BikeNet: Accurate Bike Demand Prediction Using Graph Neural Networks for Station Rebalancing

Ruiying Guo[1], Zhihan Jiang[1], Jingchun Huang[1], Jianrong Tao[2], Cheng Wang[1], Jonathan Li[1,3], Longbiao Chen*[1]

[1] Fujian Key Laboratory of Sensing and Computing for Smart City,
School of Information Science and Technology, Xiamen University, Xiamen, China
[2] Fuxi AI Lab, NetEase Inc., Hangzhou, China
[3] WatMos Lab, University of Waterloo, Waterloo, Canada
Email: longbiaochen@xmu.edu.cn

***Abstract*—Bike sharing systems are widely operated in many cities as a green transportation means to solve the last mile problem and to reduce traffic congestion. One of the key challenges in operating high quality bike sharing systems is rebalancing bike stations from being full or empty. To this end, operators usually need to foresee the bike demands and schedule trucks to reposition bikes among stations. However, an accurate prediction of city-wide bike demands is not trivial due to the spatial correlation and temporal dependency of user mobility dynamics. Moreover, finding an optimal station rebalancing strategy from potentially enormous candidates is challenging given resource optimization objectives. In this work, we propose a two-phase framework to accurately predict city-wide bike demands and effectively rebalance bikes stations leveraging state-of-the-art deep learning techniques. First, we build a spatiotemporal graph neural network (ST-GNN) to model and predict city-wide bike demands, simultaneously capturing the spatial correlation by Graph Convolutional Networks (GCN) and the temporal dependency by Gated Recurrent Units (GRU). Then, we formulate the truck-based station rebalancing problem as an optimization problem with transportation cost objectives, and effectively solve the problem with Integer Linear Programming (ILP) algorithm. Experiments on real-world datasets from New York City validate the performance of the proposed framework, reducing 13% of prediction error and 5% of transportation cost compared with the baseline methods.**

***Index Terms*—Bike sharing systems; graph neural networks; demand prediction; station rebalancing; data analytics**

I. INTRODUCTION

Bike sharing systems are widely operated in major cities around the world, with over 1,000 active systems as of December 2016 [1]. The users of these systems can easily pick and return public bikes at self-service stations scattered around a city to make short trips [2]. However, as the demand for bike increases, the supplies of bikes at different stations quickly become unbalanced, which greatly hurts the users' bike experience [2], [3]. On one hand, if the user demand is greater than the bike supply in a station, it becomes difficult for the users to rent bikes from the station. On the other hand, if the bike supply is greater than the user demand, it becomes difficult for users to return bikes due to insufficient number of docks. Therefore, effective station rebalancing approaches to prevent stations from over-demand are in great need for bike sharing system operators.

Currently, there are two main approaches for rebalancing bikes in a system, i.e., *user-based rebalancing* and *truck-based rebalancing* [4]. The user-based rebalancing approaches usually incentivize the users in the bike repositioning process, encouraging them to pick or return bikes in specific stations in exchange for monetary incentives [5]. However, such an approach may require users to change their intended journey, and consequently the effectiveness of station rebalancing is affected by the willingness of user participation [5]. Moreover, how much incentive reward to offer to different users is a challenging problem given specific budget constraints. Therefore, in this work, we concentrate on the truck-based rebalancing approaches widely adopted in many bike sharing systems [6].

In general, there are two steps in truck-based rebalancing approaches, i.e., *demand prediction* and *station rebalancing*. First, it is crucial to *accurately* predict the demand at each station to foresee the bike and dock availability in the future. Second, it is important to design effective strategies for truck operators to reposition bikes among stations. However, these two steps are challenging due to the following issues.

- **How to accurately model the spatiotemporal dynamics of bike demands?** Since bike sharing systems are continuously operated city-wide, the bike demands of users demonstrate strong intrinsic *spatial correlation* and *temporal dependency* [7]. Therefore, bike demand prediction models need to take both spatial and temporal features into consideration. However, due to uneven bike station distribution and dynamic user movement, it is challenging to capture the spatial correlation of bike demands. Moreover, the temporal variations of bike demands are fluctuating with latent trends and periods, making it difficult to model the underlying temporal patterns.
- **How to effectively find the optimal strategy for station rebalancing?** Due to the tremendous bike repositioning schemes among stations, station rebalancing problem is usually considered NP-hard [8]. Therefore, the time to find an optimal rebalancing solution grows exponentially and quickly becomes intractable for city-wide bike sharing systems. Moreover, since the number of available

trucks and the amount of rebalancing budgets are usually limited, these resource-constraints should also be imposed in the formulation of the station rebalancing problem.

