DATA-DRIVEN OPERATIONS MANAGEMENT IN BIKE SHARING SYSTEMS

by

JUNMING LIU

A Dissertation submitted to the

Graduate School-Newark

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Management

written under the direction of

Dr. Hui Xiong

and approved by

____

____

____

____

Newark, New Jersey

May 2019

ABSTRACT OF THE DISSERTATION

Data-driven Operations Management in Bike Sharing Systems

By JUNMING LIU

Dissertation Director: Dr. Hui Xiong

The self-service bike sharing systems, which offer an environmentally friendly option for the first-and-last mile transportation, have become prevalent in urban cities. In this dissertation, I aim to integrate the advanced Data Mining techniques and Operations Management algorithms for bike sharing system daily operations management, service area expansion, and station site selection.

Daily Operations Management. Due to the geographical and temporal unbalance of bike usage demand, a number of bikes need to be reallocated among stations during midnight so as to maintain a high service level of the system. To conduct such bike rebalancing operations, I develop a bike demand predictor for station pick-up demand and drop-off demand prediction. Then, a Mixed Integer Linear Programming (MILP) model is formulated to optimize the routing problem of rebalancing vehicles. To address the challenge of computational efficiency, I propose a data-driven hierarchical optimization methodology to decompose the multi-vehicle routing problem into smaller and localized single-vehicle routing problems.

Expansion Area Demand Analysis. Another key to success for a bike sharing systems expansion is the bike demand prediction for expansion areas. I develop a hierarchical station bike demand predictor which analyzes bike demands from functional zone level to station level. Specifically, I first divide the studied bike stations into functional zones by a novel Bi-clustering algorithm which is designed to cluster bike stations with similar POI characteristics and close geographical distances together. Then, the hourly bike check-ins and check-outs of functional zones are predicted by integrating three influential factors: distance preference, zone-to-zone preference, and zone characteristics. The station demand is estimated by studying the demand distributions among the stations within the same functional zone.

Station Site Location Selection. In an ideal bike sharing network, the station locations are usually selected in a way that there are balanced pick-ups and drop-offs among stations. This can help avoid expensive re-balancing operations and maintain high user satisfaction. Here I propose a bike sharing network optimization approach based on an Artificial Neural Network for station demand prediction and a Genetic Algorithm for station site optimization. The goal is to enhance the quality and efficiency of the bike sharing service by selecting the right station locations.

ACKNOWLEDGEMENTS

I would like to express my great gratitude to all the people for their support during my Ph.D. study.

First, I would like to express my deep gratitude to my advisor, Prof. Hui Xiong, for his supervision. I thank him for generously giving me the opportunity to join his data mining group as a research assistant. It was this opportunity that opened the door to a bright future for me. I could not survive without his support, assistance, and friendship for teaching me how to grow to be a successful teacher and researcher.

I also sincerely thank Professor Chen Weiwei for his guidance. Professor Chen has provided me great support for my research and career development over the past three years. His experience and vision in supply chain management have inspired me a lot to solve the challenging problems in my research. I have learned a great deal from the collaboration with him on my several research papers.

I also sincerely thank my other committee members: Professor Lidbetter Thomas, Professor Zhu Xingquan, and Professor Tan Yong. They not only provide constructive suggestions and comments on my work and this thesis, but also offer numerous support and help in my career choice, and I am very grateful for them.

Special thanks to Prof. Daniel Murnick, Prof. Lin Xiaodong, and Prof. Miklos Vasaehelyi for their help in my job hunting. I also owe a hefty amount of thanks to my colleagues and friends Can Chen, Yijung Wang, Meng Qu, Jingyuan Yang, Leilei Sun, Zixuan Yuan, Yanchi Liu, Farid Razzak, Hao Zhong, Qingxin Meng, Wenjun Zhou, Yong Ge, Bin Liu, Yanjie Fu, Zijun Yao for their help, friendship and valuable suggestion.

I would like to acknowledge the Department of Management Science and Information Systems (MSIS) for supplying me with the best imaginable equipment and facilities that helped me to accomplish much of this work.

Finally, my deepest thanks and love to my wife Dr. Li Qiao, who gave me infinite joy, love, and support to make this work possible. Especially, I thank her for making it very challenging to provide a good life as she has been expecting, which pushes me to a higher level in my career. I owe my deepest gratitude and love to my parents and in-laws for their unwearied devotion and unconditional love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Research Background

Recent years have witnessed worldwide prevalence and popularity of public bike sharing system[1] . A bike sharing system provides short-term bike rental service with bike stations scattered over an urban city. Bike users rent the public bike for inner-city transportation from one bike station to another. With the exploding growth, public bike sharing system has rapidly emerged as an innovative and sustainable transportation option in urban cities around the world. Furthermore the advance of IT technology has been greatly adopted to bike sharing systems, such as tracking and locating bikes with GPS, and storing bike usage information computers[2] , which has greatly improved the bike sharing service and reduced the number of stolen bikes.

The emerging prevalence of public bike sharing system has brought various benefits to commuters, transportation systems, and urban stainability. Commuters may not only leave the stress of congested downtown traffic, but also get good exercise by riding bikes. As for transportation systems, the bike sharing systems offer an environment-friendly solution for the first-and-last mile connection and for bridging the gap between existing transportation modes such as subways and rail systems. The

[1] https://en.wikipedia.org/wiki/Bicycle-sharing_system

[2] https://www.bcycle.com/

public bike sharing service is a type of collaborative consumption, which enables the optimization of resources and the reduction of carbon footprint. While the bike sharing system makes the urban area more sustainable, it also changes the landscape of bicycling, creates and enhances communities of people, which consequently increases the safety in neighborhoods.

While the public bike sharing service could provide many benefits on both macro and micro levels, there are many challenges in real practice. **First**, given an urban area, it is challenging to decide the coverage of bike sharing service. Within each selected service coverage, bike sharing service providers need to further determine the specific station locations and estimate the number of docks. To make these decisions, service providers need to take into account many factors such as daily human mobility, existing transportation facility, and the road condition. **Second**, it is very challenging to operate and manage the bike sharing systems in an effective and efficient way. For instance, the dynamics of human mobility may cause inevitable imbalance among all bike stations, i.e., some bike stations may be short of bikes and others may be overstocked for a short-term period. Thus, it will be very crucial for service providers to redistribute bikes among stations in a proactive and economical way in order to ensure the system works effectively. **Third**, It is very important yet challenging to discover the purpose of bike trips by various users. Different users have different motives for using a shared bike. For instance, some users ride bikes for avoiding the traffic and saving the daily commute time, while others may ride bikes mainly for the purpose of exercise. Understanding the purpose of trips will not only help solve the above two challenges but also help vendors decide good marketing strategies for attracting more users. In addition, bike sharing service providers may also need to address other practical challenges such as the potential rider safety issue and reservation policy.

A promising way to tackle the challenges above is to leverage a variety of data that are directly or indirectly related to the public bike sharing service. The first group of data is the bike sharing system data that includes bike trip history and station status information. The bike trip record includes the pick-up/drop off stations and times, and some user information. The station status information includes the vacancy rate of each station at different times. If analyzed properly, this group of data could provide great insight into the dynamics of demand cross different stations and the purpose of bike trips. The second group of data is the traditional public transportation data including taxi GPS logs and bus/train smart card data. Today's taxi GPS logs record locations of a taxi cab in a very fine time granularity and its operation status (e.g., carrying passengers or not) as well. Bus/train smart card data record the origin and destination of each trip and associated times as well. As each trip of a taxi, bus or train represents a movement of a human, we may obtain a good understanding of human mobility by collectively analysing all trips together. This will consequently help us identify the locations where the first/last mile trip exists and infer the demand of bicycle at different location. The third group of data is the Point-of-Interest (POI) data such as the check-ins on Foursquare, which could reveal the popularity density over times. This kind of information may help us identify the crowd areas in a city, where many people often wonder. In these areas, more bikes may be needed than other non-crowd areas. Additional data such as the weather report and geographic condition may also be helpful for addressing challenging of bike sharing service. Particularly, we have already collected partial data from different sources, including historical bike sharing system logs, public transportation data and location-based service data in several urban areas such as New York City and Beijing. Furthermore, we have accomplished some preliminary study with the data (Liu et al., 2015).

However, it is a non-trivial problem to leverage multiple-source data for improving the bike sharing service. First, there is redundant information about human mobility among different types of data such as the taxi GPS log and bus/train smart card log. How to infer the demand of shared bike at an individual station by integrating all types of data together is very challenging. Second, most data are spatio-temporal data, which means observations are associated with location and time. The computation complexity of many data mining tasks with spatio-temporal data is often very high (Shekhar et al., 2015). Therefore, efficient methods will be needed for many prediction and optimization tasks such as predicting the imbalance of bike station and re-balancing the bike station on an hourly base. Third, although we may obtain useful knowledge by mining the data, it is still very challenging to take the knowledge into consideration for optimizing bike station site selection and re-balancing.

1.2 Research Contribution

The major focus of this thesis is on integrating and mining multiple source data (e.g., bike sharing log, human mobility and transportation data) to understand and improve the emerging rental bike service in urban areas around the world. I aim to develop a smart bike sharing system that could not only optimize the selection of rental bike stations but also proactively re-balance the bikes at stations in an optimal way. To validate the developed methods, I will not only adopt traditional machine learning paradigms (e.g., leave-out validation), but also conduct the linear programming methods. Here I identify the following key research contributions:

- **Bike trip profiling**. I categorize and reveal the purpose (e.g., daily commuting) of bike trip by analyzing the rental bike records collected from vendors. The discovered trip purpose distribution over time at each station is leveraged for optimizing the bike station site selection and station re-balancing.

- **Bike station optimization**. The first critical research contribution is to optimize the location of bike stations in a city. There are two essential issued addressed in this component. The first one is to create and select useful features from various human mobility data, identify potential spots for deploying bike stations, and further estimate the dynamic demand of bike at these locations. The second one is how to optimize the selection of stations under different constraints (such as the limited number of available bikes). The objective of the optimization is to maximally meet the demand of bike and minimize the potential imbalance of station.

- **Station in-service area expansion**. To address the challenge of bike demand forecasting in expansion areas, where the historical bike trip records are not available, I develop a hierarchical station bike demand predictor which analyzes bike demands from functional zone level to station level in expansion areas.

- **Station re-balancing**. The objective of this task is to proactively re-balance the bikes at stations. An essential research contribution is to predict the potential imbalance in advance by leveraging both real-time bike station data and human mobility data. With the early prediction of imbalance at each station, the next contribution is to globally optimize the redistribution of bikes under possible constraints such as the financial cost constraint and the number of redistribution trucks.

1.3 Overview

Chapter 2 addresses the issue of bike sharing system unbalanced demand. First, I discuss the motivation and unbalanced human mobility pattern, which result in a low service level of bike rental services. Then I compute the station level bike pick-up demand and drop-off demand. Based on the bike demand prediction and its future station inventory level, I formulate a mixed integer linear programming model to redistribute bikes among bike stations. In addition, to reduce the computational cost of this large-scale optimization problem, I develop a hierarchical optimization method which integrates the capacity constraint K-centers clustering algorithm and 1-vehicle mixed integer programming model. To meet the rebalancing operation of outlier stations (stations with extremely large rebalancing targets), I further develop a partial-visiting strategy and multi-visiting strategy. Experiments based on real-world data validate the effectiveness and efficiency of the developed methods.

Chapter 3 addresses the issue of bike demand forecasting in expansion areas. First, I start from the demand analysis of existing bike sharing systems by integrating human mobility pattern discovery and urban city functional zones. Then a hierarchical demand forecasting model is developed to learn the bike demand from functional zone level to station level. Finally, the zone-zone and station-station bike transition patterns are transferred to the functional zones in expansion areas for bike demand forecasting.

Chapter 4 presents a data-driven bike station site selection model based on an artificial neural network model for bike demand and operational cost prediction and a genetic algorithm for combinatorial optimization of bike station site location selection. Specifically, for each candidate bike station network, we estimate the total demand and total operational cost. The genetic algorithm searches for the optimal station network by testing a better solution which provides a higher total demand and a smaller operational cost.

CHAPTER 2

REBALANCING BIKE SHARING SYSTEMS: A DATA-DRIVEN HIERARCHICAL OPTIMIZATION METHODOLOGY

This chapter focuses on the worldwide Bike Sharing Systems rebalancing problem. Due to the geographical and temporal unbalance of bike usage demand, a number of bikes need to be reallocated among stations during midnight so as to maintain a high service level of the system. To optimize such bike rebalancing operations, two challenges remain: (1) to accurately predict bike pick-up and drop-off demand at station level, so as to determine the rebalancing target for each station, and (2) to efficiently optimize the rebalancing route of multiple dispatching vehicles for the large-scale bike sharing system with the existence of outlier stations, which have large rebalancing targets exceeding vehicle capacity. To this end, we develop a meteorology similarity K-nearest Neighbor regressor and a nonlinear autoregressive network with exogenous meteorology factors (NARX) to predict bike pick-up demand, and a pick-drop bike transition (PDBT) predictor for transition patterns discovery and bike drop-off demand prediction. Then, a Mixed Integer Linear Programming (MILP) model is formulated to optimize the routing problem of rebalancing vehicles. To address the challenge of computational efficiency, we propose a data-driven hierarchical optimization methodology that to decompose the multi-vehicle routing problem into smaller and localized single-vehicle routing problems. Further, we propose two advanced rebalancing strategies: partial target satisfying strategy and multi-vehicle visiting strategy to deal with outlier stations while ensuring the feasibility of the route optimization solution. Finally, extensive numerical results, using real data from the New York City Citi Bike, Chicago Divvy, and Boston Hubway bike sharing systems, show the accuracy of the proposed bike demand predictors, as well as the effectiveness and efficiency of the proposed hierarchical optimization strategies.

2.1 Introduction

The self-service bike sharing systems (DeMaio & Meddin, 2018), which offer an environmentally friendly option for the first-and-last mile transportation, have become prevalent in urban cities. These systems bridge the gaps between public transportation modes such as subways, buses and rail systems, and alleviate traffic congestions. The bike sharing service is a type of collaborative consumption, which enables the optimization of resources and reduction of carbon footprint (DeMaio, 2009; Shu, Chou, Liu, Teo, & Wang, 2013). Further, as a means of exercise, cycling has become a fashion and increasingly popular transportation method in urban cities.

Despite the significant benefits of bike sharing systems, the daily operations of a large-scale bike sharing system for a high service level maintenance remain challenging and inefficient. The dynamics of human mobility often lead to bike supply and demand imbalance. Specifically, a customer may find a station empty when a bike is to be picked up, or find a station full when a bike is to be dropped off. The rebalancing operation has become one of the major cost for service provides to maintain the service level of bike stations. Thus, it is crucial to reallocate bikes among stations in a proactive and economical way. To this end, we study the multiple capacitated vehicle routing problem for bike system rebalancing optimization when the system is non-utilized and static (typically during midnight). Practically, system operators need to determine a daily schedule of bike reallocation among stations during midnight, considering the truck capacities and time constraints for such operations. During the rebalancing operations, it assumes that during the rebalancing operation time periods, the number of bikes at each station will not change, i.e., there is no exogenous demand (pick-ups or drop-offs by customers) during the rebalancing operations. This problem is critical for maintaining customer satisfaction, and thus finding an optimal solution to this problem holds a key to the success of bike sharing systems.

However, several major challenges have been observed to optimize bike rebalancing operations. The first prominent difficulty is the lack of accuracy in demand prediction. In order to determine the optimal rebalancing schedule, it is essential to decide the target inventory level (i.e., targeted number of bikes) at each station when the system resumes normal operation. Subsequently, accurate prediction of station-level pick-up and drop-off demand is desired, but remains technically challenging because of multiple impact factors, such as time, locations, weather conditions, and traffic situations. Most studies on bike demand prediction are based on historical demand average (Froehlich, Neumann, & Oliver, 2009) or model the system as a stochastic process with historical pick-up and drop-off rates (Schuijbroek, Hampshire, & van Hoeve, 2017). Recently, (Liu, Sun, Chen, & Xiong, 2016a) has shown that the impacts of other influential factors, such as meteorology reports and inter-station connections should not be neglected. To close this gap, this paper leverages a variety of data (multi-source data) directly or indirectly related to the public bike sharing service, so as to improve the bike demand prediction accuracy.

Secondly, once the rebalancing targets are determined, the remaining problem becomes a large-scale multiple capacitated vehicle routing problem (VRP) with loading and delivery operations. Bike rebalancing problems on small-scale networks (up to 100 stations) have been investigated by solving optimization models with the assumption that there exists at least one route covering all target stations (Dell'Amico, Hadjicostantinou, Iori, & Novellani, 2014). However, in practical problems, such as the ones considered in this paper, the network typically consists of hundreds of stations (up to 615 stations in our study), rendering the problem computationally challenging using traditional optimization algorithms. Furthermore, the problem becomes more complicated due to the existence of *outlier stations*, those having very large numbers of bikes to be relocated, typically exceeding truck capacities. These outlier stations may render the traditional route optimization models infeasible. Therefore, to tackle the computational issues incurred by the model size and model infeasibility, we propose a hierarchical optimization methodology which uses a Spatio-target Station Clustering Algorithm to decompose the large-scale multiple capacitated VRP into single-vehicle VRP problems with the consideration of outlier stations. Based on the clustering-first-optimization-second hierarchical method, we provide a partial target satisfying strategy to meet the part of rebalancing targets of outlier stations and a multiple vehicle visiting strategy to allow multiple vehilces to rebalance the inventory of a single station.

The remainder of this paper is organized as follows. Section 2.2 summarizes related literature. Section 2.3 presents the station-level bike demand prediction and the bike rebalancing optimization problem under study. Section 2.4 introduces the proposed bike pick-up and drop-off demand prediction models, and Section 2.5 provides the Mixed Integer Linear Programming formula and the hierarchical approach for solving the bike rebalancing optimization model. Section 2.6 presents the numerical results using real data from three major bike sharing systems. Finally, Section 2.7 concludes this paper.

2.2 Related Work

With the popularization of bike sharing systems around the world, there are increasing research interests in improving the efficiency of system utilization (Laporte, Meunier, & Wolfler Calvo, 2015). The related literature mainly focuses on system design, bike traffic demand analysis, and rebalancing operations.

Bike sharing system design. The design of bike sharing systems is critical for urban cities which have planned to adopt bike sharing systems, or to expand the service areas of their existing systems. (dell'Olio, Ibeas, & Moura, 2011) proposes a comprehensive framework for system implementation, including a prediction model for potential users' demand estimation and a location optimization model for station site selection. (Garca-Palomares, Gutirrez, & Latorre, 2012) uses a geographic information system (GIS) to determine the optimal bike station site locations. (Lin, Yang, & Chang, 2013) proposes a greedy heuristic approach to optimize bike sharing system design by providing an integrated view of transportation, inventory and facility costs, as well as service quality. (Freund, Henderson, & Shmoys, 2017) improves the bike sharing system design by considering dock capacity allocation. (Liu et al., 2015) and (Liu et al., 2017) investigate the station site optimization and service area expansion problems with the consideration of bike demand distribution and expected inventory unbalance costs. In this paper, we assume that the system design is fixed, and focus on rebalancing operations, to be reviewed in the sequel.

Bike demand prediction. Existing research on bike sharing systems focuses on studying spatial-temporal patterns of bike traffics, which discover the characteristics of bike flow distributions over a daytime (Gebhart & Noland, 2014; Corcoran, Li, Rohde, Charles-Edwards, & Mateo-Babiano, 2014; Zhou, 2015). (Singhvi et al., 2015) and (Faghih-Imani, Hampshire, Marla, & Eluru, 2017) build multi-factor statistical models for bike demand prediction based on linear mixed model and log-log regression models, respectively. The dynamics of bike demand imbalance for station inventory management is investigated by estimating station bike pick-up and drop-off rates and station inventory levels (Alvarez-Valdes, Belenguer, Benavent, Bermudez, Muoz, et al., 2016; Schuijbroek et al., 2017). Both (Liu et al., 2016a) and (Li, Zheng, Zhang, & Chen, 2015a) integrates the meteorology conditions as influential factors on bike demand forecasts. However, both of them ignores the recurrent dynamics of bike demand as time series.

Bike rebalancing optimization. In general, there are two approaches for rebalancing bikes in a network: by imposing user incentives and by using centralized rebalancing vehicles. For rebalancing the inherent asymmetry bike demand with minimum operational cost, (Kaspi, Raviv, & Tzur, 2014) explores bike reservation policies and suggests users to visit the least loaded stations. Reservations could be denied or the destinations could be diverted if no vacant docks were expected to be available at the original destinations. (Singla et al., 2015) and (Waserhole & Jost, 2016) present different dynamic pricing mechanisms that incentivize users to redistribute bikes by providing alternative rental prices. Although the proposed incentive schemes are promising in balancing system demands, the user participation rate is still low and hence existing bike sharing systems mostly rely on using rebalancing vehicles to reallocate station inventories.

Most studies on the bike rebalancing problem focus on minimizing the operational cost of the rebalancing vehicles, which is similar to the traveling salesman problem (Applegate, Bixby, Chvatal, & Cook, 2011) with additional constraints. Thus, it is an NP-hard problem and to find an exact solution remains challenging. (Erdoğan, Laporte, & Calvo, 2013) investigates a single-vehicle routing problem that allows the final bike inventory at each station to be between given lower and upper bounds. (Chemla, Meunier, & Calvo, 2013) presents a branch-and-cut procedure for the single-vehicle rebalancing problem, with numerical experiments on a system of up to 100 stations. (Erdoan, Battarra, & Calvo, 2015) develops an exact algorithm to compute the optimal route for the single-vehicle rebalancing problem. However, the instances with up to 60 stations for a single vehicle can take about 2 hours to find the optimal result.

For large-scale station networks, (Forma, Raviv, & Tzur, 2015) proposes a 3-step model for single-vehicle routing problems. The stations are first grouped into different clusters based on the geographic information and inventory capacity. The routing problem is solved within and between clusters. (Kloimüllner, Papazek, Hu, &

Raidl, 2015) proposes a logic-based Benders decomposition approach to maximize the number of stations to be rebalanced. (Schuijbroek et al., 2017) proposes a cluster-first route-second heuristic strategy with the assumption that the service level targets can always be satisfied within each cluster. However, in real-world problems, there exist many stations with extremely large rebalancing targets, which make the inner cluster route optimization infeasible. Actions have been taken to identify the outlier stations and ensure route optimization feasibility (Liu et al., 2016a), however, rebalancing those outlier stations remains problematic in operations.

Contributions. This paper contributes to the literature in the following aspects. First, the emergence of multi-source big data enables a new paradigm for enhancing bike sharing services. In this paper, we exploit fined-grained features that are related to bike demands from multi-source big data, including station-to-station bike transaction records, station status feed data, and hourly weather reports, for bike pick-up and drop-off demand prediction. Specifically, we propose a nonlinear autoregressive network with exogenous meteorology factors (NARX) model to predict the bike pick-up demand during the day at station level. The drop-off demand at each station is predicted based on our proposed pick-drop bike transition (PDBT) predictor which discover trip transition patterns and simulates the station to station bike transition probabilities and trip durations. In addition, we testify our prediction models using real-world bike sharing system data. This in turn enables a practical end-to-end solution for the bike rebalancing problem.

Furthermore, a general mixed integer linear programming (MILP) model is proposed for the multiple capacitated VRP problem with an objective of minimizing traveling distance and unsatisfied rebalancing targets. In order to deal with the large-scale rebalancing problem with outlier stations, we propose two hierarchical optimization strategies, which extend the outlier removal strategy developed in (Liu et al., 2016a). More specifically, the partial rebalancing targets satisfying strategy based on a Spatio-target Station Clustering algorithm considers the objective of minimizing unsatisfied rebalancing targets within clusters, so as to partially rebalance the outlier stations. In addition, the multiple vehicle visiting strategy supports that a single station can be covered by multiple vehicles based on a split-node Spatial-target Clustering. After the station clustering, the multi-vehicle routing problem is decomposed into multiple single-vehicle routing problems, which are much more tractable. As such, we can solve very large-scale rebalancing problems efficiently and effectively. It provides an alternative data-driven decomposition approach to traditional mathematical decomposition for large-scale optimization problems.

2.3 Problem Formulation

In this section, we first provide notation and definitions. Then, we introduce the two-stage bike rebalancing problem, including a station-level bike demand prediction problem and a rebalancing operations optimization problem.

2.3.1 Notation and Definitions

Bike Station Network

A bike station network is represented by a directed graph $G = (S, E)$, where S is the set of nodes, each representing a station, and E is the set of directed edges, each connecting a pair of stations. For stations $s_i, s_j \in S$, $e_{ij} = (s_i, s_j) \in E$ represents the directed edge from station s_i to station s_j. The station network is constructed by tracking a set of trip records. Here, $tr = (s_o, s_d, \tau_o, \tau_d)$ is a bike trip record from an origin station s_o to a destination station s_d, where τ_o is the pick-up time and τ_d is the drop-off time. Note that, in our data preprocessing, the records with trip duration $\tau_d - \tau_o$ shorter than 1 minute are treated as anomalies and filtered.

Station Pick-up and Drop-off Demand

The bike demand at each station is defined as the pick-up (drop-off) frequency per unit time when there is no lost demand; that is, there are bikes for pick-up (or available docks for drop-off). A station becomes unavailable when it is under maintenance, on a blocked street, has no bikes for pick-ups (unavailable for pick-up) or has no available dock for drop-offs (unavailable for drop-off). Since each station may have certain unavailable periods (i.e., demand lost due to an empty/full station), historical demand does not accurately capture the true demand. Hence, we use the expected pick-up (drop-off) rate to describe the true pick-up (drop-off) demand, which is formally defined below.

The historical daily bike demand is first divided into hourly time slots, with $t \in \{0, 1, ..., 23\}$. For station i and time slot t, let $pf_i(t)$ and $pa_i(t)$ be the actual pick-up frequency (i.e., number of bikes picked-up) and pick-up available duration (effective time when there are bikes available for pick-up), respectively. The station *pick-up demand*, $pd_i(t)$, is then defined in Eq (2.1). That is, the pick-up demand is augmented to take into account the demand lost when customers arrived but found no bikes available for pick-up. In a similar vein, the actual drop-off frequency and drop-off available duration for station i in time slot t are denoted by $df_i(t)$ and $da_i(t)$, respectively. The *drop-off demand* is then defined as the expected drop-off rate during the drop-off available duration, shown in Eq (2.2).

$$pd_i(t) = \frac{pf_i(t)}{pa_i(t)} \tag{2.1}$$

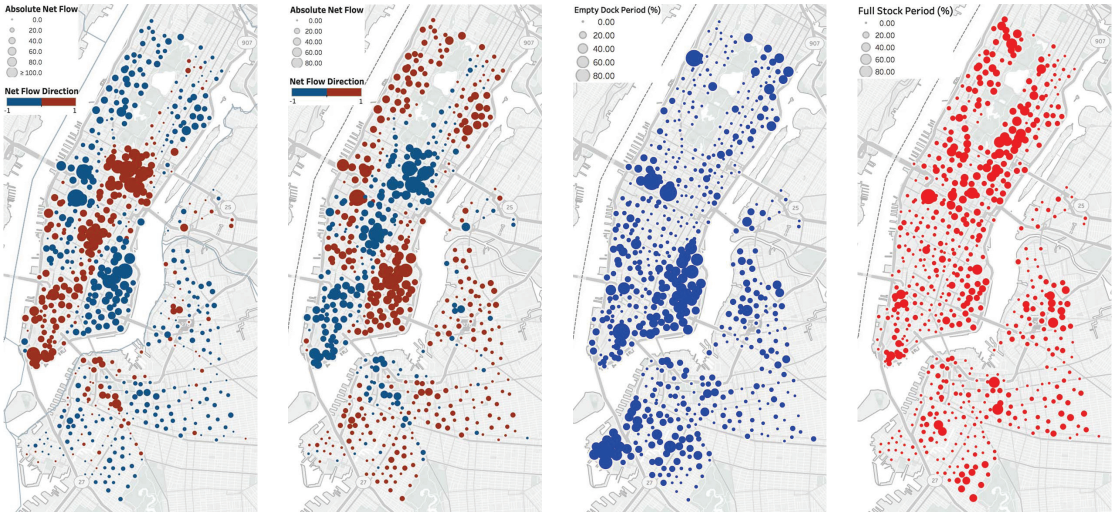
$$dd_i(t) = \frac{df_i(t)}{da_i(t)} \tag{2.2}$$

Subsequently, we define the bike net (incoming) flow, $nf_i(t)$, as follows:

$$nf_i(t) = dd_i(t) - pd_i(t) \tag{2.3}$$

As illustrative examples, the average net flows for stations in NYC Citi Bike system from August 2016 to July 2017 during the morning period (6 am – 10 am) and the afternoon period (5 pm – 9 pm) are shown respectively in Figures 2.1(a) and 2.1(b). In these two figures, each dot represents a bike station with its size indicating the absolute value of the net flow. The red color indicates a positive net flow (i.e., drop-off demand is higher than the pick-up demand), and the blue color represents a negative net flow. It is seen that the station net flow distribution is unbalanced both geographically and temporally. A large positive net flow usually results in a full station status, while a large negative net flow is usually followed by an empty station status. Further, Figures 2.1(c) and 2.1(d) show the daily averages of time percentages for a station being empty and full, respectively, over the same time period.

It is observed that many stations have a low availability percentage, and hence a low service level.



(a) Net flow (6am–10am) (b) Net flow (5pm–9pm) (c) Empty Stock (%) (d) Full Stock (%)

Figure 2.1. NYC Citi Bike station net flow and availability (August 2016 – July 2017)

2.3.2 Problem Description

In this paper, we aim to provide an end-to-end solution for the (static) bike rebalancing problem. To this end, two technical components are needed: (1) an accurate prediction of station-level bike demand, which will be used to determine the rebalancing target (i.e., number of bikes to be reallocated) for each station; and (2) a fast and robust optimization approach for the routing of rebalancing vehicles.

Bike Demand Prediction

Given a set of bike trip records $\{tr\}$ and a set of meteorology reports $\{R\}$, the problem of bike demand prediction is to forecast the future pick-up (drop-off) demand $pd_i(t)$ ($dd_i(t)$) of each station as defined in Section 2.3.1. The main challenge here is to fully utilize the information provided by multi-source data, so as to improve the prediction accuracy.

Once the bike demand is determined, the hourly net flow of each station can be calculated using Eq. (2.3). Given the initial number of bikes I_i and the (predicted) bike net flow $nf_i(t)$ at station i, the rebalancing target rt_i is defined as the total number of bikes to be dropped-off or picked-up by rebalancing (dispatching) vehicles. Note that $rt_i < 0$ indicates the need for pick-ups and $rt_i > 0$ indicates the need for drop-offs. If the rebalancing target rt_i is 0, it indicates that the station is self-balanced, as the initial inventory I_i can provide sufficient bikes and available docks throughout the day. Note further that, in our problem, the rebalancing is performed only once before the system resumes its operation every day. Thus, the optimal rebalancing target for each station is the one that maximizes the duration when the station remains in-service since the system starts its operation. Formally, for station i, we first compute the set of rt_i (denoted by Θ_i) that maximizes the station in-service duration, T:

$$\Theta_i = \underset{-I_i \leq rt_i \leq SC_i - I_i}{\arg\max} \left\{ T : 0 \leq I_i + rt_i + \sum_{t=0}^{T} nf_i(t) \leq SC_i \right\} \tag{2.4}$$

where SC_i is the capacity (i.e., number of docks) of station i. Then, the optimal rebalancing target of station i, rt_i^*, is chosen from the set Θ_i to be the one with the minimum absolute value. Ties can be broken arbitrarily.

Bike Rebalancing Operation Optimization

Once the station rebalancing targets are computed, the next stage is to optimize the rebalancing operations. Specifically, a multiple capacitated vehicle routing problem with additional objective and constraints needs to be solved, where its sets, parameters and decision variables are listed below.

Sets	
$\mathcal{V}$	Set of rebalancing vehicles
$\mathcal{N}$	Set of stations
$\mathcal{N}_v$	Set of stations covered by vehicle v, $v \in \mathcal{V}$
$\mathcal{Q}$	Set of outlier stations, $\mathcal{Q} \subset \mathcal{N}$
$\mathcal{D}$	Set of depots (starting and ending point of each vehicle)
$\mathcal{N}_0$	Set of all nodes, $\mathcal{N}_0 = \mathcal{N} \cup \mathcal{D}$

Parameters		
TC_{ij}	$i, j \in \mathcal{N}_0$	Travel distance from station i to j
rt_i	$i \in \mathcal{N}$	Rebalancing target of station i, computed from Eq. (2.4)
SC_i	$i \in \mathcal{N}$	Station capacity of station i
I_i	$i \in \mathcal{N}$	Initial inventory level of station i
C		Vehicle capacity limit
M		A positive large number
λ		Penalty for each unit of unsatisfied rebalancing target

Variables		
$x_{vij} \in \{0, 1\}$	$v \in \mathcal{V}, i, j \in \mathcal{N}_0$	Binary variables; x_{vij} equals 1 if vehicle v travels directly from station i to station$j$, and 0 otherwise
$U_i \in \mathbb{Z}_{\geq 0}$	$i \in \mathcal{N}$	Unsatisfied rebalancing target at station i
$ro_{vi} \in \mathbb{Z}$	$v \in \mathcal{V}, i \in \mathcal{N}$	Number of bikes reallocated in station i by vehicle v; a positive value of ro_{vi} indicates a drop-off value and a negative indicates a pick-up
$y_{vi} \in \mathbb{Z}_{\geq 0}$	$v \in \mathcal{V}, i \in \mathcal{N}_0$	Number of bikes carried by vehicle v after leaving station i

The objective of the routing optimization is to minimize the total traveling distance of rebalancing vehicles and the unsatisfied rebalancing targets of all stations. The MILP model for this problem is formulated as follows, denoted by v-MILP, where

v is the number of rebalancing vehicles.

$$\min \quad \mathcal{F}_1(\mathbf{x}) = \sum_{v\in\mathcal{V}}\sum_{i\in\mathcal{N}_0}\sum_{j\in\mathcal{N}_0} TC_{ij}x_{vij} + \lambda\sum_{i\in\mathcal{N}} U_i \tag{2.5}$$

$$\text{s.t.} \quad -(rt_i - \sum_{v\in\mathcal{V}} ro_{vi}) \leq U_i \qquad \forall i \in \mathcal{N} \tag{2.6}$$

$$rt_i - \sum_{v\in\mathcal{V}} ro_{vi} \leq U_i \qquad \forall i \in \mathcal{N} \tag{2.7}$$

$$y_{vi} - ro_{vj} \geq y_{vj} - (1 - x_{vij})M \qquad \forall i \in \mathcal{N}_0, \forall j \in \mathcal{N}_0, \forall v \in \mathcal{V} \tag{2.8}$$

$$y_{vi} - ro_{vj} \leq y_{vj} + (1 - x_{vij})M \qquad \forall i \in \mathcal{N}_0, \forall j \in \mathcal{N}_0, \forall v \in \mathcal{V} \tag{2.9}$$

$$y_{vi} \leq C \qquad \forall v \in \mathcal{V}, \forall i \in \mathcal{N}_0 \tag{2.10}$$

$$\sum_{j\in\mathcal{N}} x_{vij} = \sum_{j\in\mathcal{N}} x_{vji}, \qquad \forall v \in \mathcal{V}, \forall i \in \mathcal{N} \tag{2.11}$$

$$\sum_{v\in\mathcal{V}}\sum_{j\in\mathcal{N}_0} x_{vij} \leq 1 \qquad \forall i \in \mathcal{N} \tag{2.12}$$

$$\sum_{v\in\mathcal{V}}\sum_{j\in\mathcal{N}_0} x_{vji} \leq 1 \qquad \forall i \in \mathcal{N} \tag{2.13}$$

$$\sum_{v\in\mathcal{V}} ro_{vi} \leq SC_i - I_i \qquad \forall i \in \mathcal{N} \tag{2.14}$$

$$\sum_{v \in \mathcal{V}} ro_{vi} \geq -I_i \qquad \forall i \in \mathcal{N} \tag{2.15}$$

$$ro_{vi} + M \sum_{j \in \mathcal{N}_0} x_{vij} \geq 0 \qquad \forall v \in \mathcal{V}, i \in \mathcal{N} \tag{2.16}$$

$$ro_{vi} - M \sum_{j \in \mathcal{N}_0} x_{vij} \leq 0 \qquad \forall v \in \mathcal{V}, i \in \mathcal{N} \tag{2.17}$$

$$\sum_{i \in \mathcal{D}} \sum_{j \in \mathcal{N}} x_{vij} = 1 \qquad \forall v \in \mathcal{V} \tag{2.18}$$

$$\sum_{i \in \mathcal{D}} \sum_{j \in \mathcal{N}} x_{vji} = 1 \qquad \forall v \in \mathcal{V} \tag{2.19}$$

$$\sum_{i \in \mathcal{S}_v} \sum_{j \in \mathcal{S}_v} x_{vij} \leq |\mathcal{S}_v| - 1 \qquad \forall v \in \mathcal{V}, \forall \mathcal{S}_v \subset \mathcal{N}, \mathcal{S}_v \neq \emptyset \tag{2.20}$$

$$y_{vi} \in \mathbb{Z}_{\geq 0} \qquad \forall v \in \mathcal{V}, \forall i \in \mathcal{N} \tag{2.21}$$

$$x_{vij} \in \{0, 1\} \qquad \forall v \in \mathcal{V}, i, j \in \mathcal{N} \tag{2.22}$$

$$ro_{vi} \in \mathbb{Z} \qquad \forall v \in \mathcal{V}, \forall i \in N \tag{2.23}$$

$$U_i \in \mathbb{Z}_{\geq 0} \qquad \forall i \in \mathcal{N} \tag{2.24}$$

The objective (2.5) is to minimize the total transportation cost and unsatisfied rebalancing targets. Constraints (2.6) and (2.7) define the unsatisfied rebalancing target at station i as $U_i = |rt_i - \sum_{v \in \mathcal{V}} ro_{vi}|$. Constraints (2.8) and (2.9) are the bike flow conservation constraints. Constraint (2.10) sets the vehicle capacity limit. Constraint (2.11) indicates the route continuity. Constraints (2.12) and (2.13) mean that a vehicle can visit a station no more than once. Constraints (2.14) and (2.15) specify that the number of reallocated bikes at station i cannot exceed its capacity (for drop-off) or station initial inventory (for pick-up). Constraints (2.16) and (2.17) indicate that if vehicle v does not visit station i, its operation at station i is 0.

Constraints (2.18) and (2.19) indicate that each rebalancing vehicle must start and end its trip at a depot. Constraint (2.20) is the subtour elimination constraint (SEC) that removes illegal subtours in the form of circular paths (Wolsey, 1998). The SECs grow exponentially with the number of stations. Instead of explicitly including the SECs in our MILP model, we generate and add theses SECs to our model implicitly as *lazy constraints* (Aguayo, Sarin, & Sherali, 2018). Different from normal constraints which are generated in advance, lazy constraints are activated when a feasible solution is found while it violated one or multiple SECs. Specifically, there is no SEC in the initial model. Once a feasible solution is found, all complete tours are checked. If there is a tour that does not go through any depots and the tour length equals to the number of stations visited by the tour, we add the station set and its corresponding SEC to our model so as to eliminate this subtour. In our implementation, we apply the callback function from Gurobi 6.5.0 Optimizer (Gurobi Optimization, 2016) to detect and activate SECs.

Although the Gurobi MILP solver has become one of the industry standards in terms of computational speed and solution quality, the general v-MILP model is still intractable for large-scale problems. In order to improve the computational efficiency, we develop a clustering-first optimization-second strategy to reduce the optimization complexity in Section 2.5.

2.4 Bike Demand Prediction Model

In this section, we provide the technical details of the proposed meteorology similarity weighted KNN (MSWK) statin bike pick-up demand prediction and nonlinear autoregressive network with exogenous meteorology factors (NARX) and pick-drop bike transition (PDBT) predictor for station-level bike demand prediction. Business days and non-business days (including Federal Holidays) are treated separately for training and testing.

2.4.1 MSWK Station Bike Pick-up Prediction

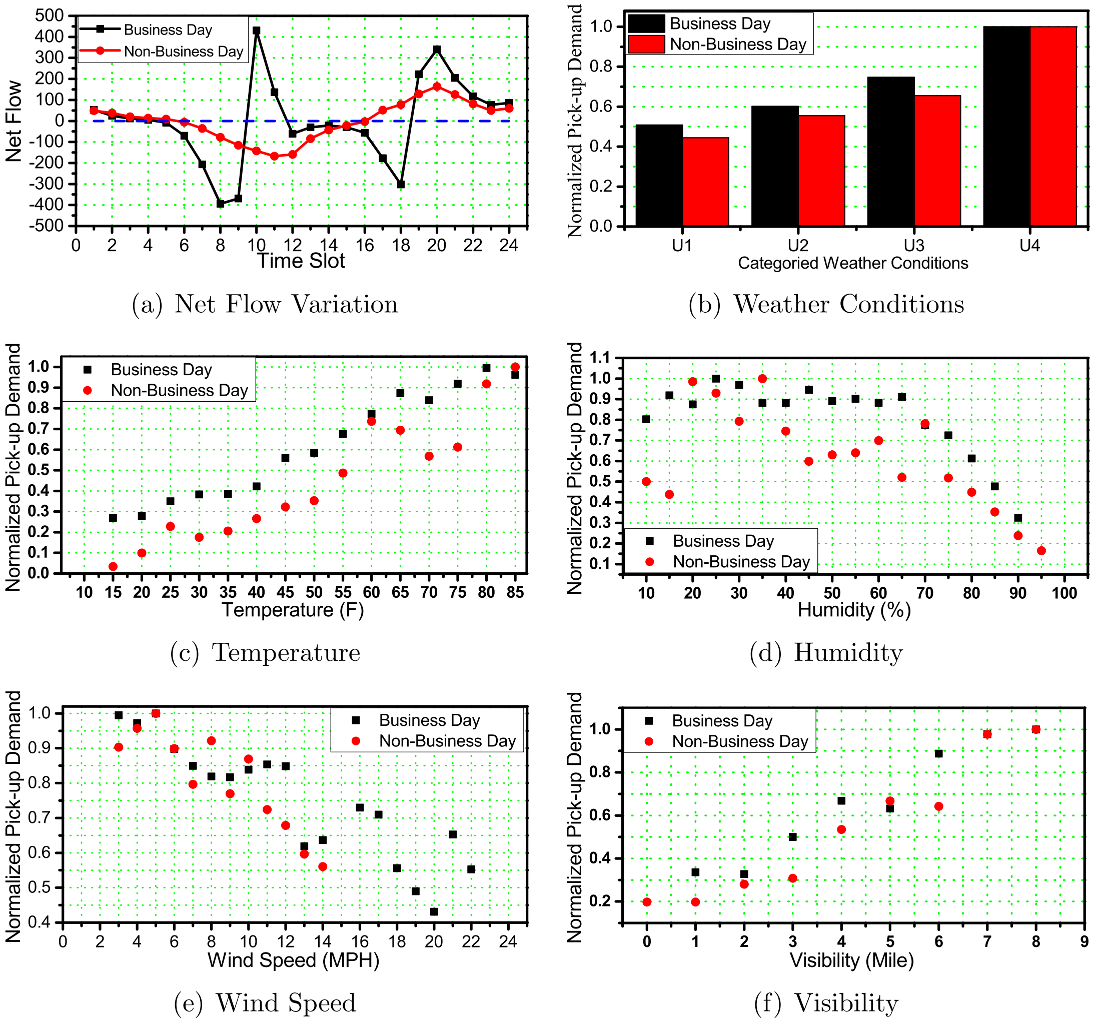
The MSWK regressor is built to predict the station level bike pick-up demand $s_i.pd(D^t)$ during time slot t of any given day D that is on the basis of a meteorology multi-similarity function.

Similarity measurement

Given the weather reports R_D^t of each time slot t, which contains weather condition $W_{D_p^t}$ (sunny, raining, etc.), temperature $F_{D_p^t}$, humidity $H_{D_p^t}$, wind speed $S_{D_p^t}$ and visibility $V_{D_p^t}$ of time slot t on day D_p, the similarity between 2 different days D_p^t and D_q^t is calculated as the linear combination of three units: weather similarity, temperature similarity and humidity-wind speed-visibility similarity. Each unit is associated with an effective coefficient a that is learned to improve the prediction accuracy.

Weather similarity. The weather conditions are first manually segmented into different levels according to their suitability for outdoor bicycling (see Figure 2.2(b)): ((heavy snowy, heavy rainy), (snowy, rainy), (hazy, foggy), (clear, cloudy))=(1, 0.75, 0.5, 0.25). Then the weather similarity is defined as follows:

$$\lambda_1(W_{D_p^t}, W_{D_q^t}) = \frac{1}{2\pi\sigma_1} e^{-\frac{(W_{D_p^t} - W_{D_q^t})^2}{\sigma_1^2}} \tag{2.25}$$



(a) Net Flow Variation

(b) Weather Conditions

(c) Temperature

(d) Humidity

(e) Wind Speed

(f) Visibility

Figure 2.2. Demand net flow (a) and the effect of multiple factors on bike pick-up demand (b)-(f).

Temperature similarity. As can be seen from Figure 2.2(c), the bike pick-up demand is sensitive to the temperature, especially when temperature is below 47 F during the weekends. We extract the temperature information and calculate its similarity based on a Gaussian Kernel function:

$$\lambda_2(F^t_{D_p}, F^t_{D_q}) = \frac{1}{2\pi\sigma_2} e^{-\frac{(F_{D_p^t} - F_{D_q^t})^2}{\sigma_2^2}} \tag{2.26}$$

Humidity, wind speed and visibility similarity. Different from the effect of temperature, the humidity, wind speed and visibility affect bike pick-up demand with similar effects (see Figure 2.2(d)-2.2(f)). We choose a 3-D Gaussian Kernel to calculate the similarity between $(H_{D_p^t}, S_{D_p^t}, V_{D_p^t})$ and $(H_{D_q^t}, S_{D_q^t}, V_{D_q^t})$:

$$\lambda_3 = \frac{1}{2\pi\sigma} e^{-(\frac{(H_{D_p^t} - H_{D_q^t})^2}{\sigma_3^2} + \frac{(S_{D_p^t} - S_{D_q^t})^2}{\sigma_4^2} + \frac{(V_{D_p^t} - V_{D_q^t})^2}{\sigma_5^2})} \tag{2.27}$$

Similarity function. To uniform these similarity calculations, we normalize the temperature, humidity, wind speed and visibility into range $[0, 1]$ and simplify equation (2.25)-(2.27) by setting $\sigma_k = 1$ $(k = 1, 2, 3, 4, 5)$. The similarity function is then defined as a linear combination of λ:

$$M(D^t_p, D^t_q; a) = \delta_w(D_p, D_q) \sum_{i=1}^{3} a_i \lambda_i \tag{2.28}$$

where $\delta_w(D_p, D_q)$ is the delta function. $\delta_w(D_p, D_q) = 1$ if D_p and D_q are both weekdays or weekends, otherwise $\delta_w(D_p, D_q) = 0$.

MSWK learning

Given K and a, we select top K days $\{D_1^t, D_2^t, ..., D_K^t\}$ with the highest similarity to our target day D_n^t according to the similarity function. Then the $s_i.pd(D_q^t)$ is predicted by a similarity weighed KNN:

$$s_i.pd(D_q^t; a) = \frac{\sum_{p=1}^{K} M(D_p^t, D_q^t; a) s_i.pd(D_p^t)}{\sum_{p=1}^{K} M(D_p^t, D_q^t; a)} \tag{2.29}$$

The weight of different similarity function a in equation (2.28) is trained to reach the minimum prediction absolute error of predicted value $\hat{s}_i.pd(D_q^t; a)$ and ground truth $s_i.pd(D_q^t)$ by brute force searching:

$$a* = \arg\min_{a} \frac{1}{N} \sum_{i=1}^{N} |\hat{s}_i.pd(D_q^t; a) - s_i.pd(D_q^t)| \tag{2.30}$$

2.4.2 NARX for Pick-up Demand Prediction

We propose a NARX model to predict the pick-up demand for next time slots t based on the meteorology reports at time t and previous 24 time slots of pick-up demands and drop-off demands. NARX, as a special case of recurrent neural network models, is developed to build complex nonlinear relationships between the prediction tartget and exogenous from different domains. It also has a feedback which comes from the output neuron rather than from hidden states to build time dependencies. The general architecture of our NARX model is shown in Figure 2.3 and the details of the specification and estimation are summarized below:

Input Layer. The input layer has 53 factors including weather conditions, temper-

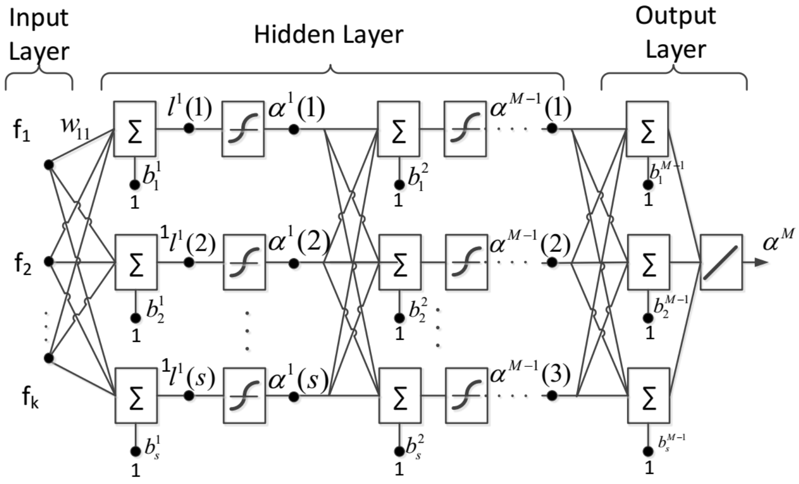


Figure 2.3. Architecture of NARX model

ature, humidity, windspeed, and visibility at time slot t; 24 pick-up demands and 24 drop-off demands from time $t-25$ to $t-1$.

Hidden Layer Input. The input of unit i in hidden layer $k+1$ is the linear combinations of the outputs α^k of units in layer k. Since the features are of different range scales, they are standardrized with a mean of 0 and standard deviation of 1:

$$\alpha^0(i) = f_i \tag{2.31}$$

$$l^{k+1}(i) = \sum_{j=1}^{S_k} w_{ji}^{k+1}\alpha^k(j) + b_i^{k+1}(i) \tag{2.32}$$

Layer Output. We use a sigmoid activation function to map a unit input to its output which is computationally efficient to implement:

$$a^{k+1}(i) = \frac{1}{1+e^{-l^{k+1}}} \tag{2.33}$$

The output layer is a linear layer for regression problem of station bike demand prediction and the final output a^M is t_{sd}.

Training Algorithm. We use the Sum Squared Error as our training objective:

$$SSE = \frac{1}{2}\sum_{i=1}^{nt}(t_i - a_i^M)^2$$

The Levenberg-Marquardt algorithm, which is proved to be one of the most efficient way for least squares curve fitting problems, is applied for our NARX model traing with sum of squared errors objective. Moreover, a validation set is used for monitoring validation error during the training iterations. When the validation error begin to rise after a few iterations, the training process will stop. The optimal paramer of our NARX model are chosen at the iteration which reaches the minimum validation error.

2.4.3 PDBT for Bike Drop-off Demand Prediction

Historical data shows that drop-off demand has a strong dependency on surrounding pick-up events and trip durations. The proposed PDBT predictor simulates the probability that a picked-up bike from station i in time slot t will be dropped-off at station j in a future time slots t'. Specifically, the drop-off demand at station j during time slot t, denoted as $dd_j(t)$, can be estimated as follows:

$$dd_j(t) = \sum_{i \in \mathcal{N}: i \neq j} \sum_{\Delta \geq 0} e_{ij}^{t-\Delta} P_{ij}^{\Delta} \tag{2.34}$$

where Δ denotes the number of time slots between a pick-up and drop-off event. e_{ij}^{Δ} denotes the number of bikes picked-up in station i during time slot $t - \Delta$ and dropped-off in station j after passing Δ time slots, and P_{ij}^{Δ} denotes the probability that a bike picked-up in station i is dropped-off in station j after passing Δ time slots. Thus, the term in Eq. (2.34) represents the estimated number of bikes dropped-off during the time slot t.

Next, we show how to estimate the values on the right-hand side of Eq. (2.34). Given bike pick-up demand in station i during time slot $t - \Delta$, $pd_i(t - \Delta)$, which we have estimated in Section 2.4.2, $e_{ij}^{t-\Delta}$ can be estimated from trip history records as follows:

$$e_{ij}^{t-\Delta} = pd_i(t - \Delta)\frac{ef_{ij}}{pd_i} \tag{2.35}$$

where pd_i is the daily average pick-up demand at station i and ef_{ij} is the daily average number of trips from station i to station j.

To estimate the second term, P_{ij}^{Δ}, we first show three typical patterns of such trip durations. The dots in Figure 2.4 are the data points of trip durations in a typical day for three station to station connections for each bike system. The trip transition patterns are summarized as follows:

- Most trips from station 3141 to 3140 in NYC, from station 115 to 153 in Chicago, and from station 14 to 32 (shown in small dot markers) are *commuter trips*, that is, the trip durations are very close to the route recommended by Google Maps for commuters (see the second column of Table 2.2).
- Most trips from station 514 to 3256 in NYC, from station 25 to 35 in Chicago, and from station 67 to 60 in Boston (shown in square dot markers) are *tourist and exerciser trips*, as their routes are much longer than the shortest route

between the pick-up and drop-off stations.

- Trips from station 3137 to 3144 in NYC, from station 6 to 76 in Chicago, and from station 47 to 98 in Boston (shown in diamond markers) are *mixed trips*, that is, a portion of them are commuter trips, and the rests are tourist and exerciser trips.

Thus, we propose to depict the trip durations between station i and j, denoted as h, using the following two-peak Gaussian function, which is capable of capturing all three patterns above. The two peaks represent two different human mobility patterns: commuting and hanging out, which is related to the point of interests distribution in urban cities. For example, for the trips connecting stations located in residential areas and business areas, we may find more bike users as commuters.

$$D_{ij}(h) = a_1 e^{-(\frac{h-\mu_1}{\sigma_1})^2} + a_2 e^{-(\frac{h-\mu_2}{\sigma_2})^2} \tag{2.36}$$

where a_i, μ_i and σ_i $(i = 1, 2)$ are fitted from data. As illustrative examples, the curves in Figure 2.4 and their parameters in Table 2.2 show the fitted two-peak Gaussian functions for the data points in Figure 2.4. More specifically, for the three patterns, we have the following curves.

- For commuter trips, the curves in black color are the fitted two-peak Gaussian functions. Note that, since most trips are commuter trips (shortest trips between two stations), two peaks are very close and hence almost overlap in the figure.

- For tourist and exerciser trips, the curves in red color are the fitted two-peak

Gaussian functions. Similarly, two peaks almost overlap in the figure, since most trips are tourist and exerciser trips.

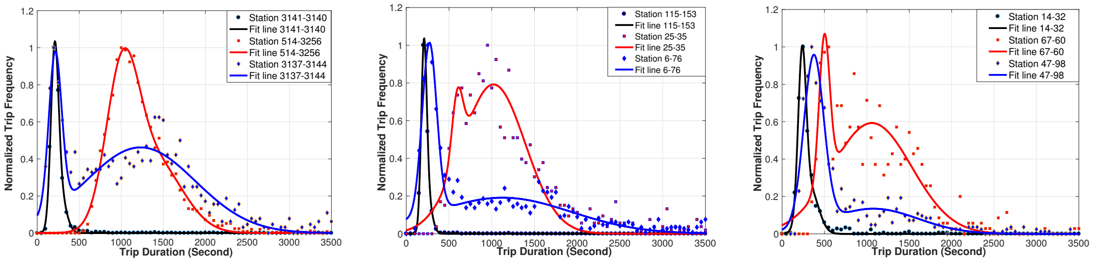
- For mixed trips, the curves in blue color clear show two peaks of the Gaussian functions.

Further, let t_0 represent the bike pick-up time relative to the start time of each time slot, e.g., $t_0 = 20$ if a bike is picked up at 10:20 am (20 minutes in time slot 10am-11am), and let $|t|$ be the length of time slot t. The probability that a bike picked-up at station i will arrive at station j in the same time slot is calculated as follows:

$$\begin{aligned} P_{ij}^0 &= P(t_0 + h \leq |t|) \\ &= \int_0^{|t|} P(h \leq |t| - t_0 | t_0) p(t_0) dt_0 \\ &= \frac{1}{|t|} \int_0^{|t|} \int_0^{|t|-t_0} D_{ij}(h) dh dt_0 \end{aligned} \tag{2.37}$$

Here, the trip duration h follows the two-peak Gaussian function in Eq. (2.36), and we assume that the pick-up times in time slot t follow uniform distribution $(p(t_0) \sim U(0, |t|))$.

Recall that the probability of trip duration exceeding 1 hour is extremely low, therefore, a drop-off should happen in the same time slot or the next time slot as the pick-up. That is, $P_{ij}^0 + P_{ij}^1 \approx 1$.



(a) NYC CitiBike (b) Chicago Divvy (c) Boston Hubway

Figure 2.4. Trip duration histograms of different pick-drop bike trainsition patterns

Table 2.2. Examples of fitting results of $D_{ij}(h)$ and estimations of P^0_{ij} and P^1_{ij}

System	Edges	Map	μ_1 (95% CI)	μ_2 (95% CI)	R^2	SSE	P^0_{ij}	P^1_{ij}
Citi	3141-3140	300	208(204,211)	284(204,368)	0.9983	0.0031	0.91	0.09
	514-3256	720	1013(999,1026)	1389(1211,1576)	0.9932	0.0048	0.76	0.24
	3137-3144	240	214(204,225)	1230(1165,1294)	0.9103	0.4143	0.83	0.17
Divvy	115-153	240	207(204,210)	263(197,330)	0.9999	0.0001	0.92	0.08
	25-35	600	597(570,625)	1020(981,1059)	0.9139	0.5857	0.71	0.29
	6-76	240	270(264,278)	1127(964,1289)	0.9348	0.2052	0.81	0.19
Hubway	14-32	240	236(235,238)	329(304,353)	0.9957	0.0081	0.89	0.11
	67-60	540	499(486,513)	1059(998,1120)	0.8868	0.7468	0.78	0.22
	47-98	360	374(365,383)	1077(876,1277)	0.945	0.2045	0.85	0.15

2.5 Hierarchical Optimization for Rebalancing Operations

In this section, we propose two clustering-first optimization-second algorithms that can efficiently solve the v-MILP model for large-scale instances that are intractable using commercial solvers. We mention that, although the algorithm is proposed to solve the bike rebalancing operations optimization, the idea of clustering-based decomposition is applicable for solving other NP-hard problems with similar structures.

Clustering algorithms have been used in data mining problems to group instances with similar patterns. However, the potential of using clustering algorithms to decompose large-scale optimization problems has not been fully explored. Further, as pointed out by (Liu et al., 2016a), another challenge of solving the bike rebalancing operations optimization problem defined in Section 2.3 is the existence of outlier stations, namely, the stations with very large rebalancing targets exceeding the vehicle capacity. (Liu et al., 2016a) proposed an outlier-removal strategy to detect and filter these outlier stations from the optimization problem, rendering the problem partially unsolved and leading to sub-optimal solutions. To achieve better optimality, in this paper, we propose two hierarchical optimization algorithms to be described in details next.

2.5.1 Station Clustering

The implementation of clustering algorithms for bike rebalancing optimization problem has two challenges: 1) not only distances between stations should be taken into account, the station inventory target should also be considered for inventory constrains; 2) outlier stations should be discovered to guarantee inner cluster feasible routes. Although constrained clustering algorithms have been detailed studied, the constrained conditions in previous studies are mainly in the manner of *must-link* or *cannot link* pairs under the name of semi-supervised clustering (Wagstaff, Cardie, Rogers, Schrödl, et al., 2001; Basu, Bilenko, & Mooney, 2004). In this problem, whether two stations belong to the same cluster is not determined by themselves, but is affected by the total balance of stations in the same cluster. To the best of our knowledge, such kind of constrained clustering algorithm has not been studied ever before. In this paper, we propose a constrained K-Centers Clustering algorithm for bike stations to fill the research gap.

Algorithm 1 presents the proposed algorithm. It begins with an initial center set E, and assigns each station to its nearest stations. Then for a cluster, if the

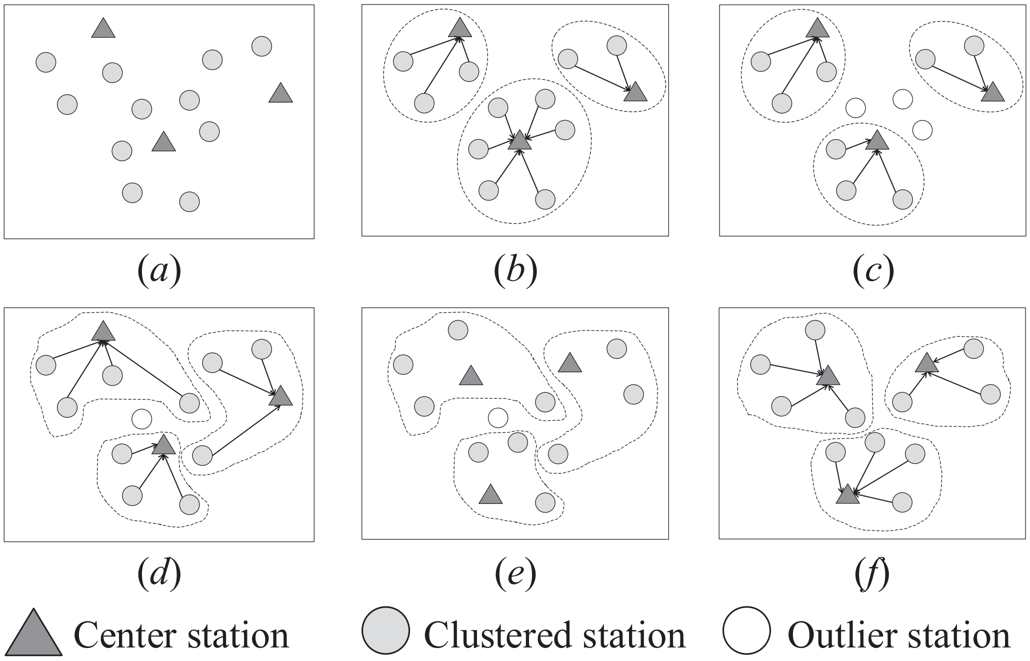


Figure 2.5. Constrained K-Centers Clustering procedure illustration.

balance condition is not satisfied ($B(C_k) > VC$, where $B(C_k) = |\sum_{i\in C_k} b_i|$), we pick some stations out of the cluster. The stations, which are able to reduce the total balance of the cluster and close to other centers, are firstly picked out. In step 7∼11, the unlabeled stations are assigned with new cluster label. For each unlabeled station, the new cluster label is determined by the total balance of a cluster and the distance between the station and its nearest station in the cluster. The unlabeled outlier stations that are far from cluster centers are preferentially processed. This step ensures these outliers scattered at the central region of the studied area, and can be easily covered by other clusters. After adjusting clustering result according to balance conditions, new centers are selected in Step 12∼13. Step 1∼13 are iterated until convergence (centers are unchanged). Step 15 outputs the clustering result.

Figure 2.5 presents a toy example of the capacity constrained K-centers algorithm.

```
1 Input: $TD_{ij}$, $b_i$, $VN$, $VC$, $\delta$, $E$;
2 Output: c;
  1: for all stations do
  2:     $c(i) = \arg\min_{j\in E}(TD_{ij})$;
  3: for each cluster $C_k = \{i|c(i) = E(k)\}$, if $B(C_k) > VC$ do
  4:     while $B(C_k) > VC$ do
  5:         $q = \arg\min_{i\in C_k^*}(\sum_{j\in E} TD_{ij})$, where $C_k^* = \{i|B(C_k \setminus i) < B(C_k)\}$;
  6:         $c(q) = 0$,$C_k = C_k \setminus q$;
  7: find $l = \arg\max_{i,c(i)=0}(\sum_{j\in E} TD_{ij})$
  8:     if $\exists\ S_l = \{k|B(C_k \bigcup l) < VC, \min_{j\in C_k} TD_{lj} < \delta\}$
  9:     then do $c(l) = E(k)$, $k = \arg\min_{q\in S_l}(\min_{j\in C_q} TD_{lj})$
 10:     else do $c(l) = -1$;
 11: go to step 7 till $\nexists\ i$ that $c(i) = 0$;
 12: for each cluster $C_k$
 13:     $E'(k) = \arg\min_{i,c(i)=E_k} \sum_{j,c(j)=E_k} TD_{ij}$;
 14: if $E' \neq E$ then $E = E'$ go to step 1, else go to step 15;
 15: return clustering result c.
```

Algorithm 1: $CCKC(\mathbf{TD},\mathbf{b},VN,VC,\delta,E)$

Fourteen stations are given in Figure 2.5(a), and three stations are marked as initial centers by triangles. All other stations are assigned to their nearest centers in (b). Capacity condition is examined in (c). If the capacity of a cluster is over the vehicle capacity, then some stations are pressed out as temporary outliers. Stations near other centers are preferentially excluded as they are much easier to be visited by vehicles from other clusters. The outliers are assigned to other centers with respect to cluster capacity and traveling distance in (d). New centers are generated according to the current clustering result in (e) and stations are assigned to new centers in (f). This procedure is repeated till convergence.

The clustering result obtained by Algorithm 1 has the following features: 1) the total balance of a cluster is under the capacity of vehicle; 2) stations in a same cluster are close to each other; 3) clusters are overlapped, their boundaries are not discriminative; 4) outliers (if any) are at the central region of the studied area. All the features mentioned above are very helpful for designing bicycle station rebalancing routes. Feature 1) guarantees that there exists feasible solutions in the optimization of inner cluster, while Feature 2) means the traveling cost can be largely reduced as most of the routes are internal/short-term travels. Feature 3) shows vehicles can travel in an overlapping manner to serve as many stations as possible. Feature 4) means outliers are surrounded by a lot of vehicles. Therefore, the stations can be easily served by adding the capacity of vehicles nearby or by assigning two or more vehicles from other clusters to serve them. Although these features show advantages of Algorithm 1 in solving the bicycle station rebalancing problem, this algorithm still has some shortages as a K-Centers based method. On the one hand, the number of clusters needs to be specified. On the other hand, the clustering result is influenced by the initial center set. We further improve Algorithm 1 by proposing an Adaptive Capacity Constrained K-centers Clustering (AdaCCKC) to overcome these shortages.

Algorithm 2 presents the proposed AdaCCKC algorithm. In each round, it begins with a randomly generated initial center set in Step 3. In step 4∼9, Algorithm CCKC is implemented to get a temporary clustering result. If there exists unlabeled bicycle stations, a new cluster center is added to the current center set. The new added center is determined by all unlabeled stations as shown in Step 7. The break condition in Step 6 can also be activated if the number of outliers is below a specified threshold instead of 0, which makes the proposed algorithm more flexible for bicycle stations rebalancing problem. Considering the effect of initial center set on the final clustering result, the number of initial centers is set to vary from 1 to $VNmax$ in Step 2, where

```
1 Input: $TD_{ij}$, $b_i$, $VC$, $\delta$, $VNmax$, $NI$;
2 Output: $\mathbf{c}$;
   1: $VNbest = VNmax$, $zbest = \sum_p \sum_q TD_{pq}$;
   2: for $i$ from 1 to $VNmax$ do
   3:     Generate initial center set $E$;
   4:     for $j$ from $i$ to $VNmax$ do
   5:         $\mathbf{c}$=$CCKC(\mathbf{TD}, \mathbf{b}, VN, VC, \delta, E)$;
   6:         if $\nexists\, h$ that $c(h) = 0$ then break
   7:         else $l = \arg \min_{p,c(p)=0} \sum_{q,c(q)=0} TD_{pq}$,
   8:              $E = unique(\mathbf{c})$, $E = E \bigcup l$;
   9:     end
  10:     $VN = |E|$, $z = \sum_{k=1}^{|E|} \sum_{p,q \in C(k)} TD_{pq}$;
  11:    if $(VN < VNbest)\&(z < zbest)$ then
  12:         $VNbest = VN$, $zbest = z$, $\mathbf{c}^* = \mathbf{c}$;
  13: Repeat Step 2∼12 $NI$ times;
  14: return $\mathbf{c}^*$.
```

Algorithm 2: $AdaCCKC(\mathbf{TD},\mathbf{b},VC,\delta,VNmax, NI)$

$VNmax$ is the maximum number of available vehicles. Step 12 picks out the best clustering result. Steps 2∼12 are repeated many times to reduce the influence of initial center set. As a result, Algorithm 2 can automatically determine the optimal number of vehicles in a smarter way, and users do not need to provide an initial center set.

2.5.2 Clustering-Based Decomposition (CBD) Algorithm

The idea of the clustering-based decomposition (CBD) algorithm, described in Algorithm 3, is to groups stations into $|\mathcal{V}|$ clusters with the consideration of station locations and rebalancing targets. The key is to decompose the large-scale optimization problem into smaller-size problems that are more tractable.

Specifically, the algorithm starts with a set of randomly selected stations as initial cluster centers. Each station is then assigned to its nearest center to form a cluster.

For each cluster, we check inner-cluster net rebalancing targets in Eq. (3). If the balance condition is not satisfied, we select some stations which could reduce the inner-cluster rebalancing target and are close to other clusters, and remove them from the cluster until the inner-cluster balance condition is satisfied. Then, these unassigned stations are re-assigned by solving an optimal station assignment problem, specified in Eqs. (2.38)–(2.41). The optimal station assignment problem aims to minimize the transportation distance while minimizing the unsatisfied rebalancing targets of a cluster. The station assignment and cluster center re-evaluation are iterated until convergence (i.e., cluster centers no longer change).

For outlier station assignment, given a set of clusters of stations $\mathcal{N}_v$ that are visited by vehicle v and a set of outlier stations $Q = \{q_1, q_2, ...q_m\}$, the optimal assignment is determined by solving the following optimization model:

$$\min \quad \mathcal{F}' = \sum_{m\in\mathcal{Q}}\sum_{v\in\mathcal{V}} TC_{mv}z_{mv} + \lambda \sum_{v\in\mathcal{V}} U'_v \tag{2.38}$$

$$\text{s.t.} \quad U'_v \geq 0 \qquad \forall v \in \mathcal{V} \tag{2.39}$$

$$U'_v \geq |\sum_{m\in\mathcal{Q}} rt_m z_{mv} + \sum_{i\in\mathcal{N}_v} rt_i| - C \qquad \forall v \in \mathcal{V} \tag{2.40}$$

$$\sum_{v\in\mathcal{V}} z_{mv} = 1 \qquad \forall m \in \mathcal{Q} \tag{2.41}$$

where z_{mv} is the binary decision variable that equals 1 if outlier station q_m is assigned to cluster v and 0 otherwise, and U'_v is the gap between the total net rebalancing targets and vehicle capacity. The objective (2.38) aims at minimizing the transportation distance between the outlier stations and their assigned cluster centers, as well as the

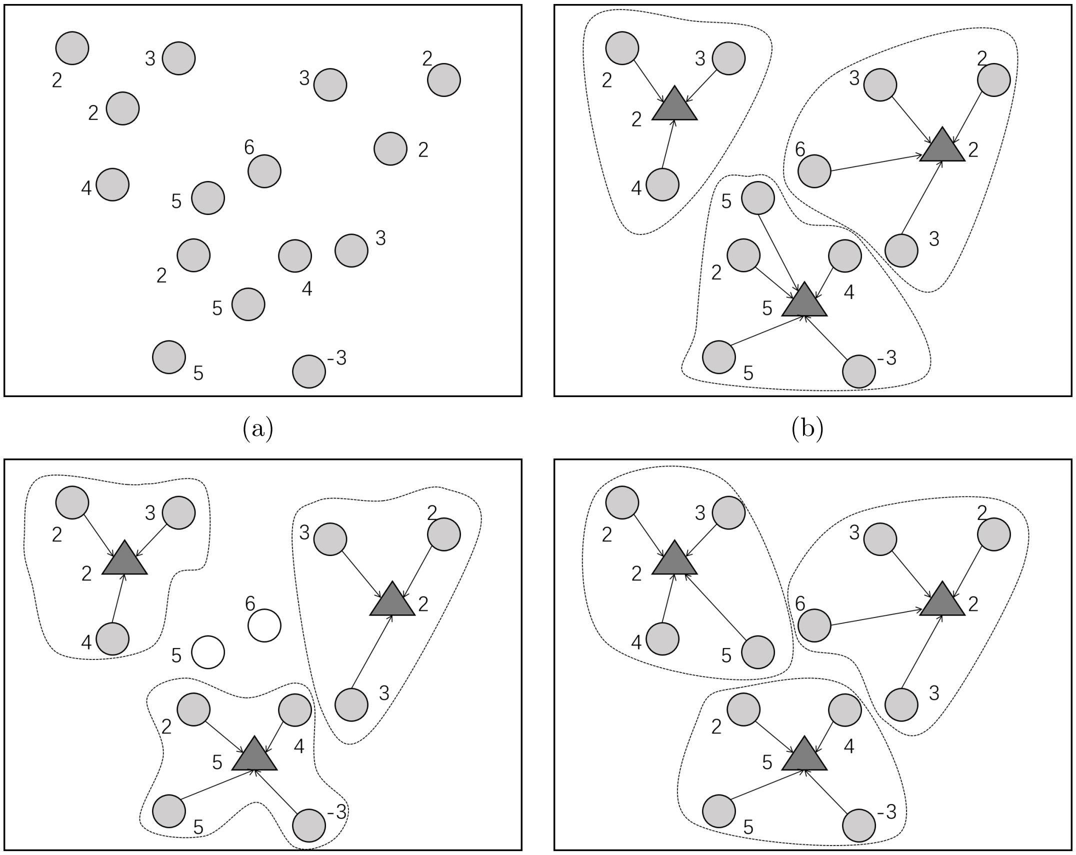
```
1: (Initialization) Set iteration count $k = 0$, and randomly select initial $|\mathcal{V}|$
   stations as cluster centers, denoted by $E^k = \{c_1^k, c_2^k, \ldots, c_{|\mathcal{V}|}^k\}$.
2: (Initial Clustering) Assign each station $i$ to its closest cluster center in $E^k$,
   that is, station $i$ is assigned to the following center:
       $\arg\min_{j \in E^k}(TC_{ij}), \forall i \in \mathcal{N}$
   where ties can be broken arbitrarily.
3: (Outlier Stations Identification) Check the balance condition for each cluster $\mathcal{N}_v^k$:
       $B(\mathcal{N}_v^k) = |\sum_{i \in \mathcal{N}_v^k} rt_i| \leq C, \forall v \in \mathcal{V}$
   For each cluster $v$ that does not satisfy the balance condition, select
   outlier stations iteratively as follows; otherwise, continue to the next step.
4:     Select station $q$ as an outlier station from cluster $\mathcal{N}_v^k$ such that
           $q \in \arg\min_{i \in \mathcal{N}_v^k \setminus \{c_v^k\}} (\sum_{j \in E^k \setminus \{c_v^k\}} TC_{ij})$
   where $c_v^k$ is the current center for cluster $v$, and ties can be broken arbitrarily.
5:     Un-assign $q$ from cluster $\mathcal{N}_v^k$, and update the cluster $\mathcal{N}_v^k \leftarrow \mathcal{N}_v^k \setminus \{p\}$.
6:     Re-check the balance condition for cluster $\mathcal{N}_v^k$. If the balance condition
   is not satisfied, go to Step 2.1; otherwise, continue to the next step.
7: (Station Assignment Optimization) Solve the outlier station assignment
   optimization problem (Eqs. (2.38)–(2.41)). Then, update clusters $\mathcal{N}_v^k$ based on
   the optimization results.
8: (Clustering Update) Obtain the center for each cluster $\mathcal{N}_v^k$ as follows:
       $c_v^{k+1} \in \arg\min_{i \in \mathcal{N}_v^k} \sum_{j \in \mathcal{N}_v^k \setminus \{i\}} TC_{ij}, \forall v \in \mathcal{V}$
   where ties can be broken arbitrarily. Then, set $E^{k+1} = \{c_1^{k+1}, c_2^{k+1}, \ldots, c_{|\mathcal{V}|}^{k+1}\}$.
9: (Termination) If $E^{k+1} = E^k$, continue to the next step; otherwise, increase the
   iteration count $k \leftarrow k + 1$, and go to Step 2.
10: (Routing Optimization) For each cluster $\mathcal{N}_v^k$ of stations, solve the $v$-MILP model
   ($v = 1$ in this case) in Section 2.3.2 (Eqs. (2.5)–(2.24)), and output the optimal
   routing solution.
```

Algorithm 3: Clustering-Based Decomposition (CBD) Algorithm

unsatisfied rebalancing targets of all clusters. Constraints (2.39) and (2.40) specify the definition of $U_v' = max(0, |\sum_{m \in \mathcal{Q}} rt_m z_{mv} + \sum_{i \in \mathcal{N}_v} rt_i| - C)$ for the inner cluster. Constraint (2.41) ensures that each outlier station is assigned to one cluster. As a result, the clustering can always reach the stage where each station is covered by one cluster.

Figure 2.6(a) presents a toy example with 15 stations, assuming the vehicle capacity is 15. The number associated with each station in the figure represents the rebalancing targets. We first randomly select three stations as initial centers which are marked by triangles (see in Figure 2.6(b)). Stations that are the closest to their nearest centers are grouped as one cluster. The capacity condition is checked in each cluster. If one vehicle cannot satisfy the rebalancing targets in one cluster, that is, the net rebalancing target of a cluster is larger than the vehicle capacity, some stations are removed from the cluster as temporary outlier stations (see in Figure 2.6(c)). The stations closer to other clusters are preferentially removed. The outlier stations are then assigned to other centers by solving the outlier assignment optimization problem (Eqs. (2.38)–(2.41)) in Figure 2.6(d). New centers are generated according to the current clustering result and stations are assigned to new centers. This procedure is repeated until convergence, when the cluster centers and station assignments do not change between consecutive iterations.

Using the CDA algorithm, the large-scale multiple capacitated vehicles routing problem is decomposed to v single-vehicle routing problems with small or median problem sizes. Each decomposed problem is actually a v-MILP model with $v = 1$, which is solvable by commercial MILP solvers within a reasonable amount of time. We mention that these 1-MILP models can be solved in parallel, further reducing the computational time.



(a) (b)

(c) (d)

Figure 2.6. Illustrative example of the CBD algorithm

2.5.3 Splitting-Clustering-Aggregation (SCA) Algorithm

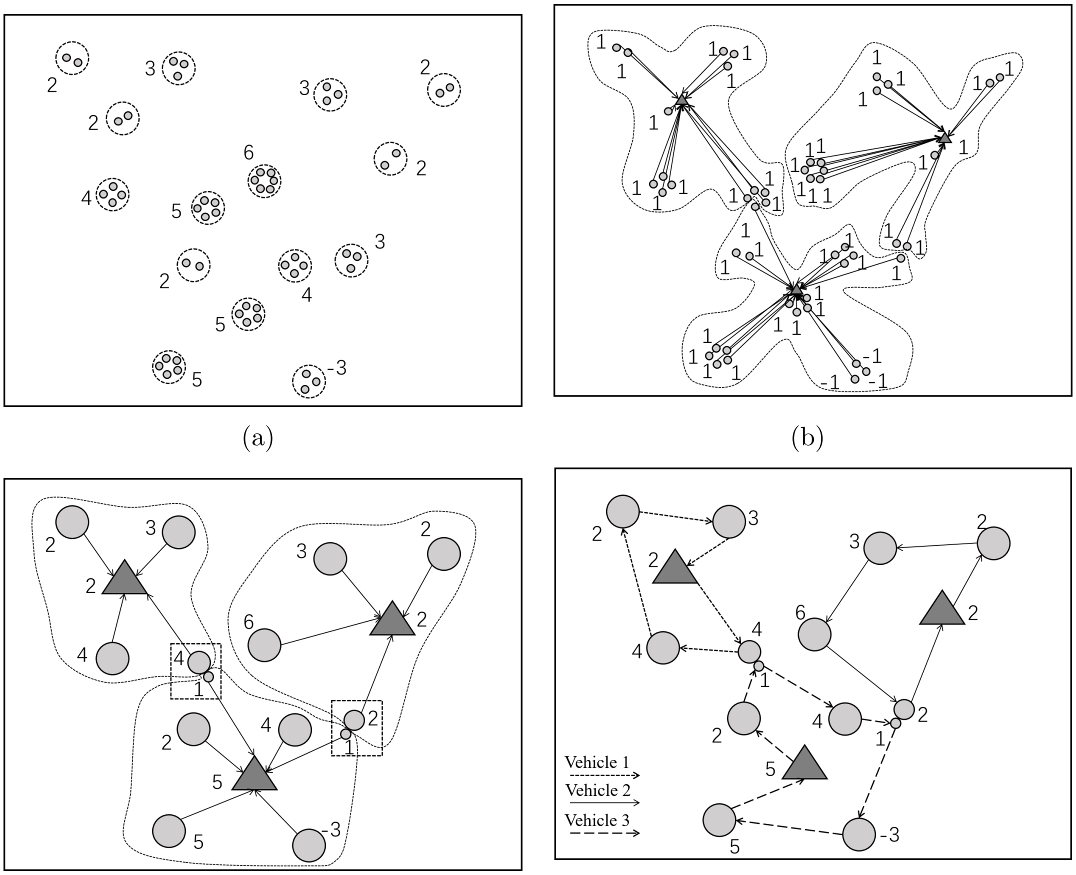
Although the CBD algorithm can decompose the multi-vehicle routing optimization problem into multiple small-scale single-vehicle routing problems, stations within each cluster is only served by one vehicle and hence the rebalancing targets of those outlier stations may only be partially satisfied. In order to further improve the optimality of the problem by allowing multiple vehicles to serve those outlier stations, we propose the splitting-clustering-aggregation (SCA) algorithm described in Algorithm 4. In this algorithm, we first split each station i into $|rt_i|$ substations, where each substation has a rebalancing target of $rt_i/|rt_i|$ (1 or -1). Then we implement the CBD algorithm

```
1: (Splitting) Split each bike station $i$ into $|rt_i|$ substations.
2: (Clustering) Run Steps 0–5 of the CBD algorithm (Algorithm 3)
   to cluster the substations into $|\mathcal{V}|$ clusters.
3: (Aggregation) For each cluster, aggregate substations at the
   same location into one station, and obtain the clusters of stations.
4: (Routing Optimization) For each cluster of stations, solve the $v$-MILP model
   ($v = 1$ in this case) in Section 2.3.2 (Eqs. (2.5)–(2.24)), and output
   the optimal routing solution.
```

Algorithm 4: Splitting-Clustering-Aggregation (SCA) Algorithm

on the set of substations. As a result, the substations are grouped into $|\mathcal{V}|$ clusters. Next, for each cluster of substations, aggregate the substations at the same location into one station. Since substations split from the same station may be grouped into different clusters by the CBD algorithm, the cluster of stations obtained by the SCA algorithm may see some stations belonging to different clusters, which enables those stations to be rebalanced by multiple vehicles. We mention that in the routing optimization step (Step 4) of Algorithm 4, the v-MILP model has been decomposed into $|\mathcal{V}|$ single-vehicle models with a smaller set of stations.

Figure 2.7 demonstrates the key steps of the SCA algorithm on the same example shown in Figure 2.6(a). The SCA algorithm first splits each single station into a number of substations, where the number of substations equals the rebalancing target of the original stations (see Figure 2.7(a)). Then, the CBD algorithm is used to cluster the substations, resulting in clusters in Figure 2.7(b). In Figure 2.7(c), we aggregate the substations into stations when appropriate. From the figure, it can be seen that there are two stations, each belonging to two clusters. Figure 2.7(d) is the corresponding optimal routing result, where the two aforementioned stations are visited by two rebalancing vehicles. This example illustrates that the SCA algorithm can satisfy the rebalancing targets of outlier stations using multiple vehicles.



(a) (b)

(c) (d)

Figure 2.7. Illustrative example of the SCA algorithm

2.6 Experimental Results

To validate the efficiency and effectiveness of our proposed methods, extensive experiments have been performed using real-world data from NYC Citi Bike, Chicago Divvy, and Boston Hubway. Summary statistics of the trip data, station status data and meteorology data used in our tests are presented in Table 2.3. All experiments were conducted on a computer with 3.6 GHz Intel(R) Core i7-4790 CPU and 16 GB RAM.

2.6.1 Experiment Data

Bike sharing system data. The bike transition records of the NYC Citi Bike[1] , Chicago Divvy bike share[2] , and Boston Hubway bike sharing system[3] are publicly available on their official websites. These datasets contain the following information: station id, bike pick-up station, bike pick-up time, bike drop-off station and bike drop-off time. In addition, the station status data, including service status, currently available number of bikes and station capacity, was crawled every 10 minutes from the station status feed site of Citi Bike[4] , Divvy[5] and Hubway[6] .

Hourly weather reports. The weather report data consists of hourly weather reports, including time, weather condition, temperature, humidity, wind speed and visibility, which are publicly available from Weather Underground[7] . The missing meteorology data is completed according to the previous hourly record weather report and the missing wind speed is estimated by the average value of its previous and next reports.

2.6.2 Results for Bike Demand Prediction

Recall that the Nonlinear Autoregressive with Exogenous input (**NARX**) predictor and the pick-drop bike transition (**PDBT**) predictor are developed in Section 2.4 to predict the bike pick-up and drop-off demand, respectively. In order to verify the

[1] https://www.citibikenyc.com/system-data

[2] https://www.divvybikes.com/system-data

[3] https://www.thehubway.com/system-data

[4] https://feeds.citibikenyc.com/stations/stations.json

[5] http://feeds.divvybikes.com/stations/stations.json

[6] http://feeds.thehubway.com/stations/stations.json

[7] https://www.wunderground.com

Table 2.3. Details of the datasets

Data Source		New York City	Chicago	Boston
Time Span		8/1/16 to 7/31/17	7/1/16 to 6/30/17	7/1/16 to 7/31/17
# B-days + Non-B-Days		245 + 116 days	251 + 114 days	271 + 125 days
Bike Data	# of stations	615	580	187
	# of bikes	10,000	5,800	1,600
	# of trip records	15.34 million	3.68 million	1.38 million
Meteorology Data	Heavy snowy/rainy	36 hours	26 hours	46 hours
	snowy/rainy	503 hours	844 hours	998 hours
	Foggy/mist	93 hours	59 hours	37 hours
	Cloudy/sunny	7933 hours	7747 hours	8348 hours
	Temperature	[14, 96.1] ^oF	[-11.9, 96.1] ^oF	[3.9, 97] ^oF
	Visibility	[0.2, 10] mile	[0, 10] mile	[0.1, 10] mile
	Wind Speed	[3.5, 26.5] mph	[3.5, 40.3] mph	[3.5, 46] mph
	Humidity	[13%, 100%]	[18%, 100%]	[15%, 100%]

prediction accuracy, