, we will divide the input rental data and the true rental data by their corresponding capacity first, and then

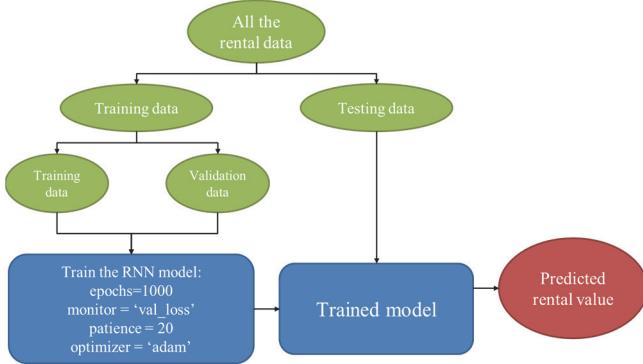


FIGURE 12. The training process.

TABLE IV
THE SETTINGS OF INTERNAL EXPERIMENT

Experiments	Value	Default
Temporal feature setting	P	P+C+G
	C	
	G	
	P+C	
	C+G	
	P+G	
None	P+C+G	
	sin & cos	sin & cos
Spatial feature setting	lat & lng (None)	lat & lng (0~1) + POI
	lat & lng (0~1)	
	lat & lng (z-score)	
	POI	
	lat & lng (0~1) + POI	
	lat & lng (z-score) + POI	
100m	100m	100m
	200m	
	300m	
	400m	
	500m	

the unit of error will be the “prediction error of the station with unit capacity”. For instance, if the error is 8%, then it represents the prediction error is 8 bicycles at a station with capacity equaling to 100.

B. Internal Experiment

We conduct two different kinds of internal experiments, setting different temporal features and spatial features. The settings and the default values are listed in Table IV. We will evaluate the performance of our method under these parameters. Note that in temporal feature setting, P, C, and G represents period, closeness, and general component respectively.

1) TEMPORAL FEATURE SETTING

In this experiment, we test the model performance in different combinations of temporal features first. We test on all the combinations of P, C, and G component. We run ten times for each combination, and each time calculate the total RMSE of all the stations at all the testing times, where the unit of RMSE is %. Finally, we calculate the average RMSE for each combination. The experiment for the number of “renting” and “returning” bicycles is shown in Fig. 13, which plots the average RMSE for each combination. It is easy to find that the more features we include in the model, the better the performance is, and it is quite reasonable for the neural network. Next, we test whether the temporal features need to be split into sin and cos to prove that splitting is better. The experiment follows the same evaluation process, as mentioned in the previous paragraph. Fig. 14 shows the experiment results. The results show that after the temporal features are split into sin and cos, the RMSE decreases apparently.

2) SPATIAL FEATURE SETTING

The second setting to be compared is the spatial feature. First, we test different combinations of spatial features. Originally, our spatial features contain both “latitude and longitude” and “POI”. In this experiment, we want to test how the effect of

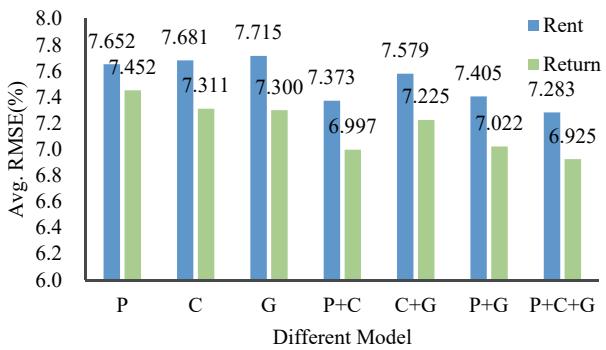


FIGURE 13. The comparison between different temporal feature settings.

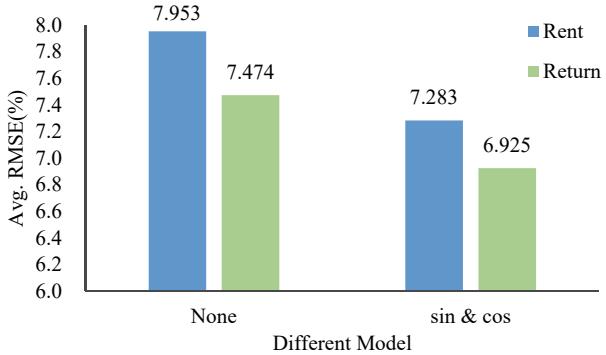


FIGURE 14. The comparison between different sin and cos settings.

