

# Dynamic Planning of Bicycle Stations in Dockless Public Bicycle-sharing System Using Gated Graph Neural Network

JIANGUO CHEN and KENLI LI, Hunan University, China  
KEQIN LI, Hunan University, China and State University of New York, USA  
PHILIP S. YU, University of Illinois at Chicago, USA  
ZENG ZENG, Agency for Science Technology and Research, Singapore

Benefiting from convenient cycling and flexible parking locations, the Dockless Public Bicycle-sharing (DL-PBS) network becomes increasingly popular in many countries. However, redundant and low-utility stations waste public urban space and maintenance costs of DL-PBS vendors. In this article, we propose a Bicycle Station Dynamic Planning (BSDP) system to dynamically provide the optimal bicycle station layout for the DL-PBS network. The BSDP system contains four modules: bicycle drop-off location clustering, bicycle-station graph modeling, bicycle-station location prediction, and bicycle-station layout recommendation. In the bicycle drop-off location clustering module, candidate bicycle stations are clustered from each spatio-temporal subset of the large-scale cycling trajectory records. In the bicycle-station graph modeling module, a weighted digraph model is built based on the clustering results and inferior stations with low station revenue and utility are filtered. Then, graph models across time periods are combined to create a graph sequence model. In the bicycle-station location prediction module, the GGNN model is used to train the graph sequence data and dynamically predict bicycle stations in the next period. In the bicycle-station layout recommendation module, the predicted bicycle stations are fine-tuned according to the government urban management plan, which ensures that the recommended station layout is conducive to city management, vendor revenue, and user convenience. Experiments on actual DL-PBS networks verify the effectiveness, accuracy, and feasibility of the proposed BSDP system.

CCS Concepts: • **Information systems** → **Clustering**; **Data analytics**; • **Computing methodologies** → **Neural networks**;

Additional Key Words and Phrases: Bicycle station layout, dockless PBS network, gated graph neural network, public bicycle-sharing system

This work is partially funded by the National Key R&D Program of China (Grant No. 2020YFB2104000), the National Outstanding Youth Science Program of National Natural Science Foundation of China (Grant No. 61625202), the Program of National Natural Science Foundation of China (Grant No. 61751204), the International (Regional) Cooperation and Exchange Program of National Natural Science Foundation of China (Grant No. 61860206011), the Natural Science Foundation of Hunan Province (Grant No. 2020JJ5084), and the International Postdoctoral Exchange Fellowship Program (Grant No. 20180024). This work is also supported in part by NSF under grants III-1763325, III-1909323, and SaTC-1930941.

Authors' addresses: J. Chen and K. Li (corresponding author), Hunan University, College of Computer Science and Electronic Engineering, Changsha, Hunan, 410082, China; emails: {jianguochen, likl}@hnu.edu.cn; K. Li, Hunan University, College of Computer Science and Electronic Engineering, Changsha, Hunan, 410082, China, State University of New York, Department of Computer Science, New Paltz, NY, 12561; email: lik@newpaltz.edu; P. S. Yu, University of Illinois at Chicago, Department of Computer Science, Chicago, IL, 60607; email: psyu@uic.edu; Z. Zeng, Agency for Science Technology and Research, Institute for Infocomm Research, 138632, Singapore; email: zengz@i2r.a-star.edu.sg.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2021 Association for Computing Machinery.

2157-6904/2021/03-ART25 \$15.00

<https://doi.org/10.1145/3446342>

**ACM Reference format:**

Jianguo Chen, Kenli Li, Keqin Li, Philip S. Yu, and Zeng Zeng. 2021. Dynamic Planning of Bicycle Stations in Dockless Public Bicycle-sharing System Using Gated Graph Neural Network. *ACM Trans. Intell. Syst. Technol.* 12, 2, Article 25 (March 2021), 22 pages. <https://doi.org/10.1145/3446342>

---

**1 INTRODUCTION**

With the advantages of zero carbon emissions and convenient cycling, public bicycles have significant benefits in urban short trips and are widely used as public transportation to solve the first/last mile problem [5, 9, 29]. In many cities in the world, there are numerous Station/Dock-based Public Bicycle-sharing (SD-PBS) systems that provide citizens with public bicycles [10, 22]. In the SD-PBS system, each bicycle station has multiple fixed docks, which greatly limits the movement of the station and the increase of bicycles, and further prevents users from renting and returning bicycles [24]. In current years, Dockless PBS (DL-PBS) systems become increasingly prevalent in many countries [1, 26, 32]. DL-PBS providers deploy public bicycles at flexible parking points (drop-off locations) instead of fixed stations, and users can return bicycles at anywhere near their destination. Benefitting from the dockless parking and low station moving cost, we can deploy stations and dispatch bicycles dynamically according to the actual demands in different periods.

Various practical problems arise during the deployment of DL-PBS networks, such as unreasonable bicycle station layout, inefficient and imprecise bicycle deployment, and difficulty in bicycle maintenance [24]. On the one hand, due to low deployment costs and vicious competition from peers, a large number of redundant bicycles are deployed in locations that are not frequently used. Meanwhile, due to the fierce competition between different providers and the low deployment cost, a large number of redundant bicycle stations are arbitrarily deployed [4]. Massive cluttered bicycle stations from different providers raise the problems of urban road management and traffic safety while causing a waste of bicycling resources at low-usage stations. On the other hand, because users may return their bicycles to any place, bicycle drop-off locations will spread across all corners of the city. Users face difficulties in finding nearby drop-off locations to rent or return bicycles. Therefore, bicycle station layout is an important factor affecting the Quality of service (QoS) of PBS, and also affects the bicycle deployment, maintenance, and dispatching.

Numerous efforts have been devoted to the research of PBS networks, such as station layout modeling, bicycle dispatching, and bicycle demand prediction [4, 19]. In station layout modeling, most existing works focused on the traditional SD-PBS networks, mainly relying on empirical experience and surveys in terms of population distribution and environmental factors [10, 16]. The traditional approaches face limitation in the novel DL-PBS networks in two aspects: (1) the large-scale actual bicycle loan datasets are not fully utilized in station modeling, while the real-time bicycle demand is not considered as well, and (2) SD-PBS networks hold the characteristics that the fixed stations locations and fixed docks in each station, which increase the cost of station migration and have difficulty in matching the flexible bicycle demands. Existing bicycle stations of the SD-PBS and DL-PBS networks are usually deployed based on the flow rate and density of visitors, lacking in a scientific and reasonable basis. Moreover, most of existing methods design the static bicycle station layout at the initial phase. Owing to the high construction cost, once these stations are built, the location of these stations will not change.

In this article, we focus on the bicycle station layout of the DL-PBS network and propose a Bicycle Station Dynamic Planning (BSDP) system. The workflow of the BSDP system is illustrated in Figure 1. The system can dynamically predict the location of bicycle stations and the number of bicycles needed at each station, providing accurate planning for bicycle station deployment and bicycle dispatching. Our contributions in this article can be summarized as follows:

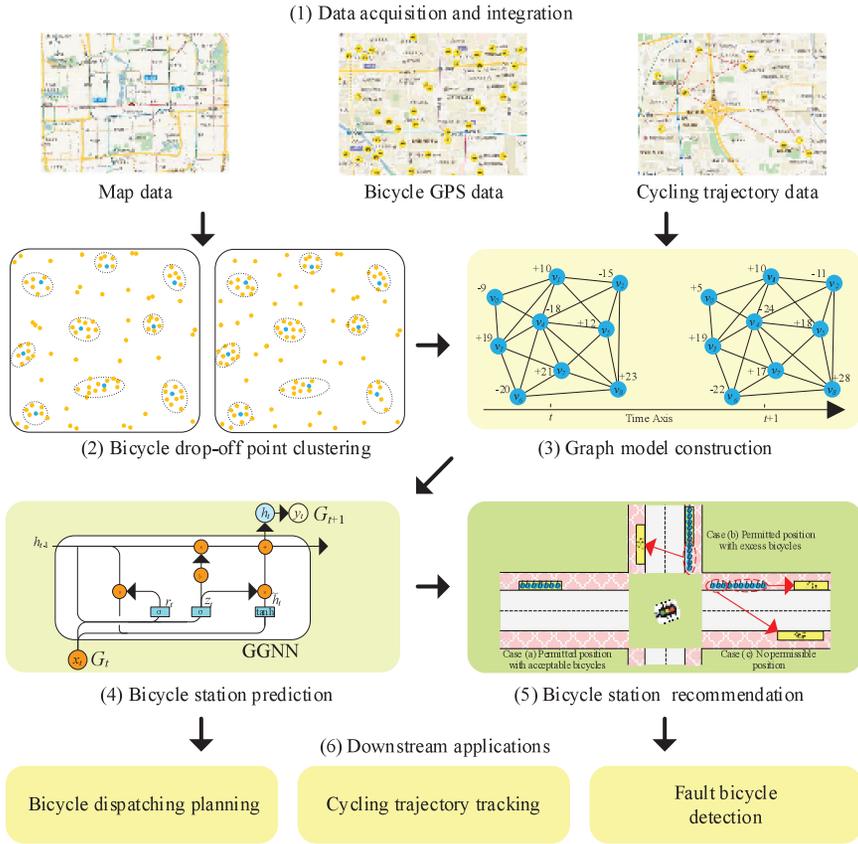


Fig. 1. Workflow of the proposed Bicycle Station Dynamic Planning (BSDP) system. (1) Collect and integrate large-scale bicycle GPS datasets, cycling trajectory datasets, and map data from the actual DL-PBS network. (2) Use a bicycle drop-off location clustering method to detect candidate bicycle stations. (3) Create a weighted digraph model based on the candidate bicycle stations after filtering inferior stations, and build a graph sequence model by linking graph models across time periods. (4) Use the GGNN model to train the graph sequence data and dynamically predict the bicycle stations in the next period. (5) Fine-tune the predicted bicycle stations according to the government’s urban management plan to ensure their legitimacy and maximize revenue. (6) The bicycle station layout can be used for downstream applications, such as bicycle dispatching planning, cycling trajectory tracking, and fault bicycle detection.