Fortunately, with the rapid evolution of deep learning techniques, the emergence of *graph neural networks* has presented new possibilities to model complicated spatial correlation and dynamic temporal dependency, providing us with new opportunities to address the above-mentioned issues. In this work, we propose to model city-wide, time-varying bike demands as spatiotemporal graph neural networks, and learn its latent graph structures with large-scale historical data. Consequently, we accurately predict city-wide bike demands. Based upon this, we formulate station rebalancing as an optimization problem, and effectively solve it with integer linear programming algorithms. In summary, our contributions include:

- We propose a two-phase framework to design an optimal bike balancing strategy with accurate bike demand prediction. In the first phase, we build a *spatiotemporal graph neural network (ST-GNN)* to model and predict the city-wide bike demands, which captures the spatial correlation by Graph Convolutional Networks (GCN) and the temporal dependency by Gated Recurrent Units (GRU). In the second phase, we formulate the truck-based station rebalancing problem as an optimization problem with transportation cost objectives, and effectively solve the problem with Integer Linear Programming (ILP) algorithm to find the exact optimal solution.
- We evaluate our proposed framework using real-world bike sharing system data collected from New York City's Citi Bike system in one year. Results show that the proposed framework can not only accurately predict bike demands with reducing 13% error, but also find an optimal station rebalancing strategy that reduce 5% of transportation cost, which consistently outperforms other baseline methods.

The rest of this paper is organized as follows. We first present preliminaries and framework overview in Section II, and then describe two phases of our proposed framework in Section III and IV, respectively. We report the evaluation results in Section V, and present a brief survey of related works in Section VI. Finally, we conclude our work and discuss future directions in Section VII.

II. Preliminaries and Framework Overview

A. Preliminaries

Definition 1: **Station Status**: the status of bike station i at time t is defined as a tuple $< B_i^{(t)}, D_i^{(t)} >$, where $B_i^{(t)}$ and $D_i^{(t)}$ are the number of available bikes and docks in station i at time t, respectively.

Definition 2: **Time Span**: we divide the duration of observation data into equal time spans Δt, each time span lasts for a period of time, e.g., one hour.

Definition 3: **Bike Demand**: the bike demand of station i during time span Δt is defined as the number of bikes rented from the station minus the number of bikes returned to the station during Δt.

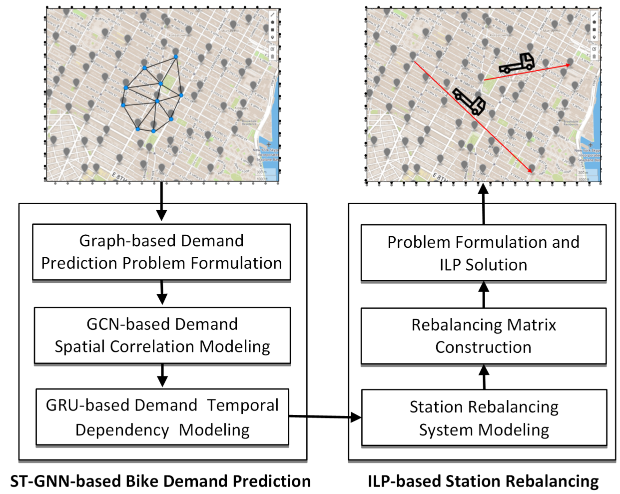


Fig. 1. Framework overview.

B. Framework Overview

As shown in Fig.1, we propose a two-phase framework to accurately predict city-wide bike demands and effectively rebalance bike stations leveraging state-of-the-art deep learning techniques. In the demand prediction phase, we first exploit a spatiotemporal graph to formulate the demand prediction problem in the bike sharing system. We then build a spatiotemporal graph neural network (ST-GNN) to predict city-wide bike demands, simultaneously capturing the spatial correlation by Graph Convolutional Networks (GCN) and the temporal dependency by Gated Recurrent Units (GRU). In the station rebalancing phase, we first formulate the truck-based station rebalancing problem as an optimization problem with transportation cost objectives, and then construct rebalancing pairs. Finally, we effectively solve the problem with Integer Linear Programming (ILP) algorithm.

III. ST-GNN-based Bike Demand Prediction

In the bike demand prediction phase, our objective is to accurately predict the city-wide bike demand patterns of a bike sharing system for a future period of time, which is not trivial due to the intrinsic spatial and temporal dynamics of bike demands. For example, Fig.2 shows the demand patterns of two stations in New York City during two weeks. The stations in the business district usually observe riding peaks during rush hours, while the station in the residential area are intensively used during after work hours. It is difficult to model the city-wide dynamics of bike demand patterns using traditional time series analysis techniques (e.g., ARIMA models [9] and feed-forward neural networks [10]). To address these challenges, we leverage the spatiotemporal graph neural network (ST-GNN) to model the spatial correlation and temporal dependency of bike sharing systems for accurate prediction. We elaborate the details as follows.