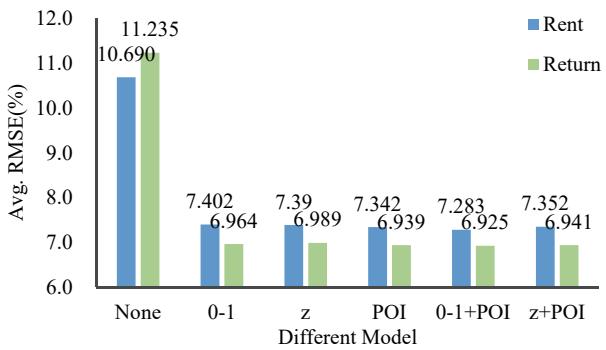


FIGURE 15. The comparison between different normalize and POI combination settings.

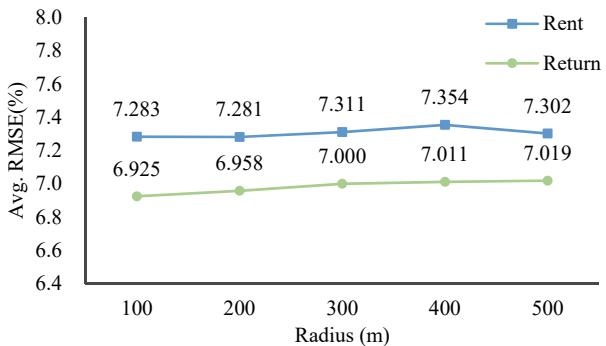


FIGURE 16. The comparison between different buffer settings.

using only one of them is, and the “latitude and longitude” is further applied two different normalizations: rescaling to 0 to 1 and standardization (z-score normalization), or not doing normalization (denoted as “none”). Fig. 15 shows the experiment results. It shows that if we don’t normalize the latitude and longitude, the result will be extremely bad. Besides, the models with POI feature perform better than those do not contain POI features, and the best combination is “latitude and longitude” rescaling to 0 to 1 and “POI”. The other spatial feature experiment is setting different radius for searching POIs near the station. We set 100 meters as default, and we want to see how the results change as the radius getting higher. Fig. 16 reveals that the bigger the buffer is, the higher the trend of RMSE will be, but their differences are not obvious.

C. External Experiment

In the external experiment, we compare our model with Poisson. Poisson and Markov are widely used in the literature; however, Markov is used to predict the inventory level, so we can just compare with Poisson. It needs to assume the rental behavior to be random event, then it will follow Poisson distribution. The meaning of Poisson distribution is that if a random event occurs λ times per unit time on average, then the probability of occurring k times per unit time is Poisson distribution: $P(k)$. The peak of the distribution falls on the average number of occurrences, that is, λ . Therefore, we can calculate the average numbers of the rental from users at the same time in the same day of week to be the predicted values at the same time in the same day of week in the future. Unlike our model, which has random characteristics when training the model, Poisson just need to calculate once. Fig. 17 shows the overall RMSE of our model and Poisson, and our model is better than Poisson, with average RMSE to be 7.283% and 6.925% for renting and 7.433% and 7.164% for returning. Among all the stations, in station “MRT Taipei City Hall Stataion(Exit 3)-2”, our method outperforms Poisson the most. The RMSE of our method and Poisson is 4.998% and 5.906% respectively, so we win $(5.906\% - 4.998\%) = 15.38\%$.