- We collect large-scale cycling trajectory records from the DL-PBS network and propose a bicycle drop-off location clustering method, in which dense bicycle drop-off locations are clustered as candidate bicycle stations.
- Based on the clustering results, we construct a bicycle station graph model for each spatio-temporal subset and remove the inferior stations with low station revenue and utility. In addition, graph models across time periods are linked as a graph sequence model.
- We introduce the Gated Graph Neural Network (GGNN) to create a bicycle station prediction method, in which bicycle demand and bicycle stations in the next time period are dynamically predicted based on the historical graph sequence model.
- We propose a bicycle station layout recommendation method, where the location of predicted stations and the number of public bicycles needed at each station are fine-tuned according to the government management plan.

The remainder of the article is organized as follows: Section 2 reviews the related work. Section 3 introduces the bicycle drop-off location clustering and graph modeling of the DL-PBS network. Section 4 describes the bicycle station dynamic planning system. Section 5 provides experimental evaluations. Section 6 concludes this article with future work and directions.

## 2 RELATED WORK

Various efforts have been made in researching big data and data mining techniques to build smart cities [29, 32, 33]. Numerous studies of PBS networks and smart cities attract both fields of industry and academia [3, 13, 19]. Focusing on the analysis and prediction of public bicycle demands, Zhang et al. used a circular distribution method to obtain the daily peak of the public bicycle trips. They used a time series model to predict the bicycle demands during rush hours [3]. Etienne et al. analyzed the use of PBS networks by using model-based count series clustering [9]. Cazabet et al. studied cycling rules from the perspective of signal processing and data analysis [2], in which a cycling periodic model was built by analyzing the characteristics of time, space, and riders. However, most existing models are built on external factors such as population density and travel probability, rather than the laws of cycling trajectory records themselves.

There are numerous approaches focused on location prediction [15] in temporal-spatial environments. In Reference [15], Jia et al. proposed a temporal-spatial Bayesian model to predict user's location based on his influential friends' locations. In Reference [30], Ying et al. introduced a Geographic-Temporal-Semantic (GTS) model and proposed a GTS-based location prediction method. They collected the trajectories of users and calculated the similarity of the movement and trajectories between users. Focusing on the layout of public bicycle stations, various schemes were carried out in References [16, 24]. Ma et al. proposed a hierarchical public bicycle dispatching strategy for dynamic demand [24]. In Reference [7], Deng et al. discussed the layout optimization of public bicycle stations based on the AHP method. Jiang et al. analyzed the GPS trajectory of urban bicycles and detected  $K$  primary corridors on the road network [16]. The scale of bicycle stations includes the number of bicycles, the grade of the station, and the scope of bicycle-sharing services. However, most of the existing studies focus on the traditional SD-PBS network, and they design the static bicycle station layout at the initial phase. Owing to the high construction cost, once these stations are built, the location of these stations will not change. Different from the existing research, we focus on the dynamic layout of the bicycle stations for the DL-PBS network and update the bicycle stations between different time periods according to the actual bicycle usage demands.

Graph Neural Network (GNN) is an effective deep learning model used in graph or network applications [5, 18, 25, 28]. In Reference [17], Khodayar et al. proposed a GNN model and applied it to wind speed prediction. Levie et al. introduced a spectral-domain convolutional architecture of DL on graph models [20]. Lin et al. used the GNN method to predict the hour-level demands of bicycle stations in the SD-PBS network. They used the Long Short-Term Memory (LSTM) neural network to capture the temporal dependency in bicycle-sharing demand sequences [23]. In Reference [11], Gast et al. introduced a generalized regression neural network to predict the public bicycle demands of SD-PBS. Li et al. modified the GNN model and proposed a Gated Graph Neural Network (GGNN) to achieve a flexible and broadly output sequence [21]. In this work, we use the GGNN model to train the bicycle station graph sequence of the DL-PBS network and predict the location of bicycle stations and their bicycle-sharing demands.

## 3 DL-PBS NETWORK CLUSTERING AND GRAPH MODELING

In this section, we will describe the DL-PBS network and highlight its characteristics with practical issues that arise during the deployment. Then, based on the cycling trajectory records, we establish

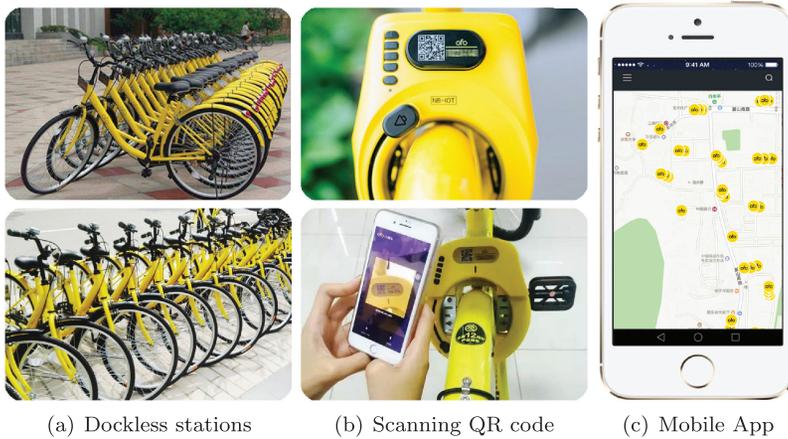


Fig. 2. Principal components of a DL-PBS network, including public bicycles, dockless bicycle stations, a GPS module, a QR-code-based locking module, and a mobile application.

a weighted digraph model for the DL-PBS network and calculate the update of bicycle stations to create a graph sequence model.

### 3.1 Dockless Public Bicycle-sharing (DL-BPS) Network

As a new generation of PBS systems, the DL-PBS network provides personalized and convenient services during bicycle rental and return [26]. The principal components of a DL-PBS network include public bicycles, dockless stations, Global Positioning System (GPS), a Quick Response (QR) code-based locking module, and a mobile application, as shown in Figure 2.

- Public bicycles. Public bicycles have a distinctive look to be quickly identified by users. Each bicycle is equipped with a GPS to record its position in real time and a QR code-based locking module.
- Dockless bicycle stations (also called drop-off locations or parking points). The DL-PBS providers deploy bicycles at any permitted public parking area, such as the curbside, the entrances of parks, communities, and shopping malls. Each dense bicycle parking point is termed as a temporary dockless bicycle station, as shown in Figure 2(a).
- QR code. Each bicycle has a unique QR code that can be scanned via a mobile App to unlock the bicycle and pay the rent, as shown in Figure 2(b).
- Mobile application (APP). Mobile APP is an important component of the DL-PBS network. It provides functions including bicycle GPS locating, QR code scanning and unlocking, payment, and cycling trajectory tracking, as shown in Figure 2(c).

### 3.2 Bicycle Drop-off Location Clustering

We collect large-scale historical bicycle GPS datasets and cycling trajectory records from DL-PBS networks deployed in different cities and administrative regions. Bicycle GPS records are routinely collected from all stationary bicycles. Each bicycle GPS record contains the bike ID, longitude, latitude, and the timestamp of collection. Cycling trajectory records are saved from the users' cycling behaviors. Each cycling trajectory record contains user ID, bike ID, the timestamp of pick-up (rent) and drop-off (return), longitude and latitude of rent and return positions, cycling distance, and cost. Considering that the layout of bicycle stations is usually suitable for a certain city or administrative region, we need to build a bicycle station graph model for each city or administrative region at



Fig. 3. Examples of cycling trajectory and GPS records. (a) is the cycling trajectory records of bicycles, where each yellow icon is a bicycle GPS location, and the red directed line is the cycling route. (b) is the GPS records of stationary bicycles.

Table 1. Examples of Cycling Trajectory Records

User	Bicycle	Departure Info.			Arrival Info.		
		Time stamp	Latitude	Longitude	Time stamp	Latitude	Longitude
01	e1xx4	2018/10/25 10:20:22	39.914548	116.440848	2018/10/25 10:48:13	39.900323	116.484110
02	e1xx9	2018/10/25 09:11:19	39.914326	116.482170	2018/10/25 09:43:27	39.899604	116.425325
03	elxx3	2018/10/25 19:15:10	39.899604	116.425325	2018/10/25 19:44:23	39.890705	116.483715
...	...	...	...	...	...	...	...

each time period. Therefore, we divide the cycling trajectory dataset into multiple spatial subsets based on administrative divisions and further divide each spatial subset into multiple temporal subsets by time periods. Examples of cycling trajectory and GPS records are given in Figure 3 and Table 1.

The bicycle drop-off locations collected from bicycle GPS datasets have the characteristics of varying density distribution (VDD), equilibrium distribution (ED), and multiple domain-density maximums (MDDM). Specifically, bicycles are densely distributed in some areas (i.e., urban areas), while sparsely distributed in other areas (i.e., suburban areas), which is in line with the VDD characteristic. In addition, in the initial deploy state, the bicycles at each drop-off location are parked neatly at the same interval, so each bicycle at the drop-off location has an ED. Moreover, it is easy to know that the bicycle position may have multiple domain density maximums during use and return. The Domain Adaptive Density Clustering Algorithm (DADC) algorithm proposed in our previous work [6] can obtain more reasonable clustering results on data with VDD, ED, and MDDM characteristics. Therefore, we introduce the DADC algorithm for bicycle drop-off location clustering.

Given a set of cycling trajectory records in a temporal subset, we extract all GPS information of pick-up and drop-off from each record. Assume that there are  $M$  records in the dataset  $X$ , hence  $2M$  positions are extracted. Considering that the same bicycle may be used many times during the current period, the position will be extracted repeatedly. In the rest of this article, we perform bicycle drop-off location clustering and graph modeling for each temporal subset  $X$ . We filter the position information of the same bicycle ID with the same latitude and longitude. We get the dataset of bicycle positions in the current period  $X = \{x_1, \dots, x_N\}$ , and the number of positions  $N$

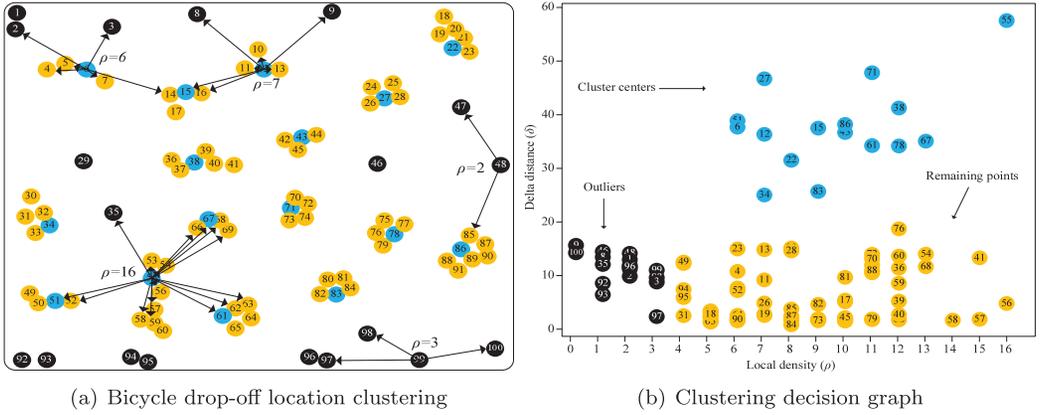


Fig. 4. Example of DADC-based bicycle drop-off location clustering. (a) For each data point, we calculate the local density and the delta distance. (b) We draw a clustering decision graph based on local density and delta distance to detect cluster centers, outliers, and remaining points.

satisfies  $M \leq N \leq 2M$ . For each data point  $x_i$  in  $X$ , we define the local density  $\rho_i$  of  $x_i$  as:

$$\rho_i = \sum_{x_j \in N(x_i)} \chi(d_{ij} - d_c), \quad (1)$$

where  $N(x_i)$  is the neighbors of  $x_i$ ,  $d_c$  is a cutoff distance, and  $\chi(d_{ij} - d_c) = 1$ , if  $d_{ij} < d_c$ ; otherwise,  $\chi(d_{ij} - d_c) = 0$ . Therefore,  $\rho_i$  is equal to the number of points closer than  $d_c$  to  $x_i$ . In addition, we calculate the delta distance  $\delta_i$  of  $x_i$  by computing the shortest distance between  $x_i$  and any other data points with a higher density:

$$\delta_i = \min_{j: \rho_j > \rho_i} d_{ij}. \quad (2)$$

For the highest density point,  $\delta_i = \max_{x_j \in X} (d_{ij})$ .

After calculating the local density  $\rho$  and the delta distance  $\delta$ , we can draw a clustering decision graph based on  $\rho$  and  $\delta$ , where  $\rho$  is the x-axis and  $\delta$  is the y-axis. Then, we observe the distribution of these points from the graph. Data points with a high  $\rho$  ( $\rho_i > \theta_\rho$ ) and a high  $\delta$  ( $\delta_i > \theta_\delta$ ) are considered as cluster centers, while points with a low  $\rho$  and a high  $\delta$  are considered as outliers. In practical applications, the values of  $\theta_\rho$  and  $\theta_\delta$  are manually set based on experience. In this work, the effective thresholds are set as  $\theta_\rho = \frac{1}{3}\rho_{max}$  and  $\theta_\delta = \frac{1}{3}\delta_{max}$ . After finding the cluster centers, each remaining point is assigned to the same cluster as its nearest neighbor with a higher density. An example of the DADC-based bicycle drop-off location clustering process is illustrated in Figure 4.

The detailed steps of DADC-based bicycle drop-off location clustering are presented in Algorithm 3.1. The process of the bicycle drop-off location clustering includes sub-processes of cluster center detection and remaining point assignment. Assuming that the number of data points in  $X$  is equal to  $N$ , the computational complexity of Algorithm 3.1 is  $O(N)$ .

### 3.3 Bicycle Station Graph Model

Based on the bicycle drop-off location clustering results, we obtain a series of candidate bicycle stations. We build a weighted digraph model for these stations, where the stations are considered as vertices, and the cycling records between them are collected as the corresponding directed edges. In addition, we detect inferior stations with low station revenue and utility and remove them from the graph to obtain a high-quality bicycle station graph. An example of the bicycle station graph modeling process is shown in Figure 5.



where  $R$  is the radius of the earth, usually set to 6,371.0 km, and the Haversine function  $H(\theta)$  is defined as:

$$H(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1}{2}(1 - \cos(\theta)). \quad (4)$$

For each directed edge  $e_{ij}$ , we count the number of cycling records starting from station  $v_i$  and arriving at  $v_j$ , and treat it as the weight  $w_{ij}$  of the edge. Note that  $w_{ij} \neq w_{ji}$ . In this way, we obtain the set of weights  $W$  of all directed edges  $E$  and create a weighted digraph model  $G = (V, E, D, W)$ .

(2) Remove inferior bicycle stations. As mentioned above, due to low deployment costs and vicious competition from peers, a large number of redundant bicycles are deployed in locations that are not frequently used. We need to detect and remove these stations from the bicycle station graph to maximize the benefits and utility of the DL-PBS network.

*Definition 1 (Station Revenue).* The revenue of a bicycle station is the sum of the cycling costs of all bicycles rented (departing) from the station. The revenue of each station  $v_i$  is calculated as:

$$P_i = \sum_{w_{ij} \in W} (w_{ij} \times d_{ij} \times \alpha), \quad (5)$$

where  $\alpha$  is the cycling cost per unit distance, and  $(d_{ij} \times \alpha)$  is the cycling cost of a bicycle from  $v_i$  to  $v_j$ .

*Definition 2 (Station Utility).* The throughput of a bicycle station refers to the number of bicycles leaving or arriving at this station. The utility of a station is the ratio of the station's throughput to the entire graph's throughput. The throughput of a station  $v_i$  is defined as:

$$TP_i = \sum_{w_{ij}, w_{j'i} \in W} (w_{ij} + w_{j'i}), \quad (6)$$

where  $w_{ij}$  is the weight of the directed edges leaving  $v_i$  to any station  $v_j$  and  $w_{j'i}$  is the weight of the directed edges arriving at  $v_i$  from any station  $v_j$ . Let  $TP_G = \sum_{w_{ij} \in W} w_{ij}$  be the throughput of the entire graph, we can calculate the utility of station  $v_i$  as:

$$U_i = \frac{TP_i}{2TP_G} = \sum_{w_{ij}, w_{j'i} \in W} (w_{ij} + w_{j'i}) \Bigg/ 2 \sum_{w_{ij} \in W} w_{ij}. \quad (7)$$

The relationship of station utilization and station degrees: Utilization is the number of borrowed and returned bicycles at this station, and station degrees refers to the station of the number of access.

*Definition 3 (Inferior Bicycle Stations).* A bicycle station  $v_i$  is regarded as an inferior station if its revenue  $P_i$  is below the given threshold  $\theta_P$  and its utility  $U_i$  is below the given threshold  $\theta_U$ .

Station revenue is positively related to station utility. Namely, an increase in station utility will bring an increase in station revenue. In the actual operation of the DL-PBS network, due to different DL-PBS network layouts and densities in different cities, distinct values are set to these thresholds. Based on the station revenue and utility, we can detect inferior bicycle stations and remove them and their associated edges from the bicycle station graph.

Algorithm 3.2 describes the detailed steps of bicycle station graph modeling. In Algorithm 3.2, the process of the bicycle station graph modeling includes sub-processes of graph construction and inferior bicycle stations deletion. Assuming that the number of bicycle station clusters is equal to  $|C|$ , that is, the number of vertices of the bicycle station graph  $G$ , the computational complexity of Algorithm 3.2 is  $O(|C|)$ .

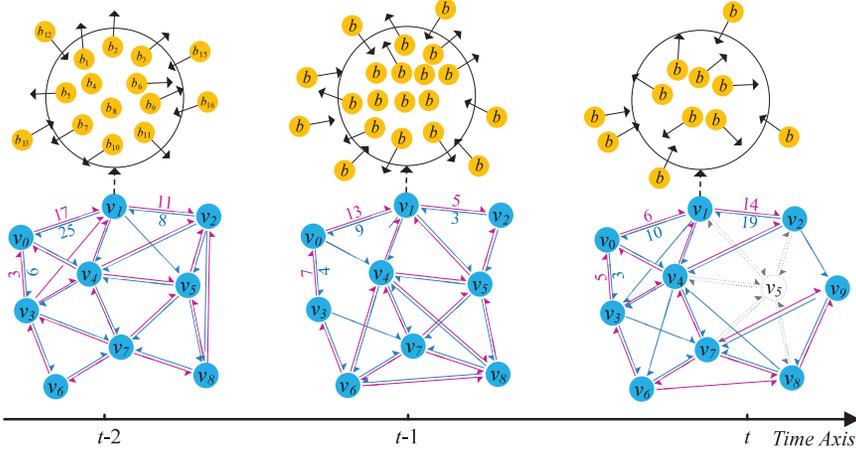


Fig. 6. Bicycle station updating and graph sequence modeling.

---

### ALGORITHM 3.2: Bicycle station graph modeling

---

**Input:**

- $C$ : The clustering results of the bicycle stations;
- $X$ : A temporal subset of cycling trajectory records;
- $\theta_P$ : The threshold of station revenue;
- $\theta_U$ : The threshold of station utility;

**Output:**

- $G$ : the graph model of the DL-PBS network.
  - 1: **for** each cluster  $c_i$  in clusters  $C$  **do**
  - 2:   identify the cluster center and create a vertex  $v_i \leftarrow \text{center}(c_i)$ ;
  - 3:   calculate attributes of  $\varphi_i$ ,  $\psi_i$ , and  $n_i$  for  $v_i$ ;
  - 4:   calculate directed edges  $E$  and weights  $W$  from  $C$  and  $X$ ;
  - 5:   calculate the distances  $D$  between vertices using Equation (3);
  - 6:   build a graph model  $G \leftarrow (V, E, D, W)$ ;
  - 7: **for** each vertex  $v_i$  in  $V$  **do**
  - 8:   calculate station revenue  $P_i$  using Equation (5);
  - 9:   calculate station utility  $U_i$  using Equation (7);
  - 10:   **if**  $P_i < \theta_P$  and  $U_i < \theta_U$  **then**
  - 11:     detect  $v_i$  as an inferior station and remove  $v_i$  from  $V$ ;
  - 12:     remove edges  $e_{ij}$  and  $e_{ji}$  from  $E$ ;
  - 13: **return**  $G$ .
- 

### 3.4 Graph Sequence Model of DL-PBS Network

In the previous section, we established a bicycle station graph for each temporal subset of cycling trajectory records in a city or administrative region. In this way, we can build a series of graphs between different time periods in the same area. In this section, we consider updates to bicycle stations across time periods, add a time dimension to these graphs, and build a graph sequence model. An example of the graph sequence modeling process is illustrated in Figure 6.

Given a time series  $T = \{1, \dots, t\}$ , we divide a region's historical cycling trajectory records into a series of temporal subsets  $X = X_1, \dots, X_t$ , and then build a graph model  $G_t$  for each subset  $X_t$ . In this way, we can build a series of graphs and create a graph sequence model  $GS = \{G_1, \dots, G_t\}$ .

## 4 BICYCLE STATION DYNAMIC PLANNING SYSTEM

Based on bicycle drop-off location clustering, we established a series of bicycle station graph models and further created a graph sequence model. In this section, we propose a Bicycle Station Dynamic Planning (BSDP) system to predict the location of stations and their bicycle-sharing demands in the next period. The GGNN model is used to train the bicycle station graph sequence

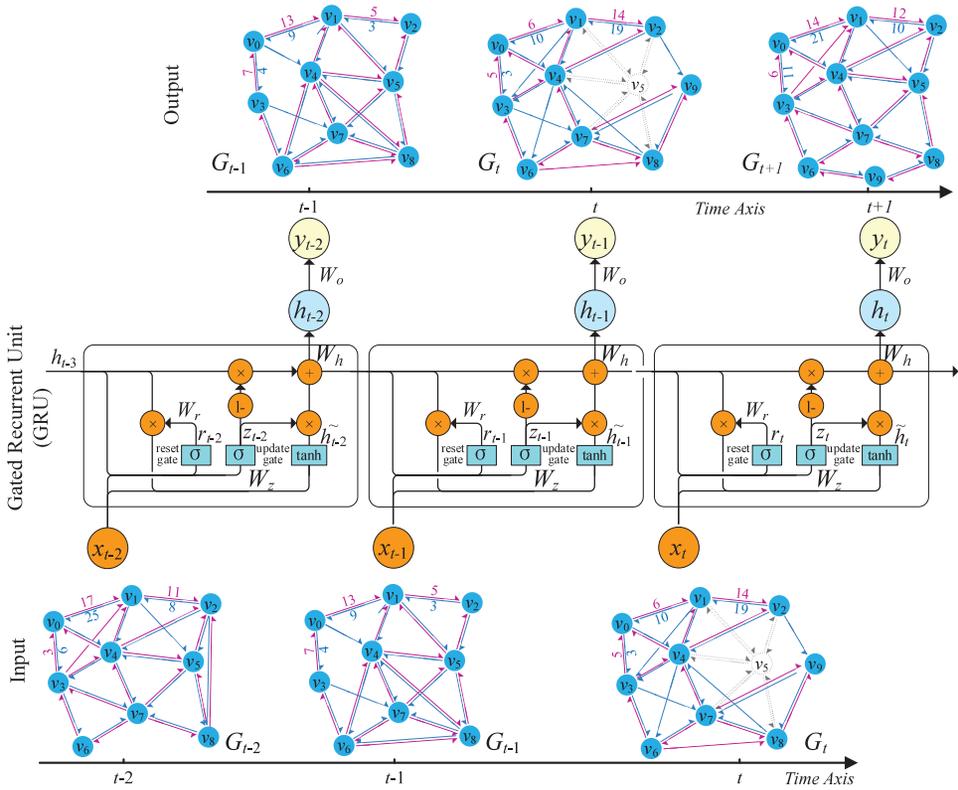


Fig. 7. Structure of the GGNN-based bicycle-station location prediction model.

model and predict a bicycle station graph in the next period. In addition, we fine-tune the location of bicycle stations by matching the predictions with the government's urban management plan. Finally, we provide a bicycle station layout recommendation for the DL-BPS system.

#### 4.1 GGNN-based Bicycle-station Location Prediction

We introduce the Gated Graph Neural Network (GGNN) model [21] to train the large-scale historical graph sequence datasets of each city and predict the bicycle station layout in the next time period. The structure of the GGNN-based bicycle-station location prediction model is shown in Figure 7.

- **Input:** We collect large-scale historical cycling trajectory records from the DL-PBS network in different cities and administrative regions and then construct a set of bicycle station graph datasets  $GS = \{G_1, \dots, G_t\}$  for each of them. Each bicycle station graph  $G_t \in GS$  in a historical time period is used as an input of the GGNN model.
- **Output:** Given an input  $G_t$ , the output of the GGNN model is a predicted bicycle station graph  $G_{t+1}$  for the  $(t + 1)$ -th time period. From  $G_{t+1}$ , we can obtain the location of the predicted bicycle stations and the number of bicycles needed at each station.

(1) The forward-propagation prediction process. We represent the bicycle station graph sequence dataset  $GS = \{G_1, \dots, G_t\}$  as  $X = \{x_1, \dots, x_t\}$  and use each graph  $x_t = G_t$  as the input of the GGNN model. We use the Gated Recurrent Unit (GRU) cells as the gate layer of the GGNN model. Let  $H = \{h_1, \dots, h_t\} \in \mathbb{R}^{D \times n}$  be the hidden state matrix of the GRU module, where  $D$  is the

dimension of the hidden state of each unit. For the input  $x_t$  of the  $t$ th time period, the values of the reset gate and update gate are calculated as:

$$\begin{aligned} r_t &= \sigma(W_r \odot [h_{t-1}, x_t] + b_r), \\ z_t &= \sigma(W_z \odot [h_{t-1}, x_t] + b_z), \end{aligned} \quad (8)$$

where  $W_r$  and  $b_r$  are the weight-parameter and bias matrices of the reset gate,  $W_z$  and  $b_z$  are those of the update gate, and operation  $\odot$  indicates an element-wise multiplication.  $\sigma()$  is a sigmoid activation function, and  $\sigma(x) = (1 + e^{-x})^{-1}$ . Based on the reset and update gates, we further calculate the values of the hidden and output layers:

$$\begin{aligned} \tilde{h}_t &= \tanh(W_{\tilde{h}} \odot [r_t \times h_{t-1}, x_t]), \\ h_t &= (1 - z_t) \times h_{t-1} + z_t \times \tilde{h}_t, \\ y_t &= \sigma(W_o \odot h_t). \end{aligned} \quad (9)$$

In this way, for each input  $x_t = G_t$ , we can obtain the corresponding output  $y_t$  from the GGNN model and treat it as the predicted bicycle station graph  $G_{t+1}$  for the  $(t + 1)$ -th time period.

(2) The backward-propagation training process. In the previous process, we obtain  $y_t = G_{t+1}$  for each input  $x_t = G_t$  in each period  $t$ . Benefiting from large-scale historical bicycle station graph sequence models  $G_1 \sim G_t$ , we can continuously train the GGNN model to be stable and convergent by comparing the predicted graph value of each historical period with the actual value. Let  $y_t$  be the predicted graph and  $y_d$  be the actual graph of the bicycle stations in the  $(t + 1)$ -th time period. The core loss functions between different gates and layers are calculated as follows:

$$\begin{aligned} \delta_{y,t} &= (y_d - y_t) \odot \sigma', \\ \delta_{h,t} &= \delta_{y,t} W_o + \delta_{z,t+1} W_{zh} + \delta_{t+1} W_{\tilde{h}h} r_{t+1} + \delta_{h,t+1} W_{rh} + \delta_{h,t+1} (1 - z_{t+1}), \\ \delta_{z,t} &= \delta_{t,h} (\tilde{h} - h_{t-1}) \odot \sigma', \\ \delta_t &= \delta_{h,t} \odot z_t \odot \tanh', \\ \delta_{r,t} &= [(\delta_{h,t} \odot z_t \odot \tanh') W_{\tilde{h}h}^-] \odot \sigma', \end{aligned} \quad (10)$$

where  $W_{rx}$  is the weight matrix between the reset gate and the input layer,  $W_{rh}$  is the weight matrix between the reset gate and the previous hidden layer, and  $W_r$  is the join link of  $W_{rx}$  and  $W_{rh}$ :

$$\begin{aligned} W_r &= W_{rx} + W_{rh}, \\ W_z &= W_{zx} + W_{zh}, \\ W_{\tilde{h}} &= W_{\tilde{h}x} + W_{\tilde{h}h}. \end{aligned} \quad (11)$$

We use the above equations to iteratively update the weight matrixes, such as  $W_r$ ,  $W_z$ ,  $W_{\tilde{h}}$ ,  $W_h$ , and  $W_o$ . In this way, the weight parameters between different gates and layers are successively updated to obtain a stable and convergent GGNN model. The detailed steps of GGNN-based bicycle station prediction are presented in Algorithm 4.1.

## 4.2 Bicycle Station Layout Recommendation

Based on the prediction results of bicycle-sharing demand, we can obtain the location of bicycle stations and the number of bicycles needed at each station in the next period. As mentioned above, in a DL-PBS network, users can not only find available bicycles at any nearest location via GPS positioning, but also return bicycle to anywhere near their destination. Given that users may return public bicycles to locations that are not permitted by the government, the bicycle stations

**ALGORITHM 4.1:** GGNN-based bicycle station prediction**Input:** $GS = \{G_1, \dots, G_t\}$ : A historical graph sequence model of DL-BPS bicycle stations;**Output:** $G_{t+1}$ : the predicted bicycle station graph for the  $(t + 1)$ -th time period.

- 1: initialize the weight parameters of the GGNN model;
- 2: **for** each iteration epoch  $i$  **do**
- 3:   **for** each graph dataset  $G_t$  in  $GS$  **do**
- 4:     forward propagation and predict  $y_t \leftarrow \text{GGNN}(G_t)$ ;
- 5:     calculate the loss functions using Equation (10);
- 6:     update the weight parameters of the GGNN model using Equation (11);
- 7: save the trained GGNN model;
- 8: predict bicycle station graph  $G_{t+1} \leftarrow \text{GGNN}(GS)$ ;
- 9: **return**  $G_{t+1}$ .

predicted from these records may be located outside of government-permitted parking positions. Therefore, we need to fine-tune the location of bicycle stations by matching the predictions with the government's urban management plan. Specifically, if a predicted station is in a permitted area, set it as a bicycle station, otherwise, we fine-tune the location of the current station to the nearest permitted area.

Let  $G_{t+1} = (V, E, D, W)$  be a predicted bicycle station graph model for the subsequent period  $t + 1$ , where  $V = \{v_1, v_2, \dots, v_n\}$  represents a set of candidate bicycle stations. Each station  $v_i = (\varphi_i, \psi_i, n_i)$  consists of three attributes: the station location (longitude  $\varphi_i$  and latitude  $\psi_i$ ) and the number  $n_i$  of bicycles needed at this station. Let  $P = \{p_1, p_2, \dots, p_m\}$  be a set of urban public places that allow parking of public bicycles (termed as legal parking positions). Each position  $p_j = (\varphi_j, \psi_j, n_j)$  also contains three attributes: the location (longitude  $\varphi_j$  and latitude  $\psi_j$ ) and the maximum number  $n_j$  of bicycles that the position can hold. For each predicted bicycle station  $v_i$ , we find the nearest legal parking position  $p_j \in P$ . There are three situations between  $v_i$  and  $p_j$ .

- Case (a):  $d_{ij} \leq \theta_d$  and  $n_j \geq n_i$ :  $v_i$  is in a legal parking position with sufficient space. We use Equation (3) to obtain the distance  $d_{ij}$  between  $v_i$  and  $p_j$ . Considering the deviation of GIS data acquisition, we set a deviation threshold  $\theta_d$  for position matching. If  $d_{ij} \leq \theta_d$ , then the locations of  $v_i$  and  $p_j$  are considered coincident, that is,  $v_i$  is in  $p_j$ . In addition, if  $n_j \geq n_i$ , then this means that  $p_j$  has sufficient space to accommodate  $v_i$ 's predicted bicycles, and we accept this station and the predicted bicycles without any adjustments.
- Case (b):  $d_{ij} \leq \theta_d$  and  $n_j < n_i$ :  $v_i$  is in a legal parking position with insufficient space. If  $d_{ij} \leq \theta_d$  and  $n_j < n_i$ , it means that the number of bicycles at  $v_i$  exceeds the available space provided by  $p_j$ . We need to split the station  $v_i$  by finding another near legal parking position  $p'_j$  for the extra bicycles  $(n_i - n_j)$ .
- Case (c):  $d_{ij} > \theta_d$ :  $v_i$  is in an illegal position. If  $v_i$  is not in any legal parking position, then we need to fine-tune the location of  $v_i$  to  $p_j$ :  $(\varphi_i, \psi_i) = (\varphi_j, \psi_j)$ . Then, we go to Case (a) or Case (b) to further judge whether the available space of  $p_j$  meets the conditions.

Examples of the relationship between predicted bicycle stations and legal parking positions are shown in Figure 8.

After matching the predicted bicycle stations to the legal parking positions, we fine-tune the stations by updating the predicted station locations and the number of bicycles at each station. Finally, we obtain the recommendation scheme of bicycle station layout for the current urban area. Algorithm 4.2 gives detailed steps of the fine-tuning and recommendation process of the predicted bicycle stations.

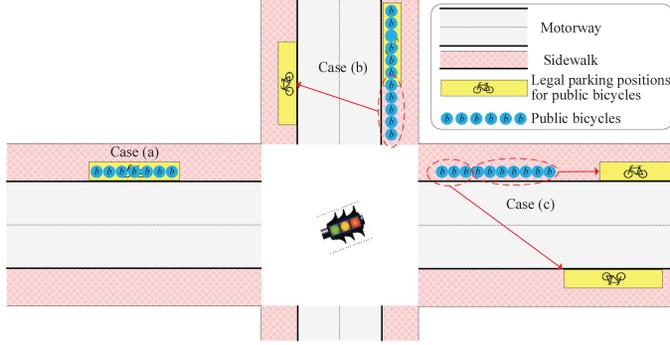


Fig. 8. Examples of the relationship between predicted bicycle stations and legal parking positions.

---

#### ALGORITHM 4.2: Bicycle station fine-tuning and recommendation

---

**Input:**

- $G_{t+1}$ : The predicted graph model of bicycle stations;
- $P$ : A set of legal parking positions for public bicycles;
- $\theta_d$ : Distance deviation threshold of station location matching;

**Output:**

- $G'_{t+1}$ : The fine-tuned bicycle station layout scheme.
- 1: load predicted bicycle stations  $V$  from  $G_{t+1}$ ;
  - 2: **while**  $V$  is not empty **do**
  - 3:   **for** each station  $v_i$  in  $V$  **do**
  - 4:     find the nearest legal parking position  $p_j \leftarrow (v_i, P)$ ;
  - 5:     **if**  $d_{ij} \leq \theta_d$  **then**
  - 6:       **if**  $n_j \geq n_i$  **then**
  - 7:         move  $v_i$  from  $G_{t+1}$  to  $G'_{t+1}$ ;
  - 8:       **else**
  - 9:         update the value of vertex  $v_i$  as  $(\varphi_i, \psi_i, n_j)$ ;
  - 10:         move  $v_i$  from  $G_{t+1}$  to  $G'_{t+1}$ ;
  - 11:         find another nearest legal parking position  $p'_j \leftarrow (v_i, P - p_j)$ ;
  - 12:         create a new vertex  $v'_i$  with the attributes of  $(\varphi_{j'}, \psi_{j'}, n_i - n_j)$ ;
  - 13:         append  $v'_i$  to  $G'_{t+1}$ ;
  - 14:     **else**
  - 15:         update the location of station  $v_i$   $(\varphi_i, \psi_i) \leftarrow (\varphi_j, \psi_j)$ ;
  - 16: **return**  $G'_{t+1}$ .
- 

## 5 EXPERIMENTS

### 5.1 Experimental Setup

We collect bicycle GPS datasets and cycling trajectory records from a DL-PBS provider in China. These datasets are collected from 16 administrative regions in Beijing, China, from January 1st, 2018, to December 31st, 2019. According to administrative divisions, these datasets are divided into 16 spatial subsets, and each spatial subset is further divided into multiple temporal subsets by days or weeks. The demand for shared bikes is real-time, and bicycle demand forecasting and station planning are the basis of bicycle dispatching. The total circulation and dispatch frequency of bicycles in different administrative regions are different. In this case, we forecast bicycle demand according to the time period of bicycle dispatching (i.e., daily or weekly).

In the comparison experiments, we use k-fold cross-validation to divide the training set and test set. We set  $k = 5$  and perform 5-fold cross-validation. For each spatial subset of an administrative division, we divide the spatio-temporal subsets into 5 groups. Then, 4 groups are used as the training set and the remaining 1 group is used as the test set. We repeat this process until every 5-fold serves as the test set. Finally, we take the average of the recorded scores as the accuracy of the corresponding algorithm.

## 5.2 Discussion of Experimental Results

In this section, we discuss the experiment results of bicycle drop-off location clustering and bicycle-station location prediction. In three cases, we discuss the experimental results of four administrative regions in Beijing, such as Dongcheng District, Xicheng District, Haidian District, and Fengtai District.

**5.2.1 Case Study 1: Dongcheng and Xicheng Districts.** In the first case, we divide the spatial DL-PBS datasets of Dongcheng and Xicheng Districts into 730 temporal subsets by day, each of which has approximately 418,392 cycling trajectory records. We implement the proposed bicycle drop-off location clustering algorithm on each historical temporal subset and obtain the corresponding clustering results. Based on the clustering result of each temporal subset, we construct a corresponding bicycle-station graph model for each time period and then collect all graph models in different periods to form a graph sequence model. Finally, we execute the GGNN algorithm on the graph sequence model to predict the bicycle-station graph model in the next time period. Taking the graph models from October 22, 2018, to October 24, 2018, as an example, the experimental results are shown in Figure 9.

Figure 9(a) shows the clustering results of bicycle drop-off locations in Dongcheng and Xicheng Districts during October 22, 2018. According to the number of bicycles that can be accommodated at each bicycle station, we divide these stations into four levels: micro station (each station can accommodate 5 to 10 bicycles), small station (10 to 20 bicycles), medium station (20 to 30 bicycles), and large station (more than 30 bicycles). In this way, from the clustering results, we obtain 145 micro stations, 137 small stations, 59 medium stations, and 5 large stations. Bicycle drop-off location clusters with less than 5 bicycles will be ignored.

Based on the clustering results, we include the cycling records between the clusters and build a bicycle-station graph model for October 22, 2018. Then, we calculate the revenue and utility of each candidate station and remove 43 inferior stations from the graph  $G_{20181022}$ , as shown in Figure 9(b). In the same way, we build a bicycle-station graph model for October 23, 2018, based on the corresponding clustering results. As shown in Figure 9(c), there are 91 micro stations, 136 small stations, 49 medium stations, and 18 large stations in the graph model  $G_{20181023}$ . By comparing the graphs  $G_{20181022}$  and  $G_{20181023}$ , we can see that the number of bicycles at almost every station has changed dynamically. From  $G_{20181022}$  and  $G_{20181023}$ , 6 micro stations and 2 small stations disappeared, but 2 new micro stations appeared. As bicycles flow, 13 micro stations expand to small stations; 6 small stations expand to medium stations and 4 shrink to micro stations; 13 medium stations expand to large stations and 3 shrink to small stations. In this way, we continue to perform the processes of bicycle drop-off location clustering and bicycle-station graph modeling for each temporal subset and create a graph sequence model.

After obtaining the graph sequence model of Dongcheng and Xicheng Districts, we train the GGNN model using the graph sequence data and get a predicted bicycle-station graph  $G_{20181024}$  in the next day (October 24, 2018), as shown in Figure 9(d). Then, we fine-tune the location of bicycle stations by matching the predictions with Dongcheng and Xicheng Districts' urban management plan. Finally, we get the location of each bicycle station and the number of bicycles needed.

**5.2.2 Case Study 2: Haidian District.** In the second case, we discuss the experimental results of the spatial DL-PBS datasets in Haidian District, Beijing. The spatial DL-PBS dataset is divided into 104 temporal subsets by week, each of which has approximately 3,521,486 cycling trajectory records. The experimental results of bicycle drop-off location clustering, bicycle-station graph construction, and bicycle station prediction are shown in Figure 10.

Figure 10(a) shows the clustering results of bicycle drop-off locations in Haidian District during the first week of June 2018. We obtain 123 micro stations, 119 small stations, 45 medium stations,

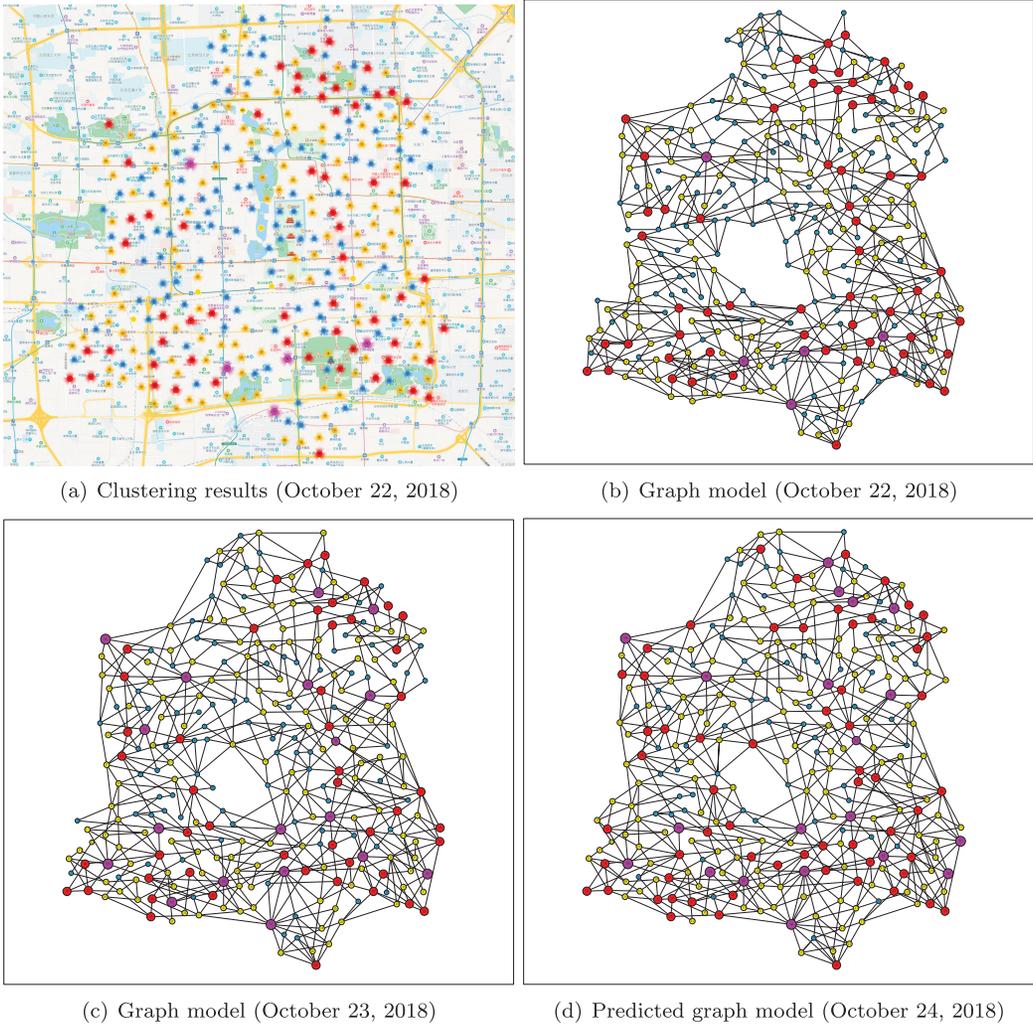


Fig. 9. Results of bicycle drop-off location clustering and bicycle station prediction in the Dongcheng and Xicheng Districts. (a) is the clustering results of bicycle drop-off locations on October 22, 2018, where four types of bicycle stations are detected, such as micro stations, small stations, medium stations, and large stations. (b) is the bicycle station graph model based on the clustering results on October 22, 2018. (c) is the bicycle station graph model based on the clustering results on October 23, 2018. (d) is the predicted bicycle-station graph for October 24, 2018.

and 7 large stations. Based on the clustering results, we include the cycling records between the clusters and build a bicycle-station graph model for this period. Then, we calculate the revenue and utility of each candidate station and remove 28 inferior stations from the graph  $G_{2018061w}$ , as shown in Figure 10(b). As shown in Figure 10(c), we build the bicycle-station graph model  $G_{2018062w}$  for the second week of June 2018, which contains 139 micro stations, 121 small stations, 41 medium stations, and 25 large stations. We continue to perform the processes of bicycle drop-off location clustering and bicycle-station graph modeling for each temporal subset in the same way and create a graph sequence model. Next step, we train the GGNN model using the graph sequence data and

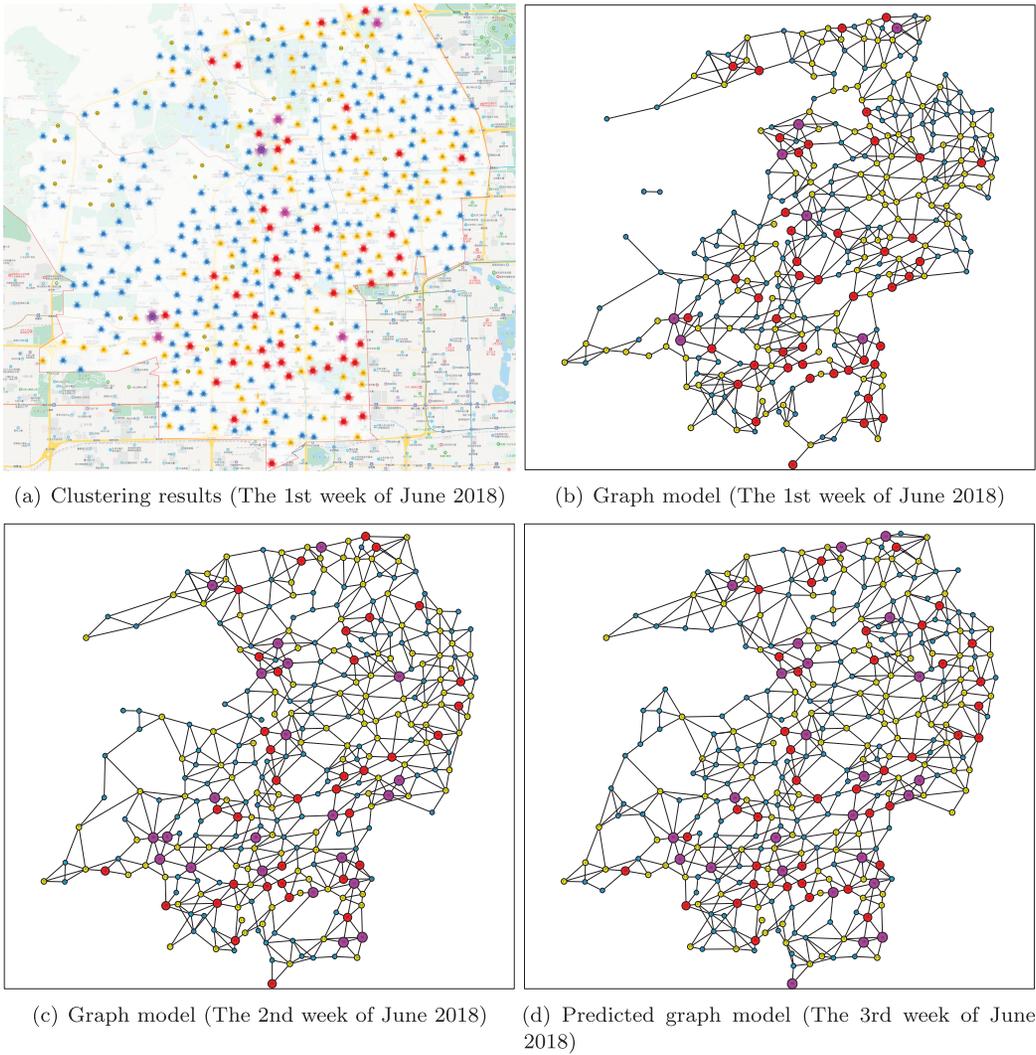


Fig. 10. Results of bicycle drop-off location clustering and bicycle station prediction in the Haidian District. We group the historical bicycle GPS datasets and predict the bicycle station graph by weeks. (a) and (b) are the clustering results of bicycle drop-off locations and the related bicycle station graph model for the 1st week of June 2018, (c) is the graph model for the 2nd week of June 2018, and (d) is the predicted bicycle-station graph for the 3rd week of June 2018.

get a predicted bicycle-station graph  $G_{2018063w}$  in the next period (the third week of June 2018), as shown in Figure 10(d). After fine-tuning the location of bicycle stations with Haidian District’s urban management plan, we get the location of each bicycle station and the number of bicycles needed.

5.2.3 *Case Study 3: Fengtai District.* In the third case, we discuss the experimental results of the spatial DL-PBS datasets in Fengtai District, Beijing. We divide the spatial DL-PBS dataset into 104 temporal subsets by week, each of which has approximately 3,107,753 cycling trajectory records.

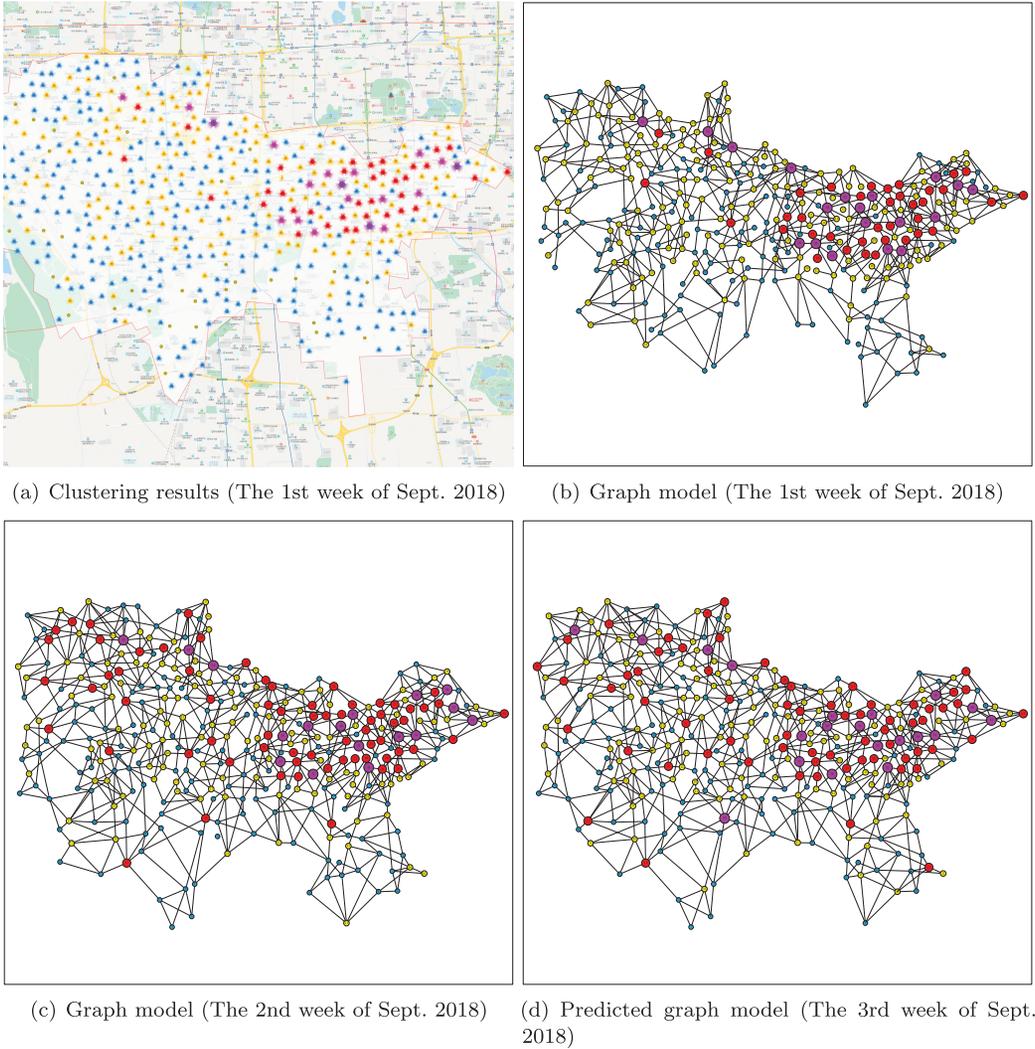


Fig. 11. Results of bicycle drop-off location clustering and bicycle station prediction in the Fengtai District. (a) and (b) are the clustering results of bicycle drop-off locations and the related bicycle station graph model for the 1st week of June 2018, where 131 micro stations, 160 small stations, 37 medium stations, and 19 large stations are detected. (c) is the graph model for the 2nd week of June 2018, where 141 micro stations, 129 small stations, 67 medium stations, and 17 large stations are detected. (d) is the predicted bicycle-station graph for the 3rd week of June 2018, where 119 micro stations, 125 small stations, 67 medium stations, and 20 large stations are predicted.

The experimental results of bicycle drop-off location clustering, bicycle-station graph construction, and bicycle station prediction are shown in Figure 11.

Figure 11(a) shows the clustering results of bicycle drop-off locations in Fengtai District during the first week of Sept. 2018. We obtain 131 micro stations, 160 small stations, 37 medium stations, and 19 large stations. Based on the clustering results, we include the cycling records between the clusters and build a bicycle-station graph model  $G_{2018091w}$  for the current period, as shown in Figure 11(b).

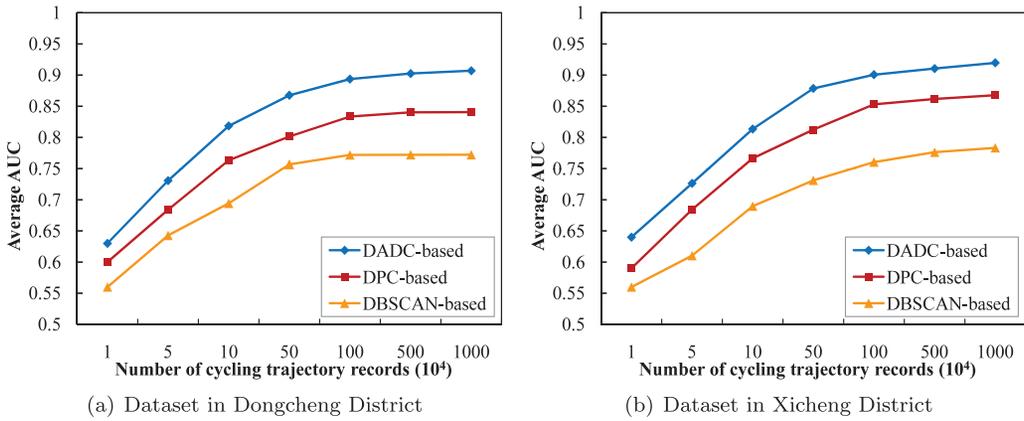


Fig. 12. Accuracy evaluation of bicycle drop-off location clustering.

We continue to build the bicycle-station graph model  $G_{2018092w}$  for the second week of September 2018, which contains 141 micro stations, 129 small stations, 67 medium stations, and 17 large stations, as shown in Figure 11(c). We continue to perform the above processes in the same way and create a graph sequence model. Then, we train the GGNN model using the graph sequence data and get a predicted bicycle-station graph  $G_{2018093w}$  in the next period (the third week of September 2018), as shown in Figure 11(d). Note that  $G_{2018093w}$  is not only predicted by  $G_{2018092w}$  and  $G_{2018093w}$ , but is predicted by all historical graph models in the current spatial dataset. After fine-tuning the location of bicycle stations with Fengtai District's urban management plan, we get the location of each bicycle station and the number of bicycles needed. The experimental results show that by learning a large number of historical graph sequence data, the GGNN model can capture changes in bicycle stations in different periods and accurately predict the bicycle station layout in the next period.

### 5.3 Evaluation of Bicycle Drop-off Location Clustering

To evaluate the accuracy of DADC-based bicycle drop-off location clustering, we use two groups of DL-PBS cycling trajectory records to perform the experiments. We matched the bicycle coordinates in the cycling trajectory records with the actual map and then manually labeled the cluster of each data point in the two datasets. Then, we compare the DADC method [6] with the Density-Peak-based clustering (DPC) [27] and DBSCAN [8] algorithms. The average of the Area Under ROC Curve (AUC) is used as a clustering accuracy indicator by comparing the clustering results with the labels. The experiment results are shown in Figure 12.

Figure 12(a) and (b) show that in all cases, the DADC-based bicycle drop-off location clustering method achieves higher AUC values than DPC and DBSCAN. Note that cycling trajectory data has the characteristics of varying density distribution, that is, there are coexisting areas with obvious different region densities, such as dense and sparse regions (i.e., some stations have more bicycles while other stations have fewer bicycles). DADC detects data points with dense neighbors by calculating the local density. Based on the delta distance, DADC can efficiently identify the density peaks in each region as cluster centers. In contrast, DPC and DBSCAN methods cannot effectively address the above issues, thereby achieving low clustering accuracy. The average AUC value of DADC is 0.82, the average AUC value of DPC is 0.76, and that of DBSCAN is the lowest, which is 0.71. In addition, the AUC value increases significantly with the number of cycling trajectory records. As the number of records increases from 104 to 107, the AUC value of DADC increases

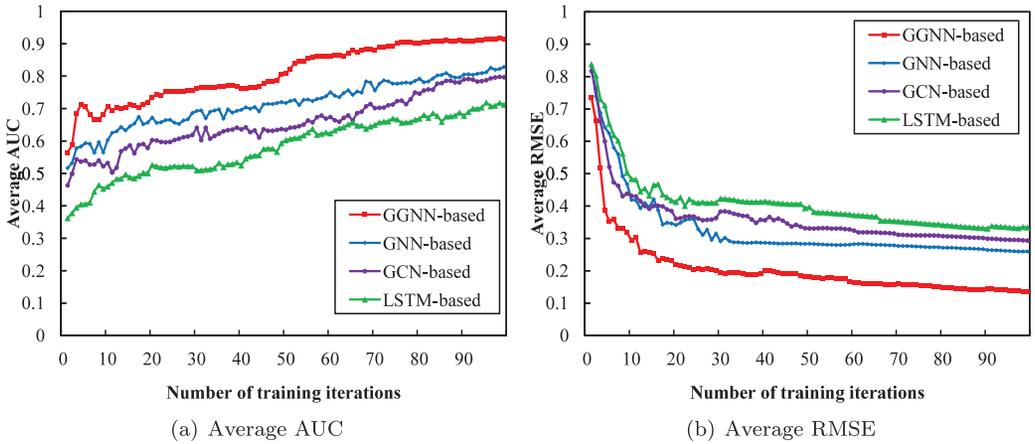


Fig. 13. Performance evaluation of bicycle station prediction.

from 0.63 to 0.91, the AUC value of DPC rises from 0.60 to 0.84, and that of DBSCAN only increases from 0.56 to 0.77. Therefore, in this work, we choose the DADC method to perform bicycle drop-off location clustering.

#### 5.4 Evaluation of Bicycle-station Location Prediction

Based on the previous experimental results, we establish distinct graph sequence models of DL-PBS networks for different administrative regions. To evaluate the performance of the GGNN model, we conduct experiments with these graph sequence models for bicycle station prediction by comparing GGNN [21], GNN [28], GCN [31], and LSTM [12] models. We discuss the performance of the comparison methods in terms of AUC and Root Mean Square Error (RMSE) [14], as shown in Figure 13.

As shown in Figure 13(a), the average AUC value of the GGNN-based bicycle station prediction method increases and reaches convergence with the number of training iterations. Compared to other methods, GGNN can make full use of graph sequence data and capture associations between graphs in different time periods, thereby obtaining higher prediction accuracy. In contrast, GNN and GCN methods separate the input graph at each time point, ignoring the association between these graphs. Although the LSTM method also uses a gated module to memorize the continuous input, it lacks effective processing of the graph structure of the input. For example, after 100 training iterations, the average AUC value of GGNN is 0.91, the average AUC value of GNN is 0.83, the average AUC value of GCN is 0.80, and the average AUC value of LSTM is 0.71. In addition, as shown in Figure 13(b), as the number of training iterations increases, the average RMSE value of GGNN decreases rapidly and converges to a minimum. For example, after 100 iterations, the average RMSE value of GGNN is 0.13, while the average RMSE values of GNN, GCN, and LSTM are 0.26, 0.29, and 0.33, respectively. The experimental results demonstrate that compared with the comparison methods, the GGNN model is suitable for bicycle station prediction and achieves the highest prediction accuracy.

## 6 CONCLUSIONS

In this article, we proposed a bicycle station dynamic planning (BSDP) system, which can dynamically predict DL-PBS bicycle demands and provide the optimized layout of public bicycle stations. First, by clustering large-scale historical cycling trajectory data, we established a

weighted digraph model for bicycle drop-off locations. In addition, by tracking the update of the stations in the time dimension, we further built a graph sequence model based on the bicycle drop-off digraph models. Based on this, the GGNN model was introduced to train the graph sequence model and predict the cycling trajectory and usage requirements in the next period. Finally, according to the requirements of public bicycle use, we provided the optimized layout of DL-PBS bicycle stations. Experiments with actual DL-PBS datasets have verified the proposed BSDP system in terms of feasibility, accuracy, and performance.

In future work, we will focus on downstream applications of the DL-PBS network, such as public bicycle dispatching, cycling trajectory tracking, and faulty bicycle detection.

## REFERENCES

- [1] Ander Audikana, Emmanuel Ravalet, Virginie Baranger, and Vincent Kaufmann. 2017. Implementing bikesharing systems in small cities: Evidence from the Swiss experience. *Transp. Polic.* 55 (2017), 18–28.
- [2] Remy Cazabet, Pablo Jensen, and Pierre Borgnat. 2017. Tracking the evolution of temporal patterns of usage in bicycle-sharing systems using nonnegative matrix factorization on multiple sliding windows. *Int. J. Urban Sci.* 13, 4 (2017), 703–712.
- [3] Cen Chen, Kenli Li, Sin G. Teo, Guizi Chen, Xiaofeng Zou, Xulei Yang, Ramaseshan C. Vijay, Jiashi Feng, and Zeng Zeng. 2018. Exploiting spatio-temporal correlations with multiple 3D convolutional neural networks for citywide vehicle flow prediction. In *ICDM'18*. IEEE, 893–898.
- [4] Cen Chen, Kenli Li, Sin G. Teo, Xiaofeng Zou, Keqin Li, and Zeng Zeng. 2020. Citywide traffic flow prediction based on multiple gated spatio-temporal convolutional neural networks. *ACM Trans. Knowl. Discov. Data* 14, 4 (2020), 1–23.
- [5] Cen Chen, Kenli Li, Sin G. Teo, Xiaofeng Zou, Kang Wang, Jie Wang, and Zeng Zeng. 2019. Gated residual recurrent graph neural networks for traffic prediction. In *AAAI'19*, Vol. 33. 485–492.
- [6] Jianguo Chen and Philip S. Yu. 2019. A domain adaptive density clustering algorithm for data with varying density distribution. *IEEE Trans. Knowl. Data Eng.* 99, 1 (2019), 1–12. DOI: <https://doi.org/10.1109/TKDE.2019.2954133>
- [7] Chao Deng, Jiaying Wang, and Weigang Zheng. 2015. Layout optimizing of public bicycle stations based on AHP in Wuhan. *Appl. Mech. Mater.* 737 (2015), 896–902.
- [8] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD'96*, Vol. 96. 226–231.
- [9] Côme Etienne and Oukhellou Latifa. 2014. Model-based count series clustering for bike sharing system usage mining: A case study with the Velib' system of paris. *ACM Trans. Intell. Syst. Technol.* 5, 3 (2014), 39–41.
- [10] Nicolas Gast, Guillaume Massonnet, Daniel Reijsbergen, and Mirco Tribastone. 2015. Probabilistic forecasts of bike-sharing systems for journey planning. In *CIKM'15*. ACM, 703–712.
- [11] Nicolas Gast, Guillaume Massonnet, Daniel Reijsbergen, and Mirco Tribastone. 2015. Public bicycle prediction based on generalized regression neural network. In *IOV'15*. Springer, 363–373.
- [12] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9, 8 (1997), 1735–1780.
- [13] Aude Hofleitner, Ryan Herring, Pieter Abbeel, and Alexandre Bayen. 2012. Learning the dynamics of arterial traffic from probe data using a dynamic Bayesian network. *IEEE Trans. Intell. Transp. Syst.* 13, 4 (2012), 1679–1693.
- [14] Rob J. Hyndman and Anne B. Koehler. 2006. Another look at measures of forecast accuracy. *Int. J. Forec.* 22, 4 (2006), 679–688.
- [15] Yantao Jia, Yuanzhuo Wang, Xiaolong Jin, and Xueqi Cheng. 2016. Location prediction: A temporal-spatial Bayesian model. *ACM Trans. Intell. Syst. Technol.* 7, 3 (2016), 1–25.
- [16] Zhe Jiang, Michael Evans, Dev Oliver, and Shashi Shekhar. 2016. Identifying K primary corridors from urban bicycle GPS trajectories on a road network. *Inf. Syst.* 57 (2016), 142–159.
- [17] Mahdi Khodayar and Jianhui Wang. 2019. Spatio-temporal graph deep neural network for short-term wind speed forecasting. *IEEE Trans. Sustain. Energy* 10, 2 (2019), 670–681.
- [18] Dalibor Krleža and Kresimir Fertalj. 2017. Graph matching using hierarchical fuzzy graph neural networks. *IEEE Trans. Fuzzy Syst.* 25, 4 (2017), 892–904.
- [19] Karim Labadi, Taha Benarbia, Jean-Pierre Barbot, Samir Hamaci, and Abdelhafid Omari. 2015. Stochastic Petri Net modeling, simulation and analysis of public bicycle sharing systems. *IEEE Trans. Autom. Sci. Eng.* 12, 4 (2015), 1380–1395.
- [20] Ron Levie, Federico Monti, Xavier Bresson, and Michael M. Bronstein. 2019. CayleyNets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Trans. Sig. Process.* 67, 1 (2019), 97–109.
- [21] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. 2017. Gated graph sequence neural networks. *arXiv e-prints*, Article arXiv:1511.05493 (June 2017).

- [22] Yexin Li, Yu Zheng, Huichu Zhang, and Lei Chen. 2015. Traffic prediction in a bike-sharing system. In *SIGSPATIAL '15*. ACM, 1–10.
- [23] Lei Lin, Zhengbing He, and Srinivas Peeta. 2018. Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach. *Transp. Res. Pt. C-Emerg. Technol.* 97 (2018), 258–276.
- [24] Yingying Ma, Xiaoran Qin, Jianmin Xu, and Wenjing Wang. 2016. A hierarchical public bicycle dispatching policy for dynamic demand. In *SOLI'16*. IEEE, 162–167.
- [25] Alessio Micheli. 2009. Neural network for graphs: A contextual constructive approach. *IEEE Trans. Neural Netw.* 20, 3 (2009), 498–511.
- [26] Ling Pan, Qingpeng Cai, Zhixuan Fang, Pingzhong Tang, and Longbo Huang. 2019. A deep reinforcement learning framework for rebalancing dockless bike sharing systems. In *AAAI'19*, Vol. 33. 1393–1400.
- [27] Alex Rodriguez and Alessandro Laio. 2014. Clustering by fast search and find of density peaks. *Science* 344, 6191 (2014), 1492–1496.
- [28] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The graph neural network model. *IEEE Trans. Neural Netw.* 20, 1 (2009), 61–80.
- [29] Zhe Xiao, Hock Beng Lim, and Loganathan Ponnambalam. 2017. Participatory sensing for smart cities: A case study on transport trip quality measurement. *IEEE Trans. Ind. Inform.* 13, 2 (2017), 759–770.
- [30] Josh Jia-Ching Ying, Wang-Chien Lee, and Vincent S. Tseng. 2014. Mining geographic-temporal-semantic patterns in trajectories for location prediction. *ACM Trans. Intell. Syst. Technol.* 5, 1 (2014), 1–33.
- [31] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *KDD'18*. ACM, 974–983.
- [32] Longxin Zhang, Kenli Li, Changyun Li, and Keqin Li. 2017. Bi-objective workflow scheduling of the energy consumption and reliability in heterogeneous computing systems. *Inf. Sci.* 379 (2017), 241–256.
- [33] Longxin Zhang, Kenli Li, Weihua Zheng, and Keqin Li. 2017. Contention-aware reliability efficient scheduling on heterogeneous computing systems. *IEEE Trans. Sustain. Comput.* 3, 3 (2017), 182–194.

Received April 2020; revised December 2020; accepted December 2020