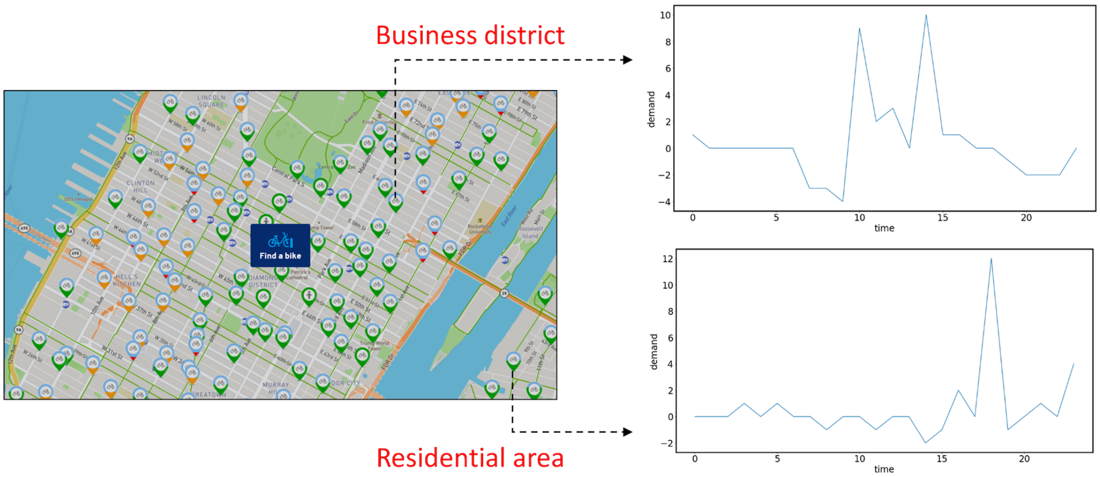


Fig. 2. An illustrative example of bike demand variations in two bike stations from New York City on 02/01/2015, observed hourly based on the Citi Bike system.

A. Graph-based Demand Prediction Problem Formulation

The objective of city-wide bike demand prediction is to predict the bike demands of each station in a future period of time, given previously demand observations. To this end, we build a spatiotemporal graph to model the demand variations in the bike sharing system. Specifically, we represent the bike sharing system as a *weighted undirected spatiotemporal graph* $G=(V,E)$, where V denotes the graph nodes that represent the bike stations in the system, and E denotes the set of edges between the station pairs. Based on this graph structure, we model the spatiotemporal dynamics of bike demands using node values and edge weights as follows.

Node Values (X): we define the value of node $v \in V$ as $X_v^{(t)}$, which is calculated as the bike demand of station v during $[t, t+\Delta t]$. Consequently, we denote the node values of graph G as a matrix $X \in \mathbb{R}^{N_t \times N_s}$, where N_t is the number of time spans, and $N_s = |V|$ the number of stations in the system.

Edge Weights (W): we define the weight of edge $e_{i,j} = \{v_i, v_j\} \in E$ as $W(v_i, v_j)$, which is calculated based on the correlation between station v_i and station v_j. In this work, we model the station correlation based on the *geographical distance* between them. We note that the correlation modeling can directly adapt to other application scenarios, e.g., by calculating the similarity of point-of-interest distributions among stations. Based on the above definitions, we formulate the demand prediction problem as follows.

Demand Prediction Problem: Let $X^{(t)}$ represent the node values of the graph observed at time t, the objective of the demand prediction problem is to learn a function $f(\cdot)$ that maps N_p historical graph node values to N_f futural graph node values given the spatiotemporal graph structure, i.e.,

$$[X^{(t+1)}, ..., X^{(t+N_f)}; W] = f([X^{(t-N_p+1)}, ..., X^{(t)}; W]) \quad (1)$$

Finding the mapping function for Problem 1 is not trivial, not only because the temporal dependency of node values X is complicated, but also because the node values exhibit strong spatial correlation constrained by W. Therefore, we construct a spatiotemporal graph neural network (ST-GNN) model to learn the mapping function from historical data for accurate node value prediction. Specifically, we exploit a graph convolutional network unit to model the spatial correlation, and a gated recurrent units to model the temporal dependency. The overview of the proposed ST-GNN architecture is shown in Fig. 3. We elaborate the details as follows.

B. GCN-based Spatial Correlation Modeling

In spatial correlation modeling, current works usually employ grid to divide the urban area, converting urban data to Euclidean domains, and then use Convolutional Neural Networks (CNN) to model spatial correlation [11]. However, in our problem, the spatial distribution of bike stations are irregular and in non-Euclidean domains. Hence, we introduce graph structure to model the spatial distribution of bike stations and exploit Graph Convolutional Networks (GCN) to model spatial correlation.

Specially, we first model the dynamics of the bike demand as a diffusion process [12], and convert it into the Fourier domain. Then, we introduce the graph convolution operation [13] to model the spatial correlation of related bike stations to capture the underlying spatial closeness and regional features. Therefore, we can explicitly capture the stochastic spatial correlation of bike demand dynamics.