In addition to comparing the overall RMSE, we are still interested in observing the results of different day of week. Fig. 18 and Fig. 19 illustrate the comparison between different day

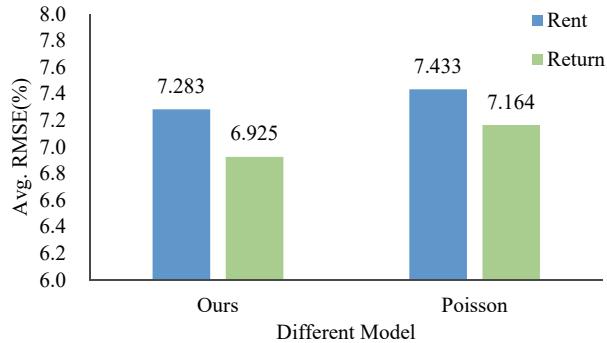


FIGURE 17. The comparison between our model and Poisson.

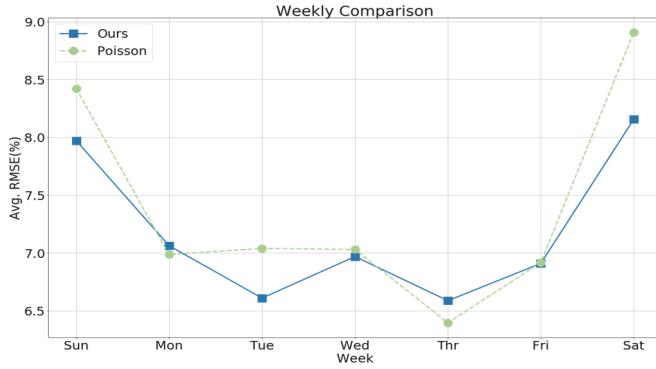


FIGURE 18. The comparisons on different day of week (rent).

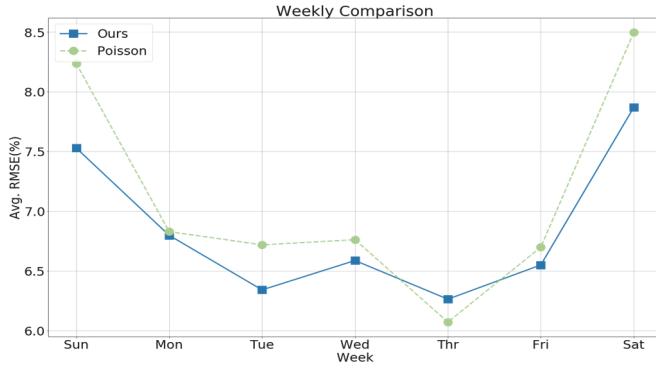


FIGURE 19. The comparisons on different day of week (return).

of week for renting and returning, respectively. It reveals that although our method is not better than Poisson on all the weekdays, which have smaller error, we can outperform Poisson when the prediction error is relatively larger on Saturday and Sunday.

We also compare the performance at different time of day, and the results are shown in Fig. 20 and Fig. 21. Similarly, our model does not completely outperform Poisson at the time when the error is relatively small, but it can beat Poisson when the prediction error is larger in the rush hour during the day. We think that it is due to some activities during the testing period, which did not occur in the training period, and it makes more or fewer people rent the bicycles. Besides, Poisson model just averages the historical data, so it cannot predict special situation in the future well, which causes larger error. In contrast, neural network can adjust appropriately according to the trend of input data.

We still interested in the result of each testing week because our testing data contains the data in the end of the year. We think that the predict error may be higher because humans are going to celebrate the new year. Fig. 22 shows the results. It is apparent that all the predict errors are much higher in the last two weeks. Finally, we compare the results that predict different hours at once in the future. From Fig. 23, we can learn that the longer time we predict, the higher the trend of the error will be. When the prediction time is six hours or less, our model can win against Poisson.

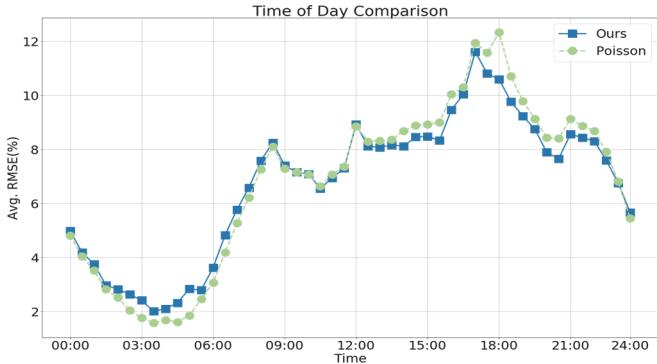


FIGURE 20. The comparisons on different day of week (rent).

