This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3145979, IEEE Transactions on Mobile Computing

RedPacketBike: A Graph-Based Demand Modeling and Crowd-Driven Station Rebalancing Framework for Bike Sharing Systems

Hang Zhu†, Tieqi Shou†, Ruiying Guo, Zhihan Jiang, Zeyu Wang, Zhiyuan Wang, Zhiyong Yu, *Member, IEEE*, Weijie Zhang, Cheng Wang, *Member, IEEE, Longbiao Chen*, Member, IEEE*

Abstract—Bike-sharing systems have been deployed globally. One of the key issues for high-quality bike-sharing systems is to rebalance city-wide stations to maintain bike availability. Traditional strategies, such as repositioning bikes by trucks and volunteers based on historical riding records, usually operate in fixed paths and limited capacities, lacking the flexibility to cope with the highly dynamic and context dependent riding demands, and usually suffer from high costs and long delays. In this work, we propose RedPacketBike, an incentive-driven, crowd-based station rebalancing framework to effectively recruit participants from hybrid fleets (e.g., volunteer riders and hired trucks) based on the accurate forecast of bike demand leveraging deep learning techniques. First, we propose a spatiotemporal clustering method to extract bike demand hotspots from fluctuating bike usage data. Then, we build a context-aware deep neural network named BikeNet to forecast the trends of bike demand hotspots, simultaneously modeling the spatial correlations by graph convolution networks (GCN), the temporal dependencies by long short-term memory networkss (RNN), and the contextual factors by autoencoders (AE). Finally, we propose a reinforcement-learning-based method to find optimal station rebalancing schemes by generating station rebalancing tasks with an integer linear programming (ILP) algorithm and allocating tasks to participants from hybrid fleets with dynamic incentive designs and reward expectations. Experiments using real-world bike-sharing system data collected from Citi Bike in New York City and Mobike in Xiamen City validate the performance of our framework, achieving a demand forecast error below 4.171 measured in MAE, and a 17.2% improvement of station availability by simulations with real-world parameter settings, outperforming the state-of-the-art baselines.

Index Terms—mobile crowdsensing, graph neural networks, bike sharing systems

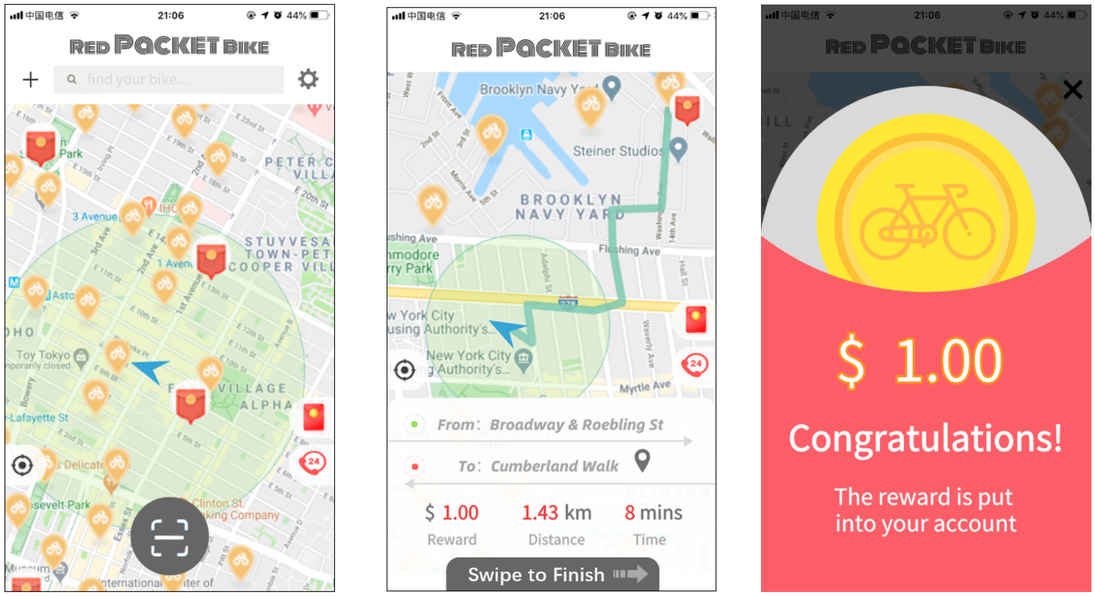
1 Introduction

Bike sharing systems are booming globally as a green transportation means to alleviate the urban traffic congestion problem. Such systems allow users to rent and return bikes freely to tackle the last mile issue [1]. However, due to the uncertainty of dynamic human mobility patterns, many bike stations suffer from over-demand problems and the user has no bikes to use or no docks to return bikes [2]. The over-demand stations may greatly affect the users' experiences because they need to find another available station to rent or return the bike, which reduces the willingness of user participation in the bike-sharing system [3]. Therefore, it is essential for bike-sharing system operators to keep bike stations in balance to avoid over-demand issues.

- † *Both authors contributed equally to this research.*
  * *Corresponding author.*
- *H. Zhu, T. Shou, R. Guo, Z. Wang, Z. Wang, C. Wang and L. Chen are with Fujian Key Laboratory of Sensing and Computing for Smart Cities (SCSC), School of Informatics, Xiamen University, China (e-mail: zhuhang@stu.xmu.edu.cn, shoutieqi@stu.xmu.edu.cn, longbiaochen@xmu.edu.cn).*
- *Z. Jiang is with Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, China (e-mail: zhjiang@connect.hku.hk).*
- *Z. Yu and W. Zhang are with Fujian Key Laboratory of Network Computing and Intelligent Information Processing, College of Mathematics and Computer Science, Fuzhou University, China (e-mail: {yuzhiyong, n190327095}@fzu.edu.cn).*

Companies running bike-sharing systems have taken various strategies to solve the over-demand problems [4], [5], such as sending a fleet of trucks to reposition bikes among stations regularly between rush hours [6], [7]. However, the truck-based approach has a fixed transportation path and limited capacity, which cannot cope with the highly fluctuating bike demands, and leads to high operation costs. Recently, operators have proposed to incorporate volunteer riders to help redistribute bikes to stations in need with bonus and incentives [8], however these approaches usually allocate rebalancing tasks and incentives simply based on statistics of historical station status, which usually does not perform well when dealing with surging bike demands under abnormal social and traffic events. Therefore, operators need to find a cost-effective and demand-responsive station rebalancing approach to solve these problems.

Fortunately, thanks to the rapid evolution of mobile computing technologies, users' travel demands can be collected and analyzed when they interact with bike-sharing systems [9]. Meanwhile, the emergence of crowdsensing paradigm has presented new possibilities to incentivize users to participate in station rebalancing [10], [11]. In this paper, we propose RedPacketBike, an *incentive-driven, crowd-based* station rebalancing framework to effectively recruit participants from hybrid fleets (e.g., volunteer riders and hired trucks) based on the *accurate forecast* of bike demand leveraging deep learning techniques. Our approach offers



(a) The participant opens the RedPacketBike app, the distribution of bike stations and red packets are displayed.

(b) The participant clicks a red packet, the rebalancing task details are displayed, including distance, reward, time, etc.

(c) The incentive is paid when the participant completes the rebalancing task.

Fig. 1: The design and application scenario of RedPacketBike, an incentive-driven, crowd-based station rebalancing application.

a cost-effective and demand-responsive station rebalancing solution by encouraging riders and hired truck drivers to rent or return bikes in specific stations in exchange for monetary incentives. As shown in Fig. 1, the distribution of the bike stations and the incentives (*red packets*) are displayed when a participant launches the crowd rebalancing mobile application. The participant can browse the task details, including rebalancing distance, monetary reward, navigation route and time, etc. After the participant completes the rebalancing task, the reward will be paid. In order to design and implement the RedPacketBike framework, we need to address the following issues.

First, *it is difficult to accurately forecast station-level bike demands*. Due to the fact that bike riders usually choose a station near their origins or destinations on an ad hoc basis, the bike usage patterns at the station level are highly dynamic and fluctuating [12]. Moreover, bike usage patterns are usually context-dependent, being affected not only by common contextual factors (e.g., rush hours or weather conditions) but also by opportunistic contextual factors (e.g., social and traffic events). Therefore, directly forecasting hourly bike usage demand at station level usually do not result in satisfactory accuracy. To address this issue, we propose to cluster stations into *demand hotspots* and extract *demand trends* to mitigate the fluctuations. We then propose to build a context-aware deep neural network to accurately forecast the trends of bike demand hotspots, simultaneously modeling the spatial correlations, temporal dependencies, and contextual factors of bike demand.

Second, *it is non-trivial to effectively incentivize potential participants for station rebalancing tasks*. City-wide bike station rebalancing task generation, i.e., generating an optimal set of station pairs to reposition bikes so as to balance city-wide stations, is usually considered an NP-hard problem [13]. Moreover, various factors, such as weather conditions and riding distances, may have significant impact on users' willingness to participate in rebalancing tasks [8], [14]. Therefore, recruiting participants and allocating tasks under the above-mentioned constrained pose great challenges. In this paper, we propose to formulate this problem as a reinforcement learning task [7], aiming at training an optimal task allocation agent with rebalancing tasks generated by an integer linear programming (ILP) algorithm, and incentives dynamically calculated based on spatiotemporal contexts and participants' willingness.

With the above-mentioned research objectives and issues, the main contributions of this paper are:

- Our work is a promising step towards a cost-effective and demand-responsive station rebalancing framework leveraging mobile computing and crowdsensing techniques. We model mobile-sensed bike usage data with graph neural networks to accurately forecast bike demand trends, and leverage reinforcement learning mechanisms to effectively incentivize participants to rebalance over-demand stations. Such a framework can also be applied to optimize various mobile systems, e.g., cell communication networks.
- We propose RedPacketBike, an incentive-driven, crowd-based station rebalancing framework to effectively recruit participants from hybrid fleets based on the accurate forecast of bike demand leveraging deep learning techniques. First, we propose a spatiotemporal clustering method to extract bike demand hotspots and trends from fluctuating bike usage data. Then, we build a context-aware deep neural network named *BikeNet* to forecast the trends of bike demand hotspots, simultaneously modeling the spatial correlations by graph convolution net-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3145979, IEEE Transactions on Mobile Computing

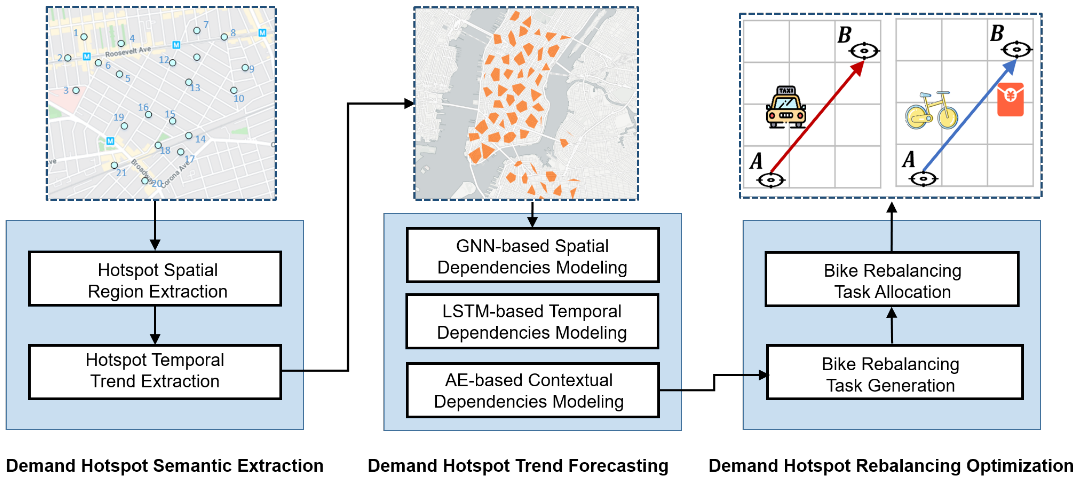


Fig. 2: Overview of the framework.

works (GCN), the temporal dependencies by long short-term memory networks (LSTM), and the contextual factors by autoencoders (AE). Finally, we propose a reinforcement-learning-based method to find optimal station rebalancing schemes by generating station rebalancing tasks with an integer linear programming (ILP) algorithm and allocating tasks to participants from hybrid fleets with dynamic incentive designs and reward expectations.