Spectral Graph Convolution. Graph convolution is very effective on non-Euclidean domains [14]. However, it is challenging to construct a convolution operator in the vertex domain. To address this challenge, Bruna et al. [14] proposed spectral networks and locally connected networks on graphs based on graph spectrum theory. Graph spectrum is the set of graph eigenvalues of the adjacency matrix of the graph. With the spectral graph convolution [13], we can easily define the convolution operator on graph in the Fourier domain. The spectral graph convolution defined as

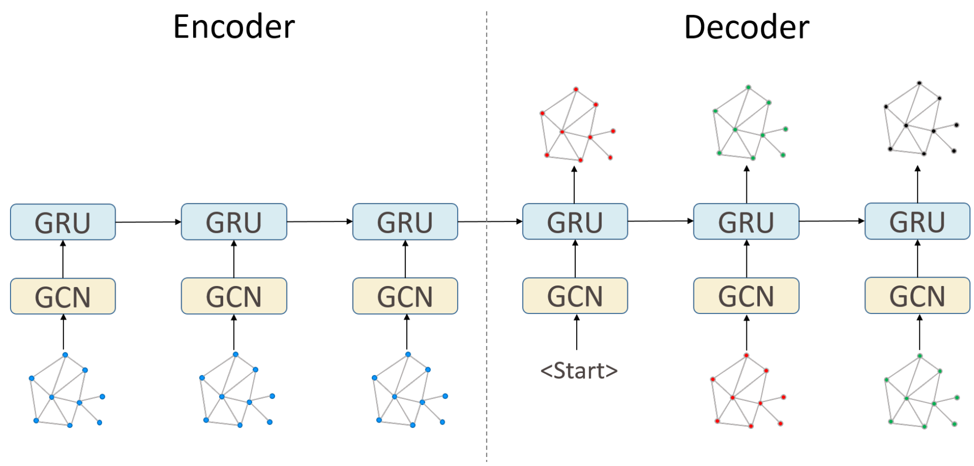


Fig. 3. Model architecture for the spatiotemporal graph neural network (ST-GNN).

$$X * \mathscr{G} y = U((U^T x) \odot (U^T y)) \tag{2}$$

where U is the eigenvectors of graph Laplacian matrix, and $\odot$ is the element-wise Hadamard product. In order to train graph neural networks model efficiently, we use GCN architecture designed by Kipf et al. [15].

C. GRU-based Temporal Dependency Modeling

We leverage the recurrent neural networks (RNNs) to model the temporal dependency. In particular, we use Gated Recurrent Units (GRU) [16], which is a powerful variant of RNNs overcoming gradient vanishing and gradient exploding problem effectively. Encoder-decoder structure is widely used in spatiotemporal sequence predicting tasks because it has been verified very effective. Therefore, we introduce encoder-decoder structure to build sequence to sequence architecture along with GRU units. Sequence to sequence is effective in multiple step ahead prediction.

Gated Recurrent Unit. Recurrent Neural Networks (RNNs) can model the dependency of time series effectively [7]. However, the traditional RNN models have limitations for long-term prediction due to gradient vanishing and gradient exploding problems. Although Long Short-term Memory (LSTM) [17] can address these challenges, it has the defect of more training time consumption special for complex structures. Hence, we introduce Gated Recurrent Unit (GRU), the variant of RNNs, to model the temporal dependency. Compared with LSTM, GRU model has a relatively simple structure with fewer parameters and faster training speed.

D. The Spatiotemporal Graph Neural Network Architecture

With both spatial correlation and temporal dependency modeling, we build a spatiotemporal graph neural network (ST-GNN) to accurately predict the bike demand. Fig.3 shows the model architecture of ST-GNN. The objective function of ST-GNN is to maximize the likelihood of predicting the time series in a future period of time. Specifically, the ST-GNN model consists of two parts, i.e. GCN-based spatial correlation modeling and GRU-based temporal dependency modeling. Moreover, the whole network is trained by using backpropagation through time (BPTT). To avoid gradient vanishing and gradient exploding problem, we employ Adam optimization algorithm [18] to address these challenges.

In particular, we employ the sequence to sequence [19] architecture to enable multiple step ahead prediction. Both the encoder and the decoder are recurrent neural networks with combination of GCN and GRU. At training step, we feed the historical time series into the encoder and use its final states to initialize the decoder. The decoder generates predictions given previous ground truth observations. During testing step, encoder-decoder model replaces the ground truth observations with predictions generated. The conflict between the input distributions of training and testing can cause degraded performance. To mitigate this issue, we introduce scheduled sampling [20] into the model to improve the performance.

IV. ILP-BASED STATION REBALANCING

In this phase, given the accurate prediction of city-wide bike demands, our objective is to find an optimal station rebalancing strategy to prevent stations from over-demand. To this end, we first identify the station rebalancing problem, and then propose an integer linear programming-based solution.

More specifically, when a station rebalancing task is to be conducted, we first accurately predict the bike demands for the next time span (e.g., one hour). We then filter the stations that need to be rebalanced, i.e., the bike numbers in these stations are out of a reasonable balanced range (the details are elaborated below). Finally, we define a rebalancing problem for these stations and the available scheduling resources.

One of the key challenge in solving the station rebalancing problem is that such a problem is usually NP-hard [8], due to the tremendous station-to-station combinations for moving bikes. In addition, this problem is resource-constrained due to limited rebalancing resources (e.g., budgets and truck numbers) [21]. In this work, we exploit the Integer Linear Programming (ILP) approach to effectively find the exact solution to this problem. We elaborate the details as follows.

A. System Modeling

We model the above-mentioned problem as an optimization problem, where the objective is to minimize the cost of transporting bikes from the over-demand stations to over-demand stations. We define the reasonable number of the bikes in a station is in range $[lb, ub]$, therefore for each over-supply station i, the number of bikes that can be removed is in range $[s_i - ub, s_i - lb]$, and for each over-demand station j, the number of bikes that can receive is in range $[lb - d_j, ub - d_j]$. We then construct a set of rebalancing pairs, as detailed in the next subsection. For each rebalancing pair (i, j), if $[s_i - ub, s_i - lb] \bigcap [lb - d_j, ub - d_j] \neq \emptyset$, which means that we can not rebalance these two stations through inner rebalancing, so we remove the corresponding column in matrix A. After removing all invalid rebalancing pairs, we get the matrix $A' = [p_1, p_2, ..., p_l]_{(n+m,l)}$, where l is the number of valid pairs, and $p_i(i = 1, ..., l)$ represents the valid rebalancing pair vector. Then we construct the cost vector $C = [c_1, c_2, ..., c_l]'$, where $c_i(i = 1, ..., l)$ is the transportation cost of rebalancing pair p_i.

B. Rebalancing Matrix Construction

Let $S = \{s_1, s_2, \ldots, s_n\}$ and $D = \{d_1, d_2, \ldots, d_m\}$ be the n over-supply stations and the m over-demand stations of bike sharing systems, respectively, where $s_i(s = 1, ..., n)$ is the initial number of bikes in over-supply station i and $d_j(j = 1, ..., m)$ is the initial number of bikes in over-demand station j. We define each pair (i, j) consists over-supply station i and over-demand station j a rebalancing pair, and denote it as a 0-1 vector $v = [v_{s_1}, v_{s_2}, ..., v_{s_n}, v_{d_1}, ..., v_{d_m}]'_{n+m,1}$, where $v_{s_i} = 1$, $v_{s_k} = 0(k = 1, ..., n, k \neq i)$, $v_{d_j} = 1$, $v_{d_k} = 0(k = 1, ..., m, k \neq j)$, which means that the bikes are removed from station i to station j. Thus we get a rebalancing matrix consists of all rebalancing pairs, i.e.,

$$
\begin{array}{c}
A = \\
1 \\ 2 \\ ... \\ n \\ n+1 \\ n+2 \\ ... \\ n+m
\end{array}
\begin{array}{c}
\begin{array}{cccccc} 1 & 2 & ... & n & ... & n \times m \end{array} \\
\begin{pmatrix}
1 & 1 & ... & 1 & ... & 0 \\
0 & 0 & ... & 0 & ... & 0 \\
... & ... & ... & ... & ... & ... \\
0 & 0 & ... & 0 & ... & 1 \\
1 & 0 & ... & 0 & ... & 0 \\
0 & 1 & ... & 0 & ... & 0 \\
... & ... & ... & ... & ... & ... \\
0 & 0 & ... & 1 & ... & 1
\end{pmatrix}
\end{array}
$$

C. Problem Formulation and ILP Solution

With the above definitions, we present the formulation of the bike transportation problem with the objective of minimizing the transportation cost under the constraint that after rebalancing, the number of the balanced stations transformed from unbalanced stations should be maximized.

$$minimize \quad C^T x \tag{3}$$

subject to

$$\sum_{j=1}^{l} a_{ij} x_j \leq 0 (i = 1, ..., n+m) \tag{4}$$

$$x_j (j = 1, ..., n+m) \geq 0 \tag{5}$$

$$\sum_{i=1}^{n+m} \sum_{j=1}^{l} a_{ij} x_j = 2min(n, m) \tag{6}$$

In the literature, various techniques have been proposed to solve this problem, such as integer linear programming and heuristic search [22]. The basic ideas include narrowing the solution space, finding integer-feasible solutions, and discarding space without better integer-feasible solutions [23]. In this work, we exploit the integer linear programming approach to effectively find the exact solution to this problem. In particular, we employ the Integer Linear Programming Solver from the MATLAB Optimization Toolbox[1] to find the optimal solution.