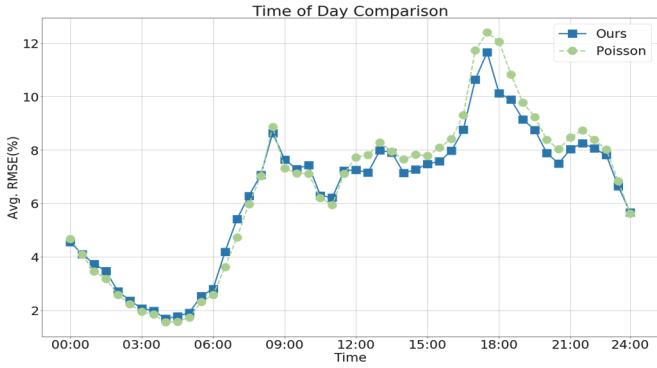


FIGURE 21. The comparisons on different day of week (return).

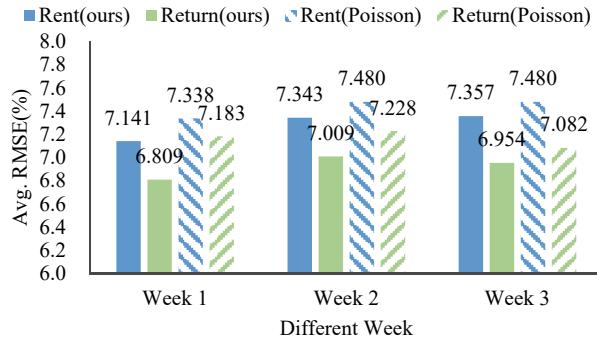


FIGURE 22. The comparisons on different day of week (return).

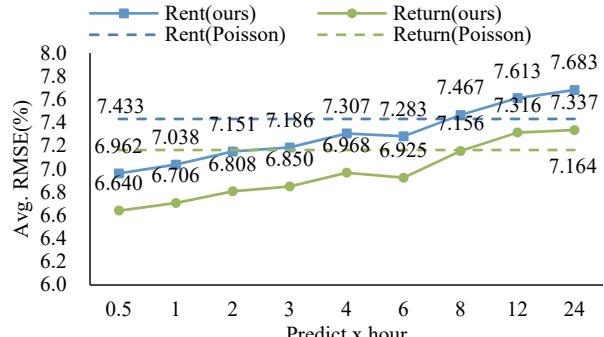


FIGURE 23. The comparisons on different day of week (return).

D. Spatial Analysis Experiment

The last experiment we conduct is the spatial analysis. We calculate the average RMSE for each station and display the stations on the Google Map according to the coordinate of the stations. There are 229 YouBike stations in Taipei City. Here we take renting as example. We calculate the RMSE for each station of our method and Poisson respectively and display the results on the map according to the amount of RMSE, as Fig. 24 shows. The darker the color is, the larger the prediction error is. However, from the figure we cannot clearly observe the difference between the two methods. Thus, we come up

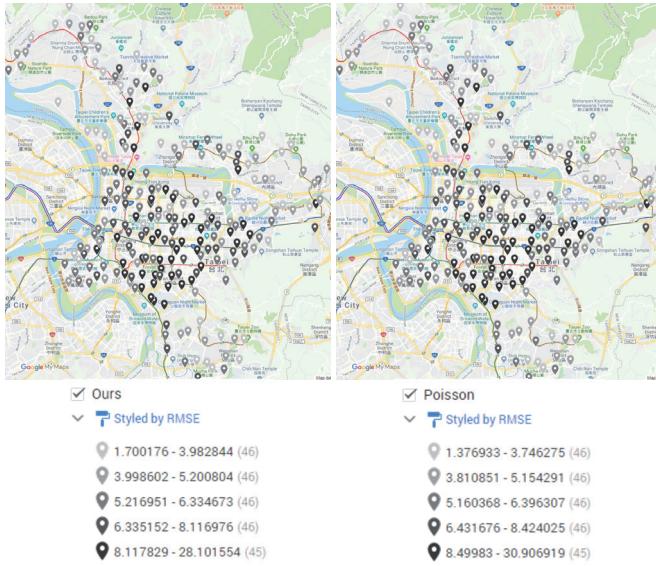


FIGURE 24. The RMSE of our method and Poisson in map.

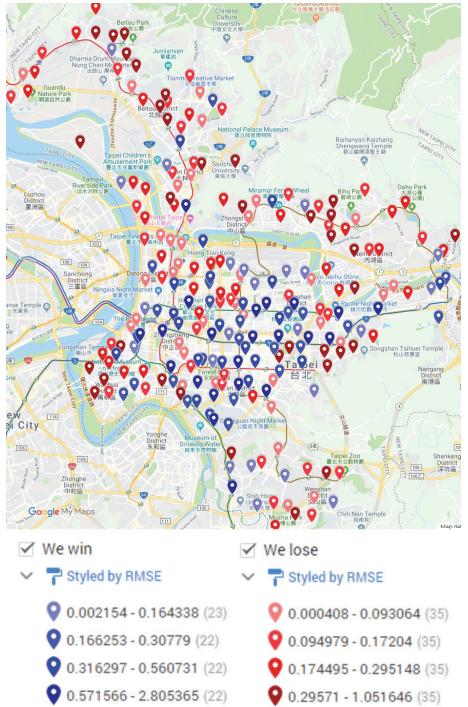


FIGURE 25. The comparison between our method and Poisson in map.

with changing the display value to the difference between our RMSE and Poisson's, as shown in Fig. 25. The stars represent we are better than Poisson, and the darker the color is, the more we win; by contrast, the squares represent we loss Poisson, and the darker the color is, the more we loss. We find an interesting phenomenon: In the downtown area, most of our predictions are better than Poisson, while Poisson wins in the suburb. Comparing to Fig. 24, we can find that the area where we predict more accurately is also the place where the prediction error is larger. We think that it is because Poisson calculates the average values. In the suburban area, where the demand is relatively less and more regular, using merely the average can predict the rental well. As for in the urban area, where the demand is higher and more difficult to predict, our neural network model can predict the rental more accurately.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a modified Recurrent Neural Network based on the concept of Du-Parking in the literature to predict the rental from users in Bicycle-Sharing System. Our study aims to predict the rental from users in the future more accurately and reduce the chances that some of the users' demands cannot be satisfied. Using historical rental data and the POI information, our model can predict the future rental from users for each station. We modify something important in the model to predict the rental from users well, such as considering both temporal and spatial characteristics of the stations and designing a POI feature vector to express the spatial characteristics. Experiments with YouBike real data show that our method can improve the accuracy by up to 15.38%. In the internal experiment, we find the best combinations of temporal features and spatial features using in our model. We discover that the spatial characteristic of POI feature is much more important than that of latitude and longitude. In the external experiment, we compare the prediction accuracy with Poisson, a kind of method often used in the literature to predict the rental in the future. Our approach outperforms Poisson as a whole, and from the point of view in different time, our approach is better than Poisson at the time when the prediction error is higher, such as weekend and rush hours. Finally, in the spatial analysis experiment, our method can predict more accurately in the city center, where the demand is relatively higher and more difficult to predict, which helps reduce more unsatisfied demands in Bicycle-Sharing System.

In the future, we can test the model on the other kinds of Bicycle-Sharing Systems, such as TBike in Tainan or Citi Bike in New York, to confirm that this model can be applied into several systems. Besides, there are still some cases that are not considered in this study. For example, we can further add other features such as weather data into our model, which may improve the prediction accuracy. It is also possible to set different weights for POIs depending on the importance or opening time of different POIs. Moreover, it may be possible to find a better neural network model to make rental prediction

more accurate. For instance, using a hybrid model which predicts the result by different model depending on the location in the country or urban area to get the best result, or clustering the stations before predicting. Furthermore, we can make the evaluation more intuitive rather than RMSE, such as cost reduction or user satisfaction. Finally, this research is devoted to predict the rental from users in Bicycle-Sharing System more accurately, and for future developments after the prediction, we can know which stations will be lack or full of bicycles in the future; accordingly, basing on the predicted values of the future demands, we can arrange the truck to rebalance the bicycles and consider the time delay as well, which is known as the bicycle rebalancing problem with time delay in Bicycle-Sharing System.