- We evaluated our proposed framework using real bike-sharing system data collected from the Citi Bike System in New York City and the Mobike System in Xiamen City. Results on demand forecasting and station rebalancing validate the effectiveness of our approach, achieving a demand forecast error below 4.171 measured in MAE, and a 17.2% improvement of station availability by simulations with real-world parameter settings, outperforming the state-of-the-art baselines.

2 Framework Overview

As shown in Fig. 2, we propose RedPacketBike, a three-phase incentive crowd-based station rebalancing system to accurately forecast the fine-grained bike demand trends and effectively rebalance over-demand bike stations leveraging graph neural network and crowdsensing techniques.

In the demand hotspot extraction phase, we first extract spatial region of bike demand hotspot, and then extract the temporal trends of bike demand hotspot, realizing the spatial and temporal modeling of bike demand hotspot. In the demand hotspot trend forecasting phase, we employ BikeNet to forecast the fine-grained bike demand trends, simultaneously modeling the spatial dependencies by graph convolution networks (GCN), modeling the temporal dependencies by long short-term memory (LSTM) and contextual dependencies by autoencoder (AE). In the demand hotspot rebalancing optimization phase, we first model the rebalancing task generation problem to the transportation problem and then propose an integer linear programming-based solution to generating the station rebalancing task with the lowest cycling distance objectives. Then we allocate station rebalancing tasks to users based on the temporal difference algorithm to optimize the rewards of users participating.

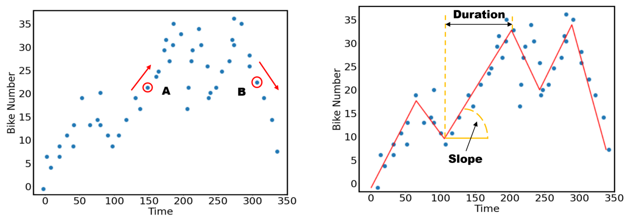
3 Bike Demand Hotspot Extraction

In the demand hotspot extraction phase, our goal is to accurately extract hotspot spatial regions and obtain the corresponding hotspot temporal trends. We need to clarify the bike demand first. The bike demand is calculated by bike flow. There are two types of bike flow: inflow and outflow, the inflow is the number of bikes returned to the station, and the outflow is the number of bikes rented from the station. The bike demand at station i is defined as outflow minus inflow during time span Δt. The time span is the duration of observation data, each time span lasts for a period of time.

Existing works focuse on the demand for the single bike station, does not consider that the formation of demand will eventually converge into multiple demand hotspots. Besides, the number of bikes at one hotspot often changes drastically within a short time interval, while the traditional fixed time interval processing ignores fine-grained temporal characteristics and ignores changes in demand at different times during the same period. For example, the time series of bike demands in Fig. 3(a) is from the bike station in the business district. Although the values of point A and point B are approximately the same, their trends are quite different because A is in an upward pattern while B is in a descending pattern [15]. These overlooked points are the key elements in the process of station rebalancing. Therefore, in the demand hotspot extraction phase, our goal is to accurately extract the hotspot spatial regions and hotspot temporal trends.

According to the bike usage in the real world, demand hotspots show the spatiotemporal characteristics at the semantic level. On the spatial dimension, the origin and destination of users' riding have semantic information of their location, which is a spatial region. On the temporal dimension, the duration of hotspots is not equal time interval, but has a certain duration range, which shows the trend of time in semantics. In this work, we propose a semantic extraction method of bike demand hotspots based on spatial and temporal characteristics.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3145979, IEEE Transactions on Mobile Computing



(a) Time series sequence of bike demand from the station in the business district.

(b) An illustrative example of bike demand trend represents the slope and duration.

Fig. 3: Bike demand of a station change over time. The values of point A and point B are approximately the same, but their trends are quite different because A is in an upward trend while B is in a downward trend. We represent the trend as slope and duration.

3.1 Hotspot Spatial Region Extraction

We use a weighted undirected graph $G = (V, E)$ to represent the bike-sharing system, where V denotes the set of nodes representing the bike stations, and E represents the edges between station pairs. The weight of edge $e_{i,j} = \{v_i, v_j\} \in E$ is defined as $W(v_i, v_j)$. Here, we calculate the geographical distance between station v_i and station v_j as the weight of edge.

We cluster bike stations according to the geographical location, and the hotspots are obtained as the research object of the subsequent bike demand forecasting and rebalancing optimization. In the implementation, the weighted graph G is used to model the relationship between bike-sharing sites. When the road network distance between station v_i and station v_j is less than the distance threshold κ, station v_i and station v_j have edges.

$$a_{ij} = \begin{cases} 1, & if\ dist\,(v_i, v_j) \le \kappa \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Secondly, according to the similarity of bike demand changes and POI features, the edge weight of the weighted graph is calculated. Similarity of bike demand changes $w_p\,(v_i, v_j)$ is calculated by the Pearson correlation coefficient $correlation$.

$$w_p\,(v_i, v_j) = correlation\,(X_{v_i},\ X_{v_j}) \tag{2}$$

Similarity of POI features $w_c\,(v_i, v_j)$ is calculated by the cosine distance $similarity$.

$$w_c\,(v_i, v_j) = similarity\,(Z_{v_i},\ Z_{v_j}) \in [0, 1] \tag{3}$$

Where Z_{v_i} represents the statistical vector of the POI category within 1 km of the area v_i. POI is the abbreviation of Point of Interest, such as transportation stations, hospitals, schools, etc.

With bike stations as nodes, the edge of the graph is established according to the distance between stations $a_{i,j}$, similarity of bike demand changes $w_p\,(v_i, v_j)$ and similarity of POI features $w_c\,(v_i, v_j)$.

$$w_{ij} = a_{ij}\left[\mu w_p\,(v_i, v_j) + (1-\mu)\,w_c\,(v_i, v_j)\right] \tag{4}$$

where μ is the weight factor, controls the influence degree of two factors.

In order to make the adjacent bike stations converge into multiple spatial hotspot regions, so that stations in the same region have the similar demand change pattern and poi feature similarity, distance constrained clustering algorithm (DCCA) [12] is used to achieve the spatial clustering of the stations.

3.2 Hotspot Temporal Trend Extraction

We define $X \in \mathbb{R}^{N_t \times N_s}$ is the node values matrix of graph G, the value of each node $v \in V$ is $X_v^{(t)}$, where N_t is the number of time spans, and $N_s = |V|$ the number of stations in the system. In order to process fine-grained temporal characteristics and pay attention to changes in demand at different times during the same period, we firstly use the cubic spline interpolation method [16] to smooth the obtained discrete point sequence, and then extracts the hotspot temporal trend based on the peak detection algorithm.

Specifically, in the cubic spline interpolation stage, we assume that the realistic trend is $f(x)$, where x represents time. According to the given interpolation interval $a = x_0 < x_1 < ... < x_n = b$, if $S(x) \in C^2[a, b]$, and in each interval $[x_k, x_{k+1}]$ is a cubic polynomial, then S(x) is called a cubic spline function on this interval. Furthermore, if $S(x) = f(x_k) = y_k, k = 0, 1, 2, ..., n$, then $S(x)$ is called the cubic spline interpolation function of $f(x)$ on $[a, b]$. S(x) is as follows:

$$S(x) = a_i + b_i x + c_i x^2 + d_i x^3, i = 1, 2, ..., n \tag{5}$$

Where a_i, b_i, c_i, d_i are coefficients. According to Continuity Conditions and Boundary Conditions [16], we can calculate the expression of the interpolation function $S(x)$. In this way, we interpolate the discrete bike-sharing demand points to obtain the cubic spline interpolation function $S(x)$.

In the hotspot temporal trend extraction stage, we focus on the moments when the demand for bikes changes greatly, because the general scheduling strategy is to dispatch bikes from places where the number increases sharply to places where the number decreases sharply. Correspondingly, we need to obtain two types of key points that extracted from $S(x)$ first: the peak formed by a sharp increase and the trough formed by a sharp decrease in demand trend.

Specifically, we use the Automatic Multiscale-based Peak Detection (AMPD) algorithm [17] to extract hotspot temporal trends. It detects the peak value by calculating and analyzing the Local Maxima Scalogram (LMS), which has the advantages of fewer threshold parameters, better versatility, higher detection efficiency, and strong robustness to high and low-frequency noise compared to other methods.

After using the AMPD algorithm to detect the peaks and troughs, we obtain all the key points. In this work, we are keen on the trend pattern of time series, i.e. upward, unchanged, and downward. Therefore, we calculate the time interval between two key points to get the duration of each demand trend change and record the slope of key points, as shown in Fig. 3(b). We utilize the slope and duration to characterize the trend pattern [15]. Given the historical trend pattern, we expect to forecast the future trend of bike demand.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3145979, IEEE Transactions on Mobile Computing

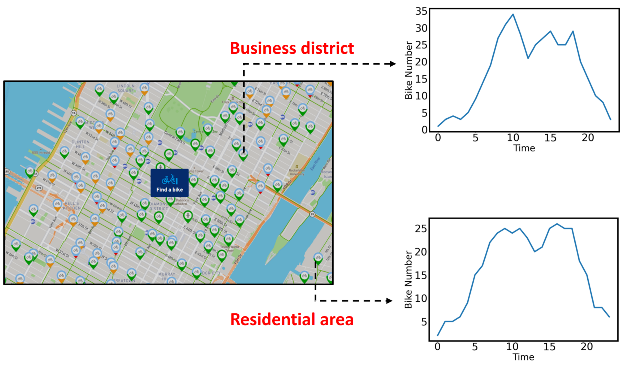


Fig. 4: The case of bike demand pattern in two different bike stations from New York City on Feb 1, 2015, observed hourly based on the Citi Bike system.

4 Demand Hotspot Trend Forecasting

In the demand hotspot trend forecasting phase, our goal is to accurately forecast the fine-grained trend pattern of over-demand stations in the future. However, it is not trivial due to the inherent spatial correlation of the distribution of bike stations and temporal dynamics of bike usage patterns. For instance, Fig. 4 shows the one-day bike demand pattern for two types of bike stations in New York City on Feb 1, 2015. The station in the business area usually observes riding peaks during rush hours, while the station in the residential area is heavily used after getting off work. Traditional bike demand forecast models using time series analysis technology (e.g., ARIMA models [18] and artificial neural networks [19]) cannot achieve high forecast accuracy, since the bike usage patterns demonstrate strong intrinsic spatiotemporal contextual dependency.

For spatial correlation modeling, traditional forecast models usually divide the city space into multiple regular grids, mapping the data into Euclidean domains, which can capture the spatial features by convolutional neural networks (CNN) [20]. However, in the station rebalancing problem, regular grids can not model the scattered and irregular bike station distribution. Recently, graph-based deep learning methods perform well in non-Euclidean domains. Therefore, we introduce graph structures to model the spatial patterns of the bike station distribution and employ graph convolutional networks (GCN) [21] to capture the potential spatial correlations of graph structure data.

For temporal contextual dependencies, the pattern of bike demand is affected by closeness (e.g. few bikes available in the rush hour), period (e.g. the daily and weekly routine of riding are similar), and trend (e.g. the number of rides decreases in winter), which requires the model to response the long short-term dependency. We introduce one Recurrent Neural Network (RNN) model called Long Short-Term Memory (LSTM) [22] to capture the temporal dynamics of each bike station due to its great performance in practice.

In addition, there are some external contextual factors that will affect the demand for bikes, such as weather and social events. In order to model these external features at the same time, we introduced AutoEncoder (AE) to achieve it. AutoEncoder is a kind of neural network for unsupervised learning and efficient coding. It uses back propagation algorithm to learn the potential representation of input and reconstruct the output to make the output value equal to the input value, so that we can input external features and get their representation.

In general, we propose a context-aware deep neural network named BikeNet to capture the spatiotemporal contextual dependencies of bike demands to accurately forecast the demand in the future. The details are elaborated as follows.

4.1 GNN-based Spatial Dependencies Modeling

The goal of the fine-grained bike demand trends forecast is to predict the trend of bike demand in the future based on previous trend observations. To this end, we model the city-wide bike-sharing system as *multiple spatiotemporal graphs*. We construct three kinds of graphs: spatial proximity graph G_n, spatial pattern graph G_p and spatial function graph G_f.

According to Tobler's first law of geography [23], the relationship between the closer regions is greater. Therefore, when building graphs, we first consider the relationship between neighboring areas. We build spatial proximity graph G_n using a gaussian kernel function to deal with distance, the formula for calculating the edge weight w_n is as follows:

$$w_n\left(v_i,\ v_j\right) = \begin{cases} \exp\left(-\frac{[dist(v_i,v_j)]^2}{2\sigma^2}\right) & \text{if } dist\left(v_i,v_j\right) \leq \kappa \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Furthermore, the adjacency matrix A_n of spatial proximity graph G_n is defined as follows:

$$A_n = \begin{pmatrix} w_n\left(0,0\right) & w_n\left(0,1\right) & \cdots & w_n\left(0,N-1\right) \\ w_n\left(1,0\right) & w_n\left(1,1\right) & \cdots & w_n\left(1,N-1\right) \\ w_n\left(2,0\right) & w_n\left(2,1\right) & \cdots & w_n\left(2,N-1\right) \\ \vdots & \vdots & \ddots & \vdots \\ w_n\left(N-1,0\right) & \cdots & \cdots & w_n\left(N-1,N-1\right) \end{pmatrix} \tag{7}$$

In reality, regions with similar demand patterns are not necessarily close to each other in space. In addition to considering the spatial relationship between neighboring regions, it is also necessary to consider the relationship between regions that are far away but related to demand patterns. We build spatial pattern graph G_p to represent the similarity of bike demand changes, the edge is $w_p\left(v_i,v_j\right)$ as shown in Equation 2. And the adjacency matrix A_p of spatial pattern graph G_p is similar to A_n.

In urban space, areas with similar functional areas usually have similar demand patterns, even if the areas are not close. Therefore, we build spatial function graph G_f to represent the similarity of POI features, the edge is $w_c\left(v_i,v_j\right)$ as shown in Equation 3. And the adjacency matrix A_f of spatial function graph G_f is also similar to A_n.

Based on multiple spatiotemporal graph structures, we use node values and edge weights to model the spatiotemporal dynamics of bike demand. Although graph neural networks perform well in non-Euclidean domains, it is difficult to do convolution operations on the graph. To overcome

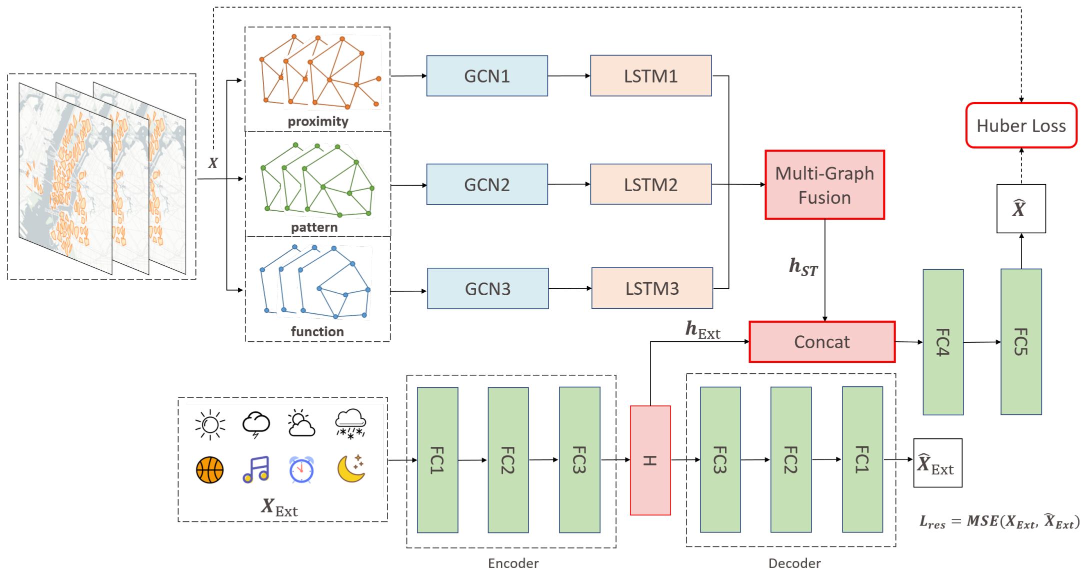


Fig. 5: Model architecture for our context-aware deep neural network named BikeNet, capturing the spatiotemporal contextual dependencies by a series of Graph Convolutional Network (GCN), Long Short-Term Memory (LSTM) and AutoEncoder (AE) layers.

this problem, Bruna et al. [24] proposed spectral graph convolution on graphs leveraging graph spectrum theory. With graph spectrum theory, the convolution operations on the graph can be easily defined in the Fourier domain. The spectral graph convolution "$*\mathcal{G}$" is defined as

$$\Theta * \mathcal{G}\, x = \Theta(L)x = \Theta(U\Lambda U^T)x \tag{8}$$

where U is graph Fourier basis means the eigenvectors of graph Laplacian matrix $L = D - A$, $D = diag(d_1, d_2, ..., d_n)$ is the diagonal degree and A is adjacency matrix; Λ is the diagonal matrix of eigenvalues of graph Laplacian matrix L, x is the graph signal and Θ is the kernel. To apply the graph neural network models in multi-graph modeling, we use *multi-graph fusion* designed by Xu et al. [25].

$$\boldsymbol{X}_{l+1} = \sigma\left(\bigcup_{\mathbf{A}\in\mathbb{A}} f\left(\mathbf{A};\theta_i\right)\boldsymbol{X}_l\mathbf{W}_l\right) \tag{9}$$

Where $\boldsymbol{X}_l$ and $\boldsymbol{X}_{l+1}$ represent the features of the i_{th} layer and the $(i+1)_{th}$ layer, respectively. σ represents the activation function, and $\bigcup$ represents the aggregation function. In this paper, we use the summation aggregation. $\mathbf{W}_l$ represents the feature transformation matrix, and $\mathbb{A}$ is the set of multiple graphs, that is the spatial proximity graph A_n, the spatial pattern graph A_p, and the spatial function graph A_f constructed in the previous section. $f(\mathbf{A};\theta_i)$ is the K-order polynomial function of the Laplacian matrix L of the corresponding graph.

4.2 RNN-based Temporal Dependencies Modeling

To capture the temporal dynamics of each bike station in urban space, we introduce an improved variant of RNN called Long Short-Term Memory (LSTM) [22]. In the time series forecast, the memory unit is commonly used to capture patterns over the long short-term with relatively little expense. We replace the matrix multiplications in LSTM with the graph convolution defined in Equation 8, which lead to our GNN-based LSTM as

$$f_t = \sigma\left(\Theta_f *_{\mathcal{G}} \cdot [h_{t-1}, x_t] + b_f\right) \tag{10}$$
$$i_t = \sigma\left(\Theta_i *_{\mathcal{G}} \cdot [h_{t-1}, x_t] + b_i\right) \tag{11}$$
$$o_t = \sigma\left(\Theta_o *_{\mathcal{G}} \cdot [h_{t-1}, x_t] + b_o\right) \tag{12}$$
$$h_{ST} = o_t \cdot tanh\left(C_t\right) \tag{13}$$

where $x(t)$, $h(t-1)$ denote the input and output of at time t and time $t-1$, $f(t)$, $i(t)$, and $o(t)$ are forget gate, input gate, and output gate at time t, respectively. $*\mathcal{G}$ denotes the convolution defined in Equation 8 and Θ_f, Θ_i, Θ_C, Θ_o are parameters for the corresponding filters. h_{ST} is the feature of spatial modeling and temporal modeling.

4.3 AE-based Contextual Dependencies Modeling

In order to model weather and social events and other external contextual features, we use AutoEncoder (AE) to achieve. We input various external features which have an impact on bike demand into AE, then it uses backpropagation algorithm to learn the potential representation of these features. Specifically, for the training samples X_{Ext}, the autoencoder first encodes the input to obtain the feature representation of the hidden layer h_{Ext}, and hopes that the original input can be reconstructed from the new features. The coding process is

$$h_{Ext} = f\left(W_1 X_{Ext} + b_1\right) \tag{14}$$

Then, the input X_{Ext} is reconstructed by using the feature h_{Ext}, that is, the decoding process is

$$\hat{X}_{Ext} = g\left(W_2 h_{Ext} + b_2\right) \tag{15}$$

where W_1 and W_2 are the weight matrix, b_1 and b_2 are the bias values.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3145979, IEEE Transactions on Mobile Computing

4.4 BikeNet: Model Architecture

With spatial correlation, temporal dependency, and contextual modeling, we build a context-aware deep neural network named BikeNet to accurately forecast the fine-grained bike demand trends. Fig. 5 shows the model architecture of BikeNet. Firstly, three graphs, namely spatial proximity graph, spatial pattern graph, and spatial function graph, are constructed to model a variety of spatial relationships. Secondly, three GCNs are used to extract the spatial features of three graphs, and LSTM is used to model their temporal features. Then, the spatial features extracted by GCN and the temporal features extracted by LSTM are fused to obtain the spatiotemporal feature representation h_{ST}. Finally, the context feature representation h_{Ext} obtained from self encoder modeling is fused with h_{ST}, after that the model is trained by using Huber loss [26] as the loss function after two fully connected layers (FC).

5 Demand Hotspot Rebalancing Optimization

After accurately forecasting bike demand trends, our next step is to find an optimal station rebalancing strategy so as to prevent stations from over-demand. However, one of the key challenges in solving the station rebalancing problem is that, due to the tremendous station-to-station combinations for moving bikes, it is usually an NP-hard problem [13]. Besides, this problem has resource constraints due to limited rebalancing resources (e.g., budgets and truck numbers) [27]. In this work, we exploit the integer linear programming (ILP) and Temporal-Difference (TD) algorithm to effectively find the optimal solution to this problem. First, we model the rebalancing task generation problem as the transportation problem, and then we propose an integer linear programming-based solution. Secondly, we model the task allocation problem as the Markov Decision Process (MDP). Then, the temporal difference algorithm is used to evaluate the policy and update the state value function. We elaborate the details as follows.

5.1 Bike Rebalancing Task Generation

5.1.1 System Modeling

We model the problem mentioned above as an optimization problem, and the objective is to minimize the cost of moving bikes from the over-supply stations to over-demand stations. We define the reasonable number of bikes in a station i at time t is in range $[lb, ub]$, where $lb = min \times T_i^{(t)}$, $ub = max \times T_i^{(t)}$, and min is minimum ratio of dock number, max is maximum ratio of dock number, $T_i^{(t)}$ is the number of total docks in station i at time t. Therefore, the station can satisfy those who want to rent bikes as well as return bikes. For each over-supply station i, the number of bikes that should be removed is in range $[s_i - ub, s_i - lb]$, and for each over-demand station j, the number of bikes that should receive is in range $[lb - d_j, ub - d_j]$, where $s_i(i = 1, 2, .., n)$ is the number of bikes in over-supply station i, $d_j(j = 1, 2, ..., m)$ is the number of bikes in over-demand station j, n and m are the number of over-supply stations and over-demand stations, respectively.

5.1.2 Rebalancing Demand Matrix Construction

We denote the initial number of bikes in over-supply station i as $s_i(s = 1, ..., n)$, and the initial number of bikes in over-demand station j as $d_j(j = 1, ..., m)$. Therefore, we define the over-supply stations and the over-demand stations as $S = \{s_1, s_2, \dots, s_n\}$ and $D = \{d_1, d_2, \dots, d_m\}$, respectively, where n is the number of over-supply stations of the bike-sharing systems and m is the number of over-demand stations. Based on the above definitions, we construct the cost matrix C, i.e.,

$$C = \begin{array}{c c} & \begin{array}{cccc} 1 & 2 & \dots & m \end{array} \\ \begin{array}{c} 1 \\ 2 \\ \dots \\ n \end{array} & \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1m} \\ c_{21} & c_{22} & \dots & c_{2m} \\ \dots & \dots & \dots & \dots \\ c_{n1} & c_{n2} & \dots & c_{nm} \end{pmatrix} \end{array}$$

where $c_{ij}(i = 1, 2..., n, j = 1, 2, ...m)$ is the route distance between over-supply station i and over-demand station j. However, in some cases, the maximal overall supply is less than the minimal overall demand or the minimal overall supply is large than the maximal overall demand, which leads to an unbalanced transportation problem. To find the optimal solution in these cases, we add a virtual supply station or demand station to transfer the unbalanced transportation problem to the balanced transportation problem, note that the bike transportation related to virtual stations does not happen in reality. Therefore, in the aforementioned cases, there must be some unbalanced stations after rebalancing. To make those stations closest to their balanced states, we should make the number of bikes offered or received by the virtual station minimal, thus the transportation cost related to the virtual station should be a number much larger than any other cost between real stations. More specifically, if $\sum_{i=1}^{n}(s_i - xT_i^{(t)}) < \sum_{j=1}^{m}(xT_j^{(t)} - d_j)$, we add a virtual supply station to the cost matrix C, i.e.,

$$C = \begin{array}{c c} & \begin{array}{cccc} 1 & 2 & \dots & m \end{array} \\ \begin{array}{c} 1 \\ 2 \\ \dots \\ n \\ n+1 \end{array} & \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1m} \\ c_{21} & c_{22} & \dots & c_{2m} \\ \dots & \dots & \dots & \dots \\ c_{n1} & c_{n2} & \dots & c_{nm} \\ c_{large} & c_{large} & \dots & c_{large} \end{pmatrix} \end{array}$$

where c_{large} is a number much larger than $c_{ij}(\forall i = 1, 2, ..n, j = 1, 2, .., m)$.

Similarly, if $\sum_{i=1}^{n}(s_i - yT_i^{(t)}) > \sum_{j=1}^{m}(yT_j^{(t)} - d_j)$, we add a virtual demand station to the cost matrix C, i.e.,

$$C = \begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & \dots & m & m+1 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ \dots \\ n \end{array} & \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1m} & c_{large} \\ c_{21} & c_{22} & \dots & c_{2m} & c_{large} \\ \dots & \dots & \dots & \dots & c_{large} \\ c_{n1} & c_{n2} & \dots & c_{nm} & c_{large} \end{pmatrix} \end{array}$$

where c_{large} is a number much larger than $c_{ij}(\forall i = 1, 2, ..n, j = 1, 2, .., m)$.

5.1.3 Problem Formulation and Rebalancing Task Generation

Based on the above definitions, we present the formulation of the bike transportation problem. The objective of this problem is to minimize the transportation cost under the constraint that after rebalancing, the unbalanced stations should be balanced.

Problem: (Rebalancing Task Generation)

$$minimize \quad \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} \tag{16}$$

subject to

$$min \times T_j^{(t)} - d_j \leq \sum_{i=1}^{n} x_{ij} \leq max \times T_j^{(t)} - d_j \tag{17}$$

$$s_i - max \times T_i^{(t)} \leq \sum_{j=1}^{m} x_{ij} \leq s_i - min \times T_i^{(t)} \tag{18}$$

$$x_{ij}(i = 1, ..., n, j = 1, ..., m) \geq 0 \tag{19}$$

Where x_{ij} is the number of bikes to be rebalanced from the over-supply station i to the over-demand station j. Because the number of bikes is an integer, we use the integer linear programming method to find the optimal solution of this problem. In particular, the Integer Linear Programming Solver from the MATLAB Optimization Toolbox[1] are employed to find the optimal solution.

5.1.4 Participants Incentive Design

After task generation, we need to effectively incentivize potential users to participate in station rebalancing. However, current user-based approaches utilize a simple monetary incentive strategy to encourage users to rebalance bikes, resulting in ineffective rebalancing plans and high-cost problems. Therefore, we propose a flexible and economical participants incentive mechanism to effectively encourage potential users to participate in station rebalancing.

For a rebalancing task T_i, we define the basic reward for a ride scheduling R_i as follows:

$$R_i = w \cdot c_i \cdot \frac{n_i}{b_i + o_i} \tag{20}$$

Where c_i is the distance between the start and end of rebalancing, n_i is the number of bikes to be rebalanced, b_i is the number of cyclists in history, o_i is the number of historical taxi orders; w is the reward coefficient, its value in some specific environments is obtained by questionnaire survey (the details can be found in Appendix), while w is 1 in the basic situation.

Then, the task generation model is further optimized. In station rebalancing, our goal is to maximize the number of users participating in scheduling, that is, to maximize the execution of the overlay task. For each rebalancing task, the constraint is that the number of tasks to be performed should be less than the number of bikes to be scheduled, because a user can only ride one shared bike at a time. And the total incentive cost is less than the given budget C. So, the optimized problem is

$$\max imize \sum_{i=1}^{n} y_i \tag{21}$$

subject to

$$\sum_{i=1}^{n} R_i < C \tag{22}$$

$$0 \leq y_i \leq n_i \,(i = 1, 2, ..., k) \tag{23}$$

$$0 \leq R_i \leq c \;\; (i = 1, 2, ..., k) \tag{24}$$

where y_i represents the number of scheduling bikes required for task T_i, and c is the maximum reward for each bike dispatch. In this paper, we take $c = 5$ based on the results of a survey [8]. In our questionnaire, 93.3 % of the participants chose the expected reward within 5, so similar conclusions can be drawn.

The optimized scheduling problem can also be solved by integer linear programming, and the CVX tool in MATLAB optimization toolbox is used.

5.2 Bike Rebalancing Task Allocation

Through the rebalancing task generation in the previous section, we get the candidate rebalancing tasks. The next step is to assign tasks to users to perform cycling schedule. In the process of task allocation, we should not only consider the benefits of users participating in the scheduling at the current time but also consider the benefits of users participating in the next riding scheduling. Because the current task allocation will affect the position and revenue of the next user, the problem is a sequential decision-making problem, which is suitable to be solved by the reinforcement learning model.

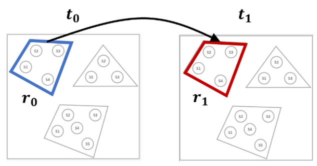
In this paper, we model the task allocation problem as the taxi order dispatch problem. The intelligent dispatch method proposed by [28] is used to allocate the rebalancing task. Specifically, the task allocation problem is modeled by the Markov Decision Process (MDP). Then, the Temporal-Difference (TD)algorithm is used to evaluate the policy and update the state value function. Finally, the overall user income in scheduling is maximized by the binary graph matching algorithm.

5.2.1 Markov Decision Process Definition

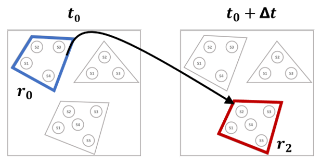
In this paper, the allocation of rebalancing tasks is modeled as a Markov Decision Process:

- **Agent**: We model the whole task allocation platform as an agent, which decides whether to assign rebalancing tasks to users.
- **State**: We define the user's state s as current spatiotemporal state. The spatial state is the current hotspot area r. The temporal state takes every 30 minutes as a state t, then the spatiotemporal state $s = (t, r)$.
- **Action**: There are two kinds of actions a, one is to assign tasks to users, the other is not to assign tasks to users.

[1]https://www.mathworks.com/help/optim/index.html



(a) The change of user state when no action is matched.



(b) The change of user state when matching actions.

Fig. 6: The change of user state.

- **State Transition and Reward**: After the user is assigned to the rebalancing task, he will ride the bike to the designated place, and his status will change, getting the corresponding reward R. However, if the user is not assigned to the rebalancing task, its status remains unchanged, and there is no reward.

After the definition of the Markov Decision Process is completed, the next step is to find the optimal allocation policy through policy evaluation to maximize the cumulative expected benefits of users participating in scheduling.

5.2.2 Temporal Difference Maximizes Expected Reward

In order to learn the optimal strategy, we use the state value function $V_\pi(s)$ to learn the state value based on the time series difference algorithm. First, define the target state values in the case of two kinds of actions. When the task is not assigned to the user, define the current state value $V(s)$ of the user as and the future state value as $V(s')$. Because the user does not participate in the scheduling, the reward is 0 and the target state value is:

$$V(s) \leftarrow V(s) + \alpha\left[0 + \gamma^{\Delta t} V(s') - V(s)\right] \tag{25}$$

where current state is $s = (t, r)$, the next state is $s' = (t+1, r)$, α is learning rate, γ is discount factor, Δt is the time to finish rebalancing task.

When a task is assigned to a user, the user's future state value is defined as $V(s'')$. Because of participating in the scheduling, the user's reward is R, and the target state value is:

$$V(s) \leftarrow V(s) + \alpha\left[R + \gamma^{\Delta t} V(s'') - V(s)\right] \tag{26}$$

where current state is $s = (t, r)$, the next state is $s'' = (t + \Delta t, r')$.

The state changes under the two actions are shown in Fig. 6. Set the user's current position as the area r_0 and the current period as t_0. When the action is not matched, because there is no riding scheduling, the user's next location area r_1 is the same as the original location, and the reward is 0. When the action is a match, the user will cycle from the region $r_0$to the end of the rebalancing task, and the next period $t_0 + \Delta t$ will be the region r_2, and gain reward R.

For a task allocation period, there are n rebalancing tasks to be assigned. The allocation of rebalancing tasks can be regarded as a binary graph matching problem. The long-term benefits of the whole can be optimized by setting the weight of the binary graph edge as the increment of the state value. Therefore, according to the current and long-term benefits, the rebalancing task allocation problem is modeled as the weighted binary graph maximum matching problem, which can be solved by Hungarian algorithm. The mathematical model is

$$\arg\max_{a_{ij}} \sum_{i=1}^{m}\sum_{j=1}^{n}\Big(R(i,j) + \gamma^{\Delta t} V\left(s'_{ij}\right) - V(s_i)\Big)a_{ij} \tag{27}$$

subject to

$$\sum_{i=1}^{m} a_{i,j} \leq 1,\ j = 1, 2, ..., n \tag{28}$$

$$\sum_{j=1}^{n} a_{i,j} \leq 1,\ i = 1, 2, ..., m \tag{29}$$

where m represents the number of users in task allocation, n represents the number of tasks that need to be scheduled, $R(i,j)$ represents the reward that user i can get by completing task j, $a_{i,j}$ represents whether user i and task j are matched.

6 Evaluation

In this section, we first introduce the experiment settings and then present the evaluation results on trend forecast and station rebalancing.

6.1 Experiment Settings

6.1.1 Dataset

To evaluate the performance of the proposed methods, two real-world datasets are used in experiments: the New York Citi Bike dataset and the Xiamen Mobike dataset. In addition, in order to model more accurately, we obtained a closely related contextual dataset including POI datasets, weather datasets, and taxi trajectory datasets for two cities.

TABLE 1: Dataset Description

Statistics	New York City	Xiamen City
Stations	782	2,659
Bike trips	62,219,223	231,730,001,453
Time span	2015.03-2019.04	2017.06-2017.10

The New York Citi Bike dataset[1] is a docked bike-sharing dataset. The data is collected from March 1, 2015 to April 30, 2019, including 1005 stations and 62,219,223 bickcle riding records. The Xiamen Mobike dataset is a dockless bike-sharing dataset. The data is collected from June 1, 2017 to October 31, 2017 and contains 231,730,001,453 riding records. The dataset includes start time, start position, stop time, stop position, and so on, the detail is presented in TABLE 1.

The POI data for New York City and Xiamen City are obtained from NYC Open Data[2] and Amap Search Service API[3], respectively. When obtaining POI data, query

[1]https://www.citibikenyc.com/system-data
[2]https://data.cityofnewyork.us/City-Government/Points-Of-Interest/rxuy-2muj
[3]https://developer.amap.com/api/webservice/guide/api/search

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3145979, IEEE Transactions on Mobile Computing

TABLE 2: Willingness Tuning Coefficient Matrix for adjusting the reward amount

Temperature	1.32 (hot)	0.96 (cold)
Weather	1.77 (bad)	0.52 (good)
Weekday	0.83 (rush hours)	0.68 (normal hours)
Holiday	0.61 (daytime)	0.66 (evening)
Social activity	0.77 (interested in)	0.85 (not interested in)
Traffic condition	1.05 (convenient)	0.63 (inconvenient)
Functional area	0.64 (urban village ∣ school ∣ scenic)	0.85 (business ∣ elderly community ∣ industrial)

the POI categories within 1km of the geographic location, perform statistics, and obtain the POI category information corresponding to each area. The weather data of New York City and Xiamen City are obtained from National Weather Service[1] and Amap Weather Query API[2], respectively. The weather data includes weather conditions, temperature, wind direction, wind force, air humidity. POI data and weather data can help us more accurately forecast bike demand trends. New York taxi trajectory data[3] and Xiamen taxi trajectory data mainly includes taxi ID, driving speed. driving direction, time information, latitude, longitude, altitude, and whether the taxi is transporting passengers. Taxi trajectory data can help us select potential participants to participate in station rebalancing and calculate rebalancing incentives.

6.1.2 Evaluation Metric

We compared the trend forecast with the ground truth dataset to evaluate the accuracy of the forecasting method. Specifically, we use two commonly used evaluation metrics in trend forecast, including (1) Root Mean Squared Error (RMSE), (2) Mean Absolute Error (MAE). They are defined as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{30}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{31}$$

In order to measure the performance of the station rebalancing method and users incentives, we use system utility, average reward, and average riding distance of users as evaluation metrics. The system utility represents the ratio of the number of balanced stations to the total number of stations. The larger the value of system utility, the more sites that can be rented and returned bikes normally, the better the user experience. We define N as the total number of bike stations, $UN^{(k)}(\mathcal{A})$ as the number of over-demand stations after algorithm $\mathcal{A}$ scheduling during the kth scheduling. We have described the detail of over-demand stations in 5. K is the num of total scheduling times, the average system utility of single scheduling is as follows:

$$Utility(\mathcal{A}) = \frac{\sum_{k=1}^{K} N - UN^{(k)}(\mathcal{A})}{NK} \times 100\% \tag{32}$$

The average reward means the average reward obtained by users after participate in rebalancing tasks. We define $R^{(k)}(\mathcal{A})$ is the total reward of all users after algorithm $\mathcal{A}$ scheduling during the kth scheduling. K is the num of total scheduling times, the average reward is as follows:

$$Reward(\mathcal{A}) = \frac{1}{K}\sum_{k=1}^{K} R^{(k)}(\mathcal{A}) \tag{33}$$

The average riding distance is defined as the ratio of the total riding distance of the user participate in rebalancing tasks to the num of riding times, and its unit is km. We define $D^{(k)}(\mathcal{A})$ is the total riding distance during the kth rebalancing, then the average riding distance of the user participating in the rebalancing is as follows:

$$Distance(\mathcal{A}) = \frac{1}{K}\sum_{k=1}^{K} D^{(k)}(\mathcal{A}) \tag{34}$$

6.1.3 Parameter Settings

In the hotspot demand forecast experiments, we select 80 % of the dataset for training each model, and the remaining 20 % is chosen as the validation set and testing set. Adam optimizer is chosen for training. The maximum epoch is set to 1,000, and the batch size is 64, the learning rate is 0.0001. We also use an early stopping mechanism to avoid overfitting, and the Early Stopping Length is set to 100. Early Stopping Patience is set to 0.1. Besides, the K-order polynomial parameter of GCN is set to 2, the number of GCN hidden layers (Hidden Layer Number) is set to 3, the number of LSTM units (Unit Number) is set to 32, and the number of fully connected layer units is set to 64. We repeat each experiment 100 times.

In the station rebalancing experiments, to dynamically adjust the reward amount according to different riding environments (including weather conditions, time periods, functional areas, etc.) to generate reasonable cycling tasks, we designed a questionnaire (the details can be found in Appendix) to investigate the willingness of cycling users to participate in various situations so as to determine reward coefficient w. We have recruited 30 users whose ages range from 18 to 48 years old (AVG=29.5) to participate in our survey. They are the main users who often use bikes to travel. Then we calculate the coefficient w as follows:

$$w = \frac{5 - mean\_score}{2.5} \tag{35}$$

Where 5 represents the perfect score for a question, and 2.5 represents the score under normal circumstances. $mean\_score$ is the average score of the questionnaire answer.

According to the results of the questionnaire survey, we can obtain a willingness tuning table (as shown in TABLE 2) for the influence of different contextual factors as the parameters of our ILP-TD rebalancing strategy.

[1]https://w2.weather.gov/climate/xmacis.php?wfo=okx
[2]https://lbs.amap.com/api/webservice/guide/api/weatherinfo/
[3]https://data.cityofnewyork.us/Transportation/2018-Yellow-Taxi-Trip-Data/t29m-gskq

TABLE 3: Trend Forecast Evaluation Results on Citi Bike and Mobike.

Dataset		Citi Bike				Mobike			
Methods		slope		duration		slope		duration	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Station -level	ARIMA	11.881	15.756	16.035	23.691	3.973	5.633	4.866	8.917
	XGBoost	11.058	14.942	14.893	21.476	3.615	4.923	4.705	8.636
	GBRT	11.186	15.205	14.608	21.266	3.562	5.235	4.703	8.631
	DCRNN	9.979	13.922	14.284	22.169	2.572	5.318	2.292	5.562
	ST-ResNet	15.812	25.843	13.238	19.226	2.527	4.228	1.565	3.013
	STGCN	9.001	12.845	12.617	19.559	1.874	3.959	1.638	3.097
	STMGCN	8.941	12.693	12.634	19.438	1.442	2.897	2.349	6.121
Trend -level	DCRNN	4.575	6.597	2.019	2.726	0.887	2.186	1.709	4.187
	ST-ResNet	8.158	17.865	4.325	12.571	1.503	4.551	1.982	4.546
	STGCN	4.452	6.372	1.947	2.696	0.849	2.122	1.552	3.928
	STMGCN	4.377	6.487	1.914	2.678	0.793	1.985	1.532	3.733
	BikeNet	**4.171**	**5.988**	**1.902**	**2.616**	**0.779**	**1.883**	**1.389**	**3.504**

TABLE 4: The Evaluation Results of Ablation Study Experiments (Using RMSE)

Method	Citi Bike		Mobike	
	slope	duration	slope	duration
Only proximity	6.3123	2.6467	3.6456	1.9645
Only pattern	6.2295	2.6861	3.6514	1.9541
Only function	6.1932	2.6726	3.6319	1.9447
Removed context	6.1811	2.6234	3.6623	1.9654
BikeNet	**5.9880**	**2.6168**	**3.5044**	**1.8830**

6.1.4 Baseline Methods

We compared our method with various baseline methods with regard to trend forecast and station rebalancing. For trend forecast, we compared our BikeNet with the following baselines:

- **ARIMA**: Auto-regressive integrated moving average (ARIMA) [18] model has a simple structure, few input parameters and good results. It is the most widely used method for univariate time series forecasting.
- **XGBoost**: Extreme Gradient Boosting (XGBoost) [29] is a more efficient implementation of Gradient Boosting. It can improve the efficiency of algorithm operation based on CPU multi-threaded parallel processing, and improve the model to improve accuracy.
- **GBRT**: Gradient Boosting Regression Tree (GBRT) [30] is a type of Boosting ensemble learning. It is composed of multiple decision trees. Its core idea is to accumulate the results of all trees as the final output result.
- **DCRNN**: Diffusion Convolutional Recurrent Neural Network (DCRNN) [31] is a deep learning method used for traffic flow forecast. It uses a two-way random walk to model spatial relationships and an Encoder-Decoder structure to model temporal dependence.
- **ST-ResNet**: Spatio-Temporal Residual Networks(ST-ResNet) [20] is deep-learning-based approach for crowd flows prediction, integrating convolution neural network and residual neural network through residul convolutional nunits. In ST-ResNet, convolution neural network and residual neural network are used to respectively model spatial dependency and temporal dynamics. And it combined with external factors (e.g., weather) to improve prediction performance.
- **STGCN**: Spatio-Temporal Graph Convolutional Network (STGCN) [32] is a deep learning framework for traffic forecasting, integrating graph convolution and gated temporal convolution through spatio-temporal convolutional blocks. In STGCN, graph convolutional layers and convolutional sequence learning layers are combined to model spatial and temporal dependencies.
- **STMGCN**: Spatiotemporal Multi-Graph Convolution Network (STMGCN) [25] fuses a variety of non-Euclidean spatial correlation features through multi-graph convolution, and then introduces global context information to model the temporal dependency, and weights the features at different moments.

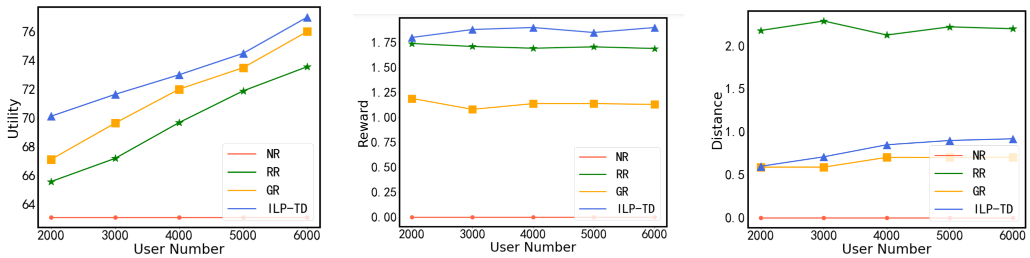
For station rebalancing, we use the following three baseline station rebalancing methods [33] for comparative studies:

- **No Rebalancing (NR)**: Do not perform any rebalancing operations on the bike-sharing system. Calculate the average utility of the bike-sharing system based on the supply and demand of each stations in each period as a benchmark for comparison. Since the user does not participate in the rebalancing, the user's average reward and riding distance are always equal to zero.
- **Random Rebalancing (RR)**: According to the trend forecasting, we divide the stations into over-supply areas and over-demand areas. The over-supply area is taken as the starting point of rebalancing, and one over-demand area is randomly selected as the ending point of rebalancing. Since the user's riding distance is up to 10km, when the randomly assigned rebalancing task distance exceeds 10km, it

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3145979, IEEE Transactions on Mobile Computing

TABLE 5: Station Rebalancing Evaluation Results

Method	Citi Bike			Mobike		
	Utility	Reward	Distance	Utility	Reward	Distance
NR	63.235%	0	0	63.105%	0	0
RR	73.192%	1.382	2.186	67.273%	1.715	2.293
GR	76.686%	0.830	1.354	69.650%	1.083	0.597
MPC	78.025%	1.532	1.386	70.892%	1.956	0.645
ILP-TD	**80.422%**	**1.404**	**1.266**	**71.658%**	**1.881**	**0.714**



(a) Impact on Utility. (b) Impact on Reward. (c) Impact on Distance.

Fig. 7: The influence of the number of users on mobike dataset in the experiments.

is considered that the user will not participate in the rebalancing. Finally, we repeat 10 times random rebalancing and calculate the average system utility, average user reward, and riding distance under the random rebalancing method.

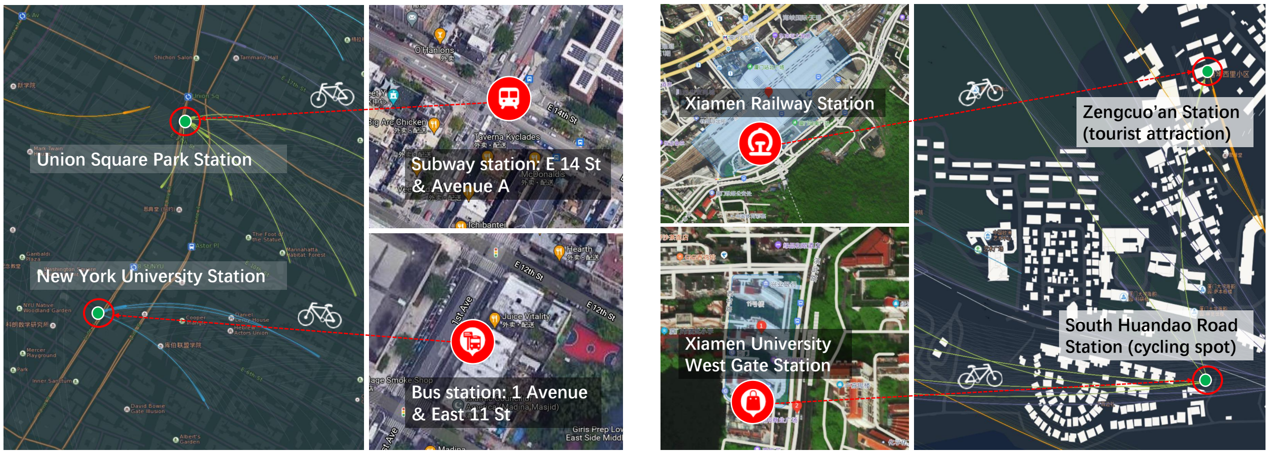
- **Greedy Rebalancing (GR)**: Similarly, according to the trend forecasting, we divide the stations into over-supply areas and over-demand areas. If the area where the user is located is an over-supply area, it can be used as the starting point for rebalancing. Then, select an area with the shortest walking distance and the shortest riding distance from the over-demand area as the ending point for rebalancing. If the greedy method assigned rebalancing task is more than 10km away, the user will not participate in this rebalancing. Then, calculate the results of each evaluation metrics under multiple rounds of greedy rebalancing.
- **Model Predictive Control (MPC)**: Pfrommer et al. [34] proposed an incentives mechanism based on model predictive control (MPC) [35], which dynamically chooses payments offered to customers to change the endpoint of their journey to a nearby station in a way that improves the overall service level. To formulate a tractable MPC problem, they approximate the behavior of customers participating in rebalancing is linearly related to the prices incentives. The state of the system in each time period of the horizon is defined as the number of bicycles present at each station. The control inputs are the price incentives offered at each station for diversions to neighboring stations. The objective function defines a tradeoff between the quality of the achieved system state and the cost for paying out incentives. The solution gives the prices incentives to be offered to customers.

6.2 Evaluation Results

6.2.1 Trend Forecast Results

TABLE 3 shows the trend forecast results on the New York Citi Bike dataset and Xiamen Mobike dataset, respectively. The prediction results under Trend-level are significantly better than those under ordinary Station-level. For the Citi Bike dataset, the results show that the RMSE and MAE of our BikeNet are the least in the slope forecasting task, respectively, 5.988 and 4.171. At the same time, BikeNet also obtained the best forecast results in the duration forecasting task, RMSE and MAE are 0.617 and 1.902 respectively. For the Mobike dataset, the results show that the RMSE and MAE of our BikeNet are the least in the slope forecasting task, respectively, 1.883 and 0.779. At the same time, BikeNet also obtained the best forecast results in the duration forecasting task, RMSE and MAE are 3.504 and 1.389 respectively. Experiments on New York City and Xiamen City show that BikeNet has good generalization in different spatiotemporal regions. In conclusion, our proposed model can capture the spatiotemporal contextual dependencies effectively.

To further verify the effectiveness of the proposed forecast model, ablation study experiments are carried out to analyze the impact of multi-graph construction and contextual feature modeling on the forecast results. Firstly, the influence of multi-graph is verified. Compared with the complete BikeNet method, the module of multi-graph construction is removed. Only spatial proximity graph, only spatial pattern graph, and only spatial function graph are established instead. The differences between the single graph model and the multi-graph model are analyzed. Then, the influence of contextual factors is verified by comparing with the model excluding contextual features. As shown in TABLE 4, the removal of any component in our model would cause an error increase.



(a) The station rebalancing cases near Union Square Park and New York University during the morning rush hour on April 3, 2019.

(b) The station rebalancing cases near Zengcuo'an West Road and South Huandao Road in Xiamen City at 8 am on October 1, 2017.

Fig. 8: Two station rebalancing cases in New York City and Xiamen City

6.2.2 *Station Rebalancing Results*

We extract the riding record of a moment from the bike trajectory data as the user's current state. From the current time to the evening, the scheduling task is generated and allocated every 30 minutes, and a total of 24 scheduling tasks are carried out. Then, the Utility, Reward, and Distance under multi-round scheduling simulations are calculated as the final experimental results of shared bike scheduling optimization. As shown in TABLE 5, the ILP-TD method proposed in this paper achieves the optimal rebalancing effect and maximum user benefit under Utility and Reward respectively. On the New York Citi Bike dataset, the system utility improves from 63.2% to 80.4%, achieving 17% of improvement. On the Xiamen Mobike dataset, the system utility improves from 63.1% to 71.7%, achieving 8.5% of improvement.

In addition, we analyze the influence of different users on the scheduling experiment results on Mobike dataset. The number of users participating in the scheduling ranges from 2000 to 6000, and the interval difference is 1000. The results of the influence of the number of users on the experiment are shown in Fig. 7 . The results show that the average availability of the system is directly proportional to the number of users under RR, GR and ILP-TD methods, but there is no obvious correlation between the average revenue of users and the average riding distance of users and the number of users. As for the MPC method, the number of users is derived from the historical average, and the scheduling results are only affected by incentives, so we don't need to add comparisons here.

6.3 Case Studies

We conducted case studies of our station rebalancing strategy in New York and Xiamen city, respectively.

6.3.1 *Station Rebalancing Cases in New York City*

As a world-class metropolis, New York has very complete traffic conditions, but citizens will also face traffic jams in the morning and evening rush hours due to complex travel conditions. Nowadays, quite a lot of people are willing to choose shared bike travel when traveling short distances, because of its low carbon and convenience. Therefore, rebalancing at bike demand hotspots in the morning and evening rush hours is particularly important. We conducted station rebalancing at bike demand hotspots in the morning rush hours at 9:30 on April 3, 2019. As Fig. 8(a) shows, there are two demand hotspots (green points) near Union Square Park and New York University. Our method can identify these hotspots and generate realistic cycling tasks to dispatch bikes from the subway station near E 14 St & Avenue A and the bus station near 1 Avenue & East 11 St with many potential participants and enough bikes, assisting citizens to work and students to class. Compared with the general situation of No Rebalancing (NR), using our strategy can increase the average system availability (Utility) by 17.187 % in several simulation experiments.

6.3.2 *Station Rebalancing Cases in Xiamen City*

Xiamen is one of the most famous tourist cities in China. There are a large number of tourists during the holidays. It is quite pleasant for tourists to be able to enjoy such beautiful scenery during the short and precious holidays. Besides, local citizens are also willing to choose shared bikes (e.g., Mobike) when traveling for short distances and there are many non-pile bike stations on Xiamen City. We conducted station rebalancing at bike demand hotspots during the National Day holiday of China at 8 a.m. on October 1, 2017. As shown in Fig. 8(b), there are two demand hotspots (green points) near South Huandao Road (one of the tourists' favorite cycling starting point) and Zengcuo'an Village (a tourist attraction area). Our method can identify these hotspots and generate realistic cycling tasks to dispatch bikes from the west gate of of Xiamen University and Xiamen Railway Station with many potential participants and enough bikes, so that tourists can get a pleasant travel experience. In several simulation experiments, using our strategy can increase the average system availability (Utility) by 8.553 % compared with the general situation of No Rebalancing (NR).

The above cases have verified the versatility and effectiveness of our station rebalancing strategy, which can not

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3145979, IEEE Transactions on Mobile Computing

only motivates participants and efficiently utilizes transportation resources, but also improves the efficiency of the bike-sharing system and contributes to low carbon.

7 Related Work

We describe the related work from two perspectives, i.e., spatiotemporal deep learning, and station rebalancing for bike-sharing system.

7.1 Spatiotemporal Deep Learning

Spatiotemporal sequence forecast is mainly to model spatial correlation and temporal dependency. Nowadays, machine-learning-based methods perform well in spatiotemporal sequence forecast problems, which have special designs for capturing the spatiotemporal features [36]. It can be divided into classical methods and deep learning methods. Classical feature-based methods are traditional machine learning models, such as Auto-Regressive Integrated Moving Average (ARIMA) [18], Support Vector Regression (SVR) [37] and GP [38]. For deep learning methods, Srivastava et al. [39] proposed to use fully connected Long Short-term Memory (LSTM) units to forecast spatiotemporal sequence, but only considered temporal feature and ignored spatial dependency. Shi et al. [40] proposed the Convolutional LSTM (ConvLSTM) to capture spatial correlation with FC-LSTM's architecture, which added the consideration of spatial correlation. Zhang et al. [20] proposed a grid-based method to divide the city and proposed Deep Spatio-Temporal Residual Networks (ST-ResNet) to represent the long-term patterns.

Recently, the graph neural networks perform well in non-Euclidean domains, which employ graph structure to model the spatial characteristic. Specifically, the Graph Convolutional Network (GCN) is the pioneering work in graph neural networks, Bruna et al. [24] first introduce spectral graph convolution by graph spectrum theory. The graph convolutional neural network architecture designed by Kipf et al. [21] is extensively used. Sometimes it combines GCN with RNN to model both spatial correlation and temporal dependency. Yu et al. [32] proposed Spatio-Temporal Graph Convolutional Networks (STGCN) for traffic forecast, introducing graph convolution and gated temporal convolution through a designed block. And Li et al. [31] proposed Diffusion Convolutional Recurrent Neural Network (DCRNN) for long-term forecast.

Wang et al. [41] propose a spatiotemporal graph neural network (ST-GNN) to accurately forecast the hospital visit demand which can simultaneously captures the spatial correlation by graph convolutional networks (GCN) and the temporal dependency by gated recurrent units (GRU). In bike demand forecasting, Lin et al. [42] proposed a novel GCNN-DDGF model for station-level hourly demand forecast in a large-scale bike-sharing network, which can automatically capture heterogeneous pairwise correlations between stations to improve forecasting. Zi et al. [43] proposed a novel deep graph convolutional network (GCN) model with temporal attention (TAGCN), which can capture the dynamical temporal correlations and comprehensive spatial patterns in bike check-out/in flow effectively. However, these methods still didn't consider various context features. Guo et al. [44] built a spatiotemporal graph neural network to accurately forecast city-wide bike demands. Based on the previous work, we use spatial region clustering, temporal trend extraction and context features fusion to improve the accuracy of bike demand forecasting, so as to complete a more realistic station rebalancing task.

7.2 Station Rebalancing for Bike Sharing System

Bike-sharing systems are booming globally as a green transportation, the key challenge is avoiding bike stations being over-demand. There are two main rebalancing methods, the vehicle-based method and the user-based method [7]. Vehicle-based methods utilize multiple trucks to redistribute bikes among stations, which can be divided into two categories, including static truck-based rebalancing approach and dynamic truck-based rebalancing approach. Static truck-based rebalancing methods mean that the bikes on the bike stations are rebalanced when they are not working or at midnight. Liu et al. [45] use an optimization model to get the minimum transportation cost under distance constraints. Dynamic truck-based rebalancing calculates the dynamic bike repositioning strategies based on real-time route planning, which is an online and real-time rebalancing method. Lowalekar et al. [46] propose a multi-step dynamic representation method to model future demand and design the best rebalancing strategy in different time spans. In recent years, the emergence of dockless bike-sharing systems has provided new opportunities for researchers to propose user-based rebalancing methods. User-based rebalancing methods are widely used, which encourage users to help rebalance the over-demand bike stations by monetary incentives. Singla et al. [8] introduce a crowdsensing mechanism to reward customers by rewarding them money, thereby they will help reposition the bike at over-demand bike stations.

What's more, Duan et al. [47] proposed a rebalancing scheme by recruiting workers. There are also lots of researches that take dynamic schemas into account which can help forecast future demand, Kloimüllner et al. [48] showed how to extend the metaheuristics developed in previous work for static bike-sharing systems to the significantly more complex dynamic variant. Chiariotti et al. [49] proposed a general framework for the dynamic rebalancing of a bike-sharing system, in order to improve the availability of the service. Brinkmannet et al. [50] presented a dynamic lookahead policy (DLA) to anticipate potential future demands in the current inventory decisions and avoid unsatisfied demand by dynamically relocating bikes. But they did not consider the uneven time distribution of bike demand which is significantly affected by various contexts (such as weather, holidays, social events, etc.). Their balancing strategies also did not comprehensively apply various available resources.

Furthermore, there are some researches that consider several available resources. [14], [34] showed that the bike-sharing system can use customer incentive methods to help balance the number of bikes to reduce operating costs. Specifically, Chiariotti et al. [51] proposed an integrated model combining incentives to users and traditional truck rebalancing for the optimization of bike-sharing systems which can obtain a higher service quality for a lower cost than using simple rebalancing. However, they ignored the realistic factors such as weather and holidays that may significantly affect the use of the bike-sharing system in the prediction of future demand. In terms of incentive strategies, Chung et al. [52] have proposed a number of data-driven policies to guide incentives for rebalancing in bike-sharing systems via crowdsourcing and displayed the performance differences more comprehensively. But their analysis is more based on statistical methods, failing to model various complex real-life conditions, and they have not considered more available vehicles in practice. To improve the existing work, our framework comprehensively models various situational factors and considers multiple types of transportation resources. In addition, we also designed a questionnaire to investigate the willingness of participants to perform station rebalancing.

8 Conclusion

In this paper, we propose RedPacketBike, a crowd-driven station rebalancing mechanism to accurately forecast the fine-grained bike demand trends and effectively rebalance over-demand bike stations leveraging deep learning and crowdsensing techniques. In order to model bike demand hotspots, we propose a spatiotemporal bike demand hotspot clustering method, realizing the spatial and temporal modeling of bike demand hotspots. In order to accurately forecast bike demands in a future period of time, we build a context-aware deep neural network named BikeNet to model and forecast the fine-grained bike demand trends. In order to find an optimal station rebalancing strategy, we exploit the integer linear programming (ILP) and Temporal-Difference (TD) algorithm to effectively find the optimal solution to this problem. Real-world datasets validate the effectiveness of our proposed mechanism.

In the future, we intend to improve this work from the following aspects. First, we plan to model the impact of more social events in predicting bicycle demand to achieve higher accuracy. Second, we plan to build an end-to-end deep learning framework by combining the spatiotemporal forecast model and the deep reinforcement learning rebalancing model. Third, we plan to deploy our framework to urban bike-sharing systems to provide real-world services.

References

[1] C. Bikeshare, "Capital bikeshare member survey report," *Washington, DC*, 2013.
[2] D. Chai, L. Wang, and Q. Yang, "Bike flow prediction with multi-graph convolutional networks," in *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2018, pp. 397–400.
[3] C. M. de Chardon, G. Caruso, and I. Thomas, "Bike-share rebalancing strategies, patterns, and purpose," *Journal of transport geography*, vol. 55, pp. 22–39, 2016.
[4] S. M. Kaufman, L. Gordon-Koven, N. Levenson, and M. L. Moss, "Citi bike: the first two years," *The Rudin Center for Transportation Policy and Management*, 2015.
[5] L. Chen, D. Zhang, G. Pan, X. Ma, D. Yang, K. Kushlev, W. Zhang, and S. Li, "Bike sharing station placement leveraging heterogeneous urban open data," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2015, pp. 571–575.
[6] C. Contardo, C. Morency, and L.-M. Rousseau, *Balancing a dynamic public bike-sharing system*. Cirrelt Montreal, Canada, 2012, vol. 4.
[7] L. Pan, Q. Cai, Z. Fang, P. Tang, and L. Huang, "A deep reinforcement learning framework for rebalancing dockless bike sharing systems," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 1393–1400.
[8] A. Singla, M. Santoni, G. Bartók, P. Mukerji, M. Meenen, and A. Krause, "Incentivizing users for balancing bike sharing systems," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.
[9] C. Chen, Y. Ding, Z. Wang, J. Zhao, B. Guo, and D. Zhang, "Vtracer: When online vehicle trajectory compression meets mobile edge computing," *IEEE Systems Journal*, 2019.
[10] S. Guo, C. Chen, J. Wang, Y. Ding, Y. Liu, X. Ke, Z. Yu, and D. Zhang, "A force-directed approach to seeking route recommendation in ride-on-demand service using multi-source urban data," *IEEE Transactions on Mobile Computing*, 2020.
[11] H. Chen, B. Guo, Z. Yu, and Q. Han, "Crowdtracking: Real-time vehicle tracking through mobile crowdsensing," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7570–7583, 2019.
[12] L. Chen, D. Zhang, L. Wang, D. Yang, X. Ma, S. Li, Z. Wu, G. Pan, T.-M.-T. Nguyen, and J. Jakubowicz, "Dynamic cluster-based over-demand prediction in bike sharing systems," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 841–852.
[13] R. M. Karp *et al.*, "Complexity of computer computations," *Reducibility among combinatorial problems*, vol. 23, no. 1, pp. 85–103, 1972.
[14] C. Fricker and N. Gast, "Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity," *Euro journal on transportation and logistics*, vol. 5, no. 3, pp. 261–291, 2016.
[15] P. Wang, H. Wang, and W. Wang, "Finding semantics in time series," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, 2011, pp. 385–396.
[16] S. McKinley and M. Levine, "Cubic spline interpolation," *College of the Redwoods*, vol. 45, no. 1, pp. 1049–1060, 1998.
[17] F. Scholkmann, J. Boss, and M. Wolf, "An efficient algorithm for automatic peak detection in noisy periodic and quasi-periodic signals," *Algorithms*, vol. 5, no. 4, pp. 588–603, 2012.
[18] A. Kaltenbrunner, R. Meza, J. Grivolla, J. Codina, and R. Banchs, "Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system," *Pervasive and Mobile Computing*, vol. 6, no. 4, pp. 455–466, 2010.
[19] T. L. Fine, *Feedforward Neural Network Methodology*. Springer Science & Business Media, 2006.
[20] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
[21] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.
[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[23] H. J. Miller, "Tobler's first law and spatial analysis," *Annals of the Association of American Geographers*, vol. 94, no. 2, pp. 284–289, 2004.
[24] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," *Computer Science*, 2013.
[25] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 3656–3663.
[26] P. J. Huber, "Robust estimation of a location parameter," in *Breakthroughs in statistics*. Springer, 1992, pp. 492–518.
[27] C. Chen, S. Jiao, S. Zhang, W. Liu, L. Feng, and Y. Wang, "Tripimputor: real-time imputing taxi trip purpose leveraging multi-sourced urban data," *IEEE Transactions on Intelligent Transportation Systems*, no. 99, pp. 1–13, 2018.
[28] Z. Xu, Z. Li, Q. Guan, D. Zhang, Q. Li, J. Nan, C. Liu, W. Bian, and J. Ye, "Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 905–913.
[29] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[30] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.
[31] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *International Conference on Learning Representations*, 2018.
[32] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 2018, pp. 3634–3640.
[33] J. Wang, Y. Wang, D. Zhang, F. Wang, Y. He, and L. Ma, "Psallocator: Multi-task allocation for participatory sensing with sensing capability constraints," in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, 2017, pp. 1139–1151.
[34] J. Pfrommer, J. Warrington, G. Schildbach, and M. Morari, "Dynamic vehicle redistribution and online price incentives in shared mobility systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1567–1578, 2014.
[35] J. M. Maciejowski, *Predictive control: with constraints*. Pearson education, 2002.
[36] X. Shi and D.-Y. Yeung, "Machine Learning for Spatiotemporal Sequence Forecasting: A Survey," *arXiv preprint arXiv:1808.06865*, 2018.
[37] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.
[38] R. Senanayake, S. O'Callaghan, and F. Ramos, "Predicting spatio-temporal propagation of seasonal influenza using variational Gaussian process regression," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
[39] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *International Conference on Machine Learning*, 2015, pp. 843–852.
[40] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems*, 2015, pp. 802–810.
[41] Z. Wang, R. Guo, L. Hong, C. Wang, and L. Chen, "Demand-responsive windows scheduling in tertiary hospital leveraging spatiotemporal neural networks," in *International Conference on Green, Pervasive, and Cloud Computing*. Springer, 2020, pp. 231–243.
[42] L. Lin, Z. He, and S. Peeta, "Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach," *Transportation Research Part C: Emerging Technologies*, vol. 97, pp. 258–276, 2018.
[43] W. Zi, W. Xiong, H. Chen, and L. Chen, "Tagcn: Station-level demand prediction for bike-sharing system via a temporal attention graph convolution network," *Information Sciences*, vol. 561, pp. 274–285, 2021.
[44] R. Guo, Z. Jiang, J. Huang, J. Tao, C. Wang, J. Li, and L. Chen, "Bikenet: Accurate bike demand prediction using graph neural networks for station rebalancing," in *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. IEEE, 2019, pp. 686–693.
[45] J. Liu, L. Sun, W. Chen, and H. Xiong, "Rebalancing bike sharing systems: A multi-source data smart optimization," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1005–1014.
[46] M. Lowalekar, P. Varakantham, S. Ghosh, S. D. Jena, and P. Jaillet, "Online repositioning in bike sharing systems," in *Twenty-Seventh International Conference on Automated Planning and Scheduling*, 2017.
[47] Y. Duan and J. Wu, "Spatial-temporal inventory rebalancing for bike sharing systems with worker recruitment," *IEEE Transactions on Mobile Computing*, 2020.
[48] C. Kloimüllner, P. Papazek, B. Hu, and G. R. Raidl, "Balancing bicycle sharing systems: an approach for the dynamic case," in *European conference on evolutionary computation in combinatorial optimization*. Springer, 2014, pp. 73–84.
[49] F. Chiariotti, C. Pielli, A. Zanella, and M. Zorzi, "A dynamic approach to rebalancing bike-sharing systems," *Sensors*, vol. 18, no. 2, p. 512, 2018.
[50] J. Brinkmann, M. W. Ulmer, and D. C. Mattfeld, "Dynamic lookahead policies for stochastic-dynamic inventory routing in bike sharing systems," *Computers & Operations Research*, vol. 106, pp. 260–279, 2019.
[51] F. Chiariotti, C. Pielli, A. Zanella, and M. Zorzi, "A bike-sharing optimization framework combining dynamic rebalancing and user incentives," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 14, no. 3, pp. 1–30, 2020.
[52] H. Chung, D. Freund, and D. B. Shmoys, "Bike angels: An analysis of citi bike's incentive program," in *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, 2018, pp. 1–9.

Hang Zhu received his B.S. from the Department of Software Engineering, Wuhan University of Technology. He is currently pursuing a M.S. Degree in the Department of Computer Science and Technology at Xiamen University. His research interests focus on Graph Neural Network and Urban Computing.

Tieqi Shou received the B.S. Degree from College of Computer Science and Technology, Zhejiang University of Technology, China in 2020. He is currently pursuing a M.S. Degree in the School of Informatics, Xiamen University. His research interests focus on Spatial Data Mining and Urban Computing.

Ruiying Guo received his B.S. from School of Environment and Spatial Informatics, China University of Mining and Technology, in 2018. He is currently pursuing a M.S. Degree in the Department of Computer Science and Technology at Xiamen University. His research interests focus on Spatiotemporal Data Mining and Urban Computing.

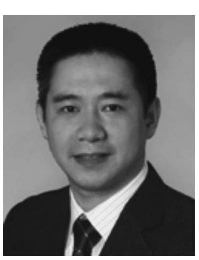
Zhihan Jiang received the B.E. and M.E. degrees in computer science and technology from Xiamen University, China in 2018 and 2021, respectively. She is currently pursuing the Ph.D. degree in Department of Electrical and Electronic Engineering, The University of Hong Kong. Her research interests include Mobile Computing, Urban Computing, Crowdsensing, and Data Analysis.

Zeyu Wang is currently pursuing the B.Eng. degree in Department of Computer Science, Xiamen University. His research interests include Big Data Analytics, Crowdsensing and Urban Computing.

Zhiyuan Wang is currently a Ph.D. student in system engineering with the University of Virginia. He received his B.E. degree from School of Informatics, Xiamen University, in 2021. His research interests include ubiquitous computing, mobile sensing and computing, and data mining.

Zhiyong Yu received the M.E. and Ph.D. degrees in computer science and technology from Northwestern Polytechnical University, Xi Xi'an, China in 2007 and 2011, respectively. He is a full professor in the Department of Computer Science, Fuzhou University, Fuzhou, China. He was also a visiting student at Kyoto University, Kyoto, Japan, from 2007 to 2009 and a visiting researcher at the Institut Mines Mines-Telecom, TELECOM SudParis, Evry, France, from 2012 to 2013. His current research interests include pervasive computing, mobi le social networks, and mobile crowd sensing. Email: yuzhiyong@fzu.edu.cn.

Weijie Zhang is currently studying for a master's degree in Fuzhou University, China. His research interests include Ubiquitous Computing and Crowdsensing.



Cheng Wang received the Ph.D. degree in information and communication engineering from the National University of Defense Technology, Changsha, China in 2002. He is currently a Professor with Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University, China. His research interests include remote sensing image processing, mobile LiDAR data analysis, and multisensor fusion.

Longbiao Chen is an assistant professor with Xiamen University, China. He obtained his Ph.D. degree in computer science from Sorbonne University, France in 2018 and Zhejiang University, China in 2016. His research interests include Mobile Computing, Big Data Analytics, Crowdsensing and Urban Computing. He has published over 30 papers in top-tier journals and conferences, including IEEE T-ITS, IEEE T-HMS, Elsevier JNCA, and ACM UBICOMP. He received two UBICOMP Honorable Mention Awards in 2015 and 2016. He is a technical committee member of ACM SIGSPATIAL China, and serves as the PC members of IJCAI and UIC.