V. Evaluation

A. Dataset Description

TABLE I
Dataset Description

Statistics	New York City
Stations	330
Bike trips	10,669,470
Station status	hourly
Time span	2015/01/01-2015/12/31

We evaluate our proposed framework on real-world datasets from New York City's Citi Bike system. We collect bike trip data for one year (2015/01/01-2015/12/31). As presented in TABLE I, all of the data are in forms of bike trip record including trip duration, start time, start station, stop time, stop station and so on.

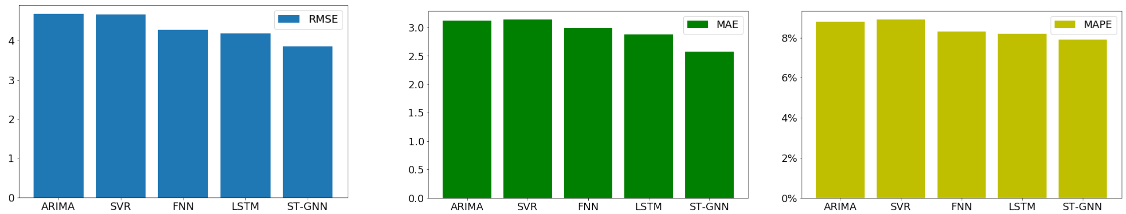
In order to evaluate our predicted result, we split the datasets as training data, validation data and test data by choosing the last 73 days as test data, the first 256 days as training data and the rest of days as validation data. Bike demand period is empirically set as one hour.

B. Evaluation on Demand Prediction

Evaluation Metrics: We use three commonly used metrics in demand prediction, including (1) Root Mean Squared Error (RMSE), (2) Mean Absolute Error (MAE), (3) Mean Absolute Percentage Error (MAPE). They are defined as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{7}$$

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



(a) RMSE (b) MAE (c) MAPE

Fig. 4. Performance comparison of the proposed method and the baseline methods.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{8}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \tag{9}$$

Baselines: We compare our method with two sets of baselines, i.e., machine learning algorithms and deep learning algorithms. We choose ARIMA, SVR, FNN and LSTM algorithms as baseline methods.

- **ARIMA**: Auto-Regressive Integrated Moving Average (ARIMA) [9] is widely used in time series analysis. This baseline method models the bike demand in a station as time series, and it does not directly consider changes in other related random variables.
- **SVR**: Support Vector Regression (SVR) [24] is the regression method of Support Vector Machine. SVR uses the same principles as the SVM to minimize error, individualizing the hyperplane which maximizes the margin, keeping in mind that part of the error is tolerated.
- **FNN**: Feedforward Neural Networks (FNN) [10] is the basis architecture of deep learning method. We design the neural networks with two hidden layers and Adam optimization algorithm.
- **LSTM**: Long Short-term Memory (LSTM) [17] has feedback connections, i.e. memory cells and forget gates, to overcome gradient vanishing and gradient exploding problem. This baseline employs recurrent neural networks with LSTM hidden units to predict the bike demand.

Results: Fig.4 shows that spatiotemporal graph neural network (ST-GNN) consistently achieves the best performance among all the baselines. As shown in TABLE II, compared with other baselines, ST-GNN shows at least 13%, 14% and 7% improvements on RMSE, MAE and MAPE relative error reduction respectively. In conclusion, ST-GNN can model spatial correlation and temporal dependency effectively.

C. Evaluation on Station Rebalancing

Evaluation Metrics: We evaluate station rebalancing performance with transportation distance, the conversion ratio and the average running time. Conversion ratio is the ratio of X stations transformed from unbalanced to balanced status after rebalancing and Y unbalanced stations before rebalancing. Average running time is the time consumption after repeating

TABLE II
PREDICTION EVALUATION RESULTS

	RMSE	MAE	MAPE
ARIMA	4.68	3.12	8.8%
SVR	4.67	3.14	8.9%
FNN	4.27	2.99	8.3%
LSTM	4.19	2.88	8.2%
ST-GNN	**3.86**	**2.58**	**7.9%**

TABLE III
REBALANCING EVALUATION RESULTS

	GA	ILP
Transportation Distance(m)	4931.5	**4684.1**
Conversion Ratio(%)	33.5	**40.0**
Average Running Time(s)	0.019	**0.006**

100 times. Transportation distance and conversion ratio are defined as follows:

$$\text{Transportation Distance} = \sum_{j=1}^{m}\sum_{j=1}^{n} c_j x_j \tag{10}$$

$$\text{Conversion Ratio} = \frac{X}{Y} \times 100\% \tag{11}$$

Baselines: The effectiveness of Integer Linear Programming algorithm is compared to heuristic algorithm as baseline method, i.e. genetic algorithm (GA) [25]. GA is a genetic algorithm supposed to be efficient for vehicle routing optimization problems [26].

Results: For the station rebalancing results, the ILP consistently achieves the best performance among all the baselines. As shown in TABLE III, compared with baseline method, ILP model reduces 5% of transportation distance cost, and is much more effective than GA by providing optimal rebalancing strategy with much smaller traveling distances.

D. Case Study

Finally, we conduct case study analysis to further understand the benefit and limits of our framework in morning rush hour in New York City. Fig.5 shows the comparison of bike stations status before rebalancing and after rebalancing, respectively. We can see that before 9:00 on December 15, 2015, the system is not balanced due to it is the busy part of the day when people are commuting to work. Based on accurate bike demand

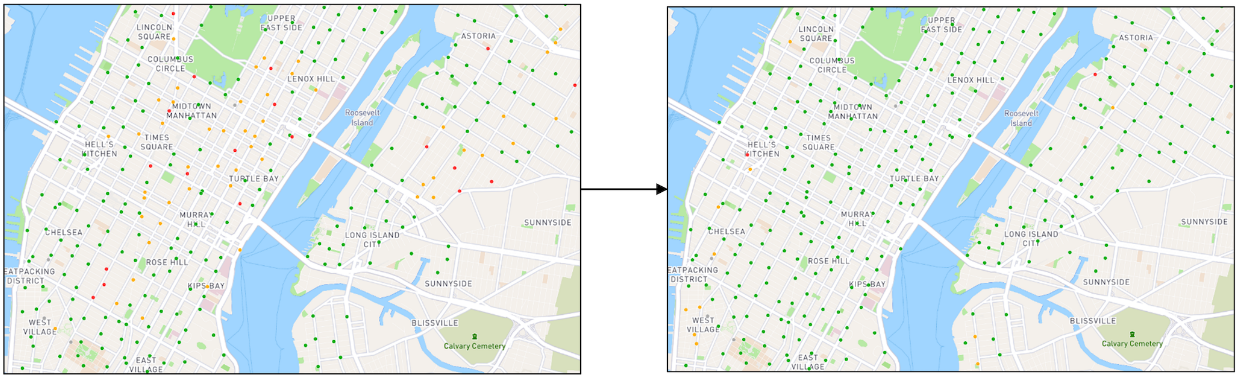


Fig. 5. Case study in morning rush hour in New York City on December 15, 2015. Stations status are represented by different color: green color represents balanced status, red color represents full status and yellow color represents empty status. Left figure shows the system is not balanced before rebalancing at 9:00. After conducting rebalancing based on our method, right figure shows the system becomes balanced.

prediction and station rebalancing, our framework successfully generate a rebalancing scheme that greatly reduces the number of unbalanced stations, validating the effectiveness of the proposed approach.

VI. Related Work

We describe the related work from two perspectives, i.e., spatiotemporal sequence prediction and station rebalancing approaches.

A. Spatiotemporal Sequence Prediction

Spatiotemporal sequence prediction is a fundamental problem for data-driven urban planning and management. Nowadays, machine learning based methods for Spatiotemporal sequence prediction, including both classical machine learning methods and advanced deep learning methods, have special designed in their model architectures for capturing these spatiotemporal correlation [27]. For classical methods, there are three kinds of methods, i.e. feature-based, state space models and gaussian process, such as Spatiotemporal indicator [28], STARIMA [29] and GP [30]. For deep learning methods, there are two categories of methods, i.e. Deep Temporal Generative Models (DTGMs) and Feedforward Neural Networks and Recurrent Neural Networks (FNN & RNN). There are tremendous amount of works on FNN & RNN topic. For example, Srivastava et al. [31] proposed to use multi-layer FC-LSTM networks to predict spatiotemporal sequence, but ignored the spatial information. Shi et al. [32] proposed the Convolutional LSTM (ConvLSTM) to capture spatial correlation, first modeling the spatial and temporal dependency successfully. Shi et al. [33] proposed the Trajectory GRU (TrajGRU) to model location-variant motions, such as translation and rotation. Zhang et al. [11] employed grid to divide city and proposed Deep Spatio-Temporal Residual Networks (ST-ResNet) to collectively predict the citywide crowd flows.

Recently, graph-based deep learning methods on non-Euclidean domains have sprung up. Unlike traditional methods modeling the spatial feature as regular and Euclidean domains, Graph Convolutional Networks (GCN) employs graph structure to model the spatial information. Moreover, it is effective to combine GCN with RNN to model both spatial and temporal dependency. Li et al. [12] proposed Diffusion Convolutional Recurrent Neural Network (DCRNN) for long-term traffic prediction. Yu et al. [34] proposed Spatio-Temporal Graph Convolutional Networks (STGCN) for traffic prediction, introducing graph convolution and gated temporal convolution through ST-Conv block.

B. Station Rebalancing Approaches

With the rapid evolution of sharing economy, the emergence of bike sharing systems has presented new opportunities for researchers to address operational challenges by leveraging big data [2]. There are two main rebalancing approaches that are truck-based approach and user-based approach [4]. Truck-based approach can be classified into two categories, including static station rebalancing and dynamic rebalancing. Static station rebalancing means that the operators reposition bikes at the bike stations when they are not operating or in the midnight. Liu et al. [26] employed optimization models to optimize the total transportation cost, i.e. distant. Dynamic station rebalancing is an online approach to compute dynamic bike repositioning and real-time routing plan. Lowalekar et al. [35] proposed a multi-stage stochastic formulation modeling the future demand to find the best rebalancing strategy for bike sharing systems. Nowadays, user-based approach is widely used in dockless bike sharing systems by incentivizing the customers to reposition bikes along designated routes at specific bike stations with a reward. Singla et al. [5] introduced crowdsourcing mechanism to station rebalancing by incentivizing the customers with a reward.

VII. Conclusion and Future Work

In this paper, we investigated the problem of accurate bike demand prediction in bike sharing systems for effective station rebalancing. We proposed a data-driven framework to address the challenges by leveraging state-of-the-art graph neural network and optimization technologies. Specifically, in the first phase, we built a spatiotemporal graph neural network (ST-GNN) model to predict the city-wide bike demands, simultaneously capturing the spatial correlation and the temporal dependency in a unified network architecture. In the second phase, we formulated the truck-based station rebalancing problem as an optimization problem with transportation cost objectives, and effectively solved the problem with Integer Linear Programming (ILP) algorithm to find the exact optimal solution. Experiments on real-world datasets from New York City's Citi Bike system validated the effectiveness of our framework, which outperformed other baseline methods.

In the future, there are still several issues to be investigated. First, for long-term demand predictions (e.g., one day ahead), we plan to incorporate the attention mechanism in the ST-GNN model to enable accurate and consistent multi-step prediction. Second, we plan to consider external contextual factors, such as weather and social events, in modeling and predicting bike demands. Third, we plan to explore more effective linear programming techniques, e.g., dimension reduction, to boost the performance when solving the bike rebalancing problem.

VIII. Acknowledgment

We would like to thank the reviewers for their constructive suggestions. This research was supported by National Natural Science Foundation of China (NSFC-61802325), Natural Science Foundation of Fujian Province (2018J01105), China Fundamental Research Funds for the Central Universities (20720170040).

References

[1] J. Schuijbroek, R. C. Hampshire, and W.-J. Van Hoeve, "Inventory rebalancing and vehicle routing in bike sharing systems," *European Journal of Operational Research*, vol. 257, no. 3, pp. 992–1004, 2017.

[2] 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.

[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] L. Pan, Q. Cai, Z. Fang, P. Tang, and L. Huang, "A Deep Reinforcement Learning Framework for Rebalancing Dockless Bike Sharing Systems," *arXiv preprint arXiv:1802.04592*, 2018.

[5] 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.

[6] C. Contardo, C. Morency, and L.-M. Rousseau, "Balancing A Dynamic Public Bike-Sharing System," CIRRELT, Tech. Rep., 2012.

[7] 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.

[8] R. M. Karp *et al.*, "Complexity of computer computations," *Reducibility among combinatorial problems*, vol. 23, no. 1, pp. 85–103, 1972.

[9] 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.

[10] T. L. Fine, *Feedforward Neural Network Methodology*. Springer Science & Business Media, 2006.

[11] 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.

[12] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[13] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in neural information processing systems*, 2016, pp. 3844–3852.

[14] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[15] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[16] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

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

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, 2014, pp. 3104–3112.

[20] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 1171–1179.

[21] 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.

[22] L. A. Wolsey and G. L. Nemhauser, *Integer and Combinatorial Optimization*. John Wiley & Sons, 2014.

[23] C. Chen, D. Zhang, X. Ma, B. Guo, L. Wang, Y. Wang, and E. Sha, "Crowddeliver: planning city-wide package delivery paths leveraging the crowd of taxis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1478–1496, 2016.

[24] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[25] F. Zhao, S. Li, J. Sun, and D. Mei, "Genetic algorithm for the one-commodity pickup-and-delivery traveling salesman problem," *Computers & Industrial Engineering*, vol. 56, no. 4, pp. 1642–1648, 2009.

[26] 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.

[27] X. Shi and D.-Y. Yeung, "Machine Learning for Spatiotemporal Sequence Forecasting: A Survey," *arXiv preprint arXiv:1808.06865*, 2018.

[28] O. Ohashi and L. Torgo, "Wind speed forecasting using spatio-temporal indicators." in *ECAI*. Citeseer, 2012, pp. 975–980.

[29] A. Cliff and J. Ord, "Model building and the analysis of spatial pattern in human geography," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 37, no. 3, pp. 297–328, 1975.

[30] 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.

[31] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *International Conference on Machine Learning*, 2015, pp. 843–852.

[32] 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.

[33] X. Shi, Z. Gao, L. Lausen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, "Deep learning for precipitation nowcasting: A benchmark and a new model," in *Advances in Neural Information Processing Systems*, 2017, pp. 5617–5627.

[34] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv preprint arXiv:1709.04875*, 2017.

[35] 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.