REFERENCES

- [1] R. Alvarez-Valdes, J.M. Belenguer, E. Benavent, J.D. Bermudez, F. Muñoz, E. Vercher, and F. Verdejo, "Optimizing the Level of Service Quality of a Bike-Sharing System," *Omega*, vol. 62, pp. 163-175, 2016.
- [2] J. Bachand-Marleau, B.H.Y. Lee, and A.M. El-Geneidy, "Better Understanding of Factors Influencing Likelihood of Using Shared Bicycle Systems and Frequency of Use," *Transportation Research Record*, vol. 2314, no. 1, pp. 66-71, 2012.
- [3] P. Borgnat, P. Abry, P. Flandrin, C. Robardet, J.-B. Rouquier, and E. Fleury, "Shared Bicycles in a City: A Signal Processing and Data Analysis Perspective," *Advances in Complex Systems*, vol. 14, no. 03, pp. 415-438, 2011.
- [4] L. Caggiani, R. Camporeale, M. Ottomanelli, and W.Y. Szeto, "A Modeling Framework for the Dynamic Management of Free-Floating Bike-Sharing Systems," *Transportation Research Part C: Emerging Technologies*, vol. 87, pp. 159-182, 2018.
- [5] L. Caggiani and M. Ottomanelli, "A Modular Soft Computing Based Method for Vehicles Repositioning in Bike-Sharing Systems," *Procedia-Social and Behavioral Sciences*, vol. 54, pp. 675-684, 2012.
- [6] L. Cagliero, T. Cerquitelli, S. Chiusano, P. Garza, G. Ricupero, and E. Baralis, "Characterizing Situations of Dock Overload in Bicycle Sharing Stations," *Applied Sciences*, vol. 8, no. 12, p. 2521, 2018.
- [7] A. Faghih-Imani, N. Eluru, A.M. El-Geneidy, M. Rabbat, and U. Haq, "How Land-Use and Urban Form Impact Bicycle Flows: Evidence from the Bicycle-Sharing System (BIXI) in Montreal," *Journal of Transport Geography*, vol. 41, pp. 306-314, 2014.
- [8] I. Frade and A. Ribeiro, "Bicycle Sharing Systems Demand," *Procedia-Social and Behavioral Sciences*, vol. 111, pp. 518-527, 2014.
- [9] I. Frade and A. Ribeiro, "Bike-Sharing Stations: A Maximal Covering Location Approach," *Transportation Research Part A: Policy and Practice*, vol. 82, pp. 216-227, 2015.
- [10] X. Gao and G.M. Lee, "Moment-Based Rental Prediction for Bicycle-Sharing Transportation Systems Using a Hybrid Genetic Algorithm and Machine Learning," *Computers & Industrial Engineering*, vol. 128, pp. 60-69, 2019.
- [11] J.C. García-Palomares, J. Gutiérrez, and M. Latorre, "Optimizing the Location of Stations in Bike-Sharing Programs: A GIS Approach," *Applied Geography*, vol. 35, no. 1-2, pp. 235-246, 2012.
- [12] N. Gast, G. Massonet, D. Reijnsbergen, and M. Tribastone, "Probabilistic Forecasts of Bike-Sharing Systems for Journey Planning," *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, pp. 703-712, 2015.
- [13] S. Ghosh, P. Varakantham, Y. Adulyasak, and P. Jaillet, "Dynamic Repositioning to Reduce Lost Demand in Bike Sharing Systems," *Journal of Artificial Intelligence Research*, vol. 58, pp. 387-430, 2017.
- [14] S. Jäppinen, T. Toivonen, and M. Salonen, "Modelling the Potential Effect of Shared Bicycles on Public Transport Travel Times in Greater Helsinki: An Open Data Approach," *Applied Geography*, vol. 43, pp. 13-24, 2013.
- [15] R. Jurdak, "The Impact of Cost and Network Topology on Urban Mobility: A Study of Public Bicycle Usage in 2 U.S. Cities," *PloS ONE*, vol. 8, no. 11, e79396, 2013.
- [16] I.J. Leontaritis and S.A. Billings, "Input-Output Parametric Models for Non-Linear Systems Part I: Deterministic Non-Linear Systems," *International Journal of Control*, vol. 41, no. 2, pp. 303-328, 1985.
- [17] J.-R. Lin and T.-H. Yang, "Strategic Design of Public Bicycle Sharing Systems with Service Level Constraints," *Transportation Research Part E: Logistics and Transportation Review*, vol. 47, no. 2, pp. 284-294, 2011.
- [18] J.-R. Lin, T.-H. Yang, and Y.-C. Chang, "A Hub Location Inventory Model for Bicycle Sharing System Design: Formulation and Solution," *Computers & Industrial Engineering*, vol. 65, no. 1, pp. 77-86, 2013.
- [19] O. O'Brien, J. Cheshire, and M. Batty, "Mining Bicycle Sharing Data for Generating Insights into Sustainable Transport Systems," *Journal of Transport Geography*, vol. 34, pp. 262-273, 2014.
- [20] T. Raviv and O. Kolka, "Optimal Inventory Management of a Bike-Sharing Station," *IIE Transactions*, vol. 45, no. 10, pp. 1077-1093, 2013.
- [21] R. Regue and W. Recker, "Proactive Vehicle Routing with Inferred Demand to Solve the Bikesharing Rebalancing Problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 72, pp. 192-209, 2014.
- [22] Y. Rong, Z. Xu, R. Yan, and X. Ma, "Du-Parking: Spatio-Temporal Big Data Tells You Realtime Parking Availability," *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 646-654, 2018.
- [23] C. Rudloff and B. Lackner, "Modeling Demand for Bikesharing Systems: Neighboring Stations as Source for Demand and Reason for Structural Breaks," *Transportation Research Record*, vol. 2430, no. 1, pp. 1-11, 2014.
- [24] J. Schuijbroek, R.C. Hampshire, and W.-J. van Hoeve, "Inventory Rebalancing and Vehicle Routing in Bike Sharing Systems," *European Journal of Operational Research*, vol. 257, no. 3, pp. 992-1004, 2017.
- [25] D. Tomaras, V. Kalogeraki, T. Liebig, and D. Gunopulos, "Crowd-Based Ecofriendly Trip Planning," *2018 19th IEEE International Conference on Mobile Data Management (MDM)*, pp. 24-33, 2018.
- [26] J. Woodcock, M. Tainio, J. Cheshire, O. O'Brien, and A. Goodman, "Health Effects of the London Bicycle Sharing System: Health Impact Modelling Study," *BMJ*, vol. 348, g425, 2014.
- [27] D. Zhang, C. Yu, J. Desai, H.Y.K. Lau, and S. Srivathsan, "A Time-Space Network Flow Approach to Dynamic Repositioning in Bicycle Sharing Systems," *Transportation Research Part B: Methodological*, vol. 103, pp. 188-207, 2017.



Eric Hsueh-Chan Lu is currently an associate professor in the Department of Geomatics, National Cheng Kung University (NCKU), Taiwan, R.O.C. Before this, he was an assistant professor in the Department of Computer Science and Information Engineering, National Taitung University, Taiwan, R.O.C. during August 2012 and July 2014. From October 2011 to July 2012, he was a postdoctoral fellow in Academia Sinica and NCKU, respectively. Dr. Lu received his Ph.D. degree from National Cheng Kung University, Taiwan, R.O.C., in 2010, majored in computer science and information engineering. Dr. Lu has a wide variety of research interests covering data mining, database systems, location-based services, geographic information systems, mobile computing, intelligent transportation systems, social networking and cloud computing. He has published more than 40 research papers in referred journals and international conferences. He received the best student paper award of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks (LBSN). He is a member of IEEE and has also served as reviewers for a number of international conferences related to data mining and mobile application.



Zhan-Qing Lin received the B.S. and Master degree in the Department of Geomatics at National Cheng Kung University (NCKU), Taiwan, R.O.C. in 2017 and 2019 respectively. His research interests include data mining, deep learning, intelligent transportation systems and geographic information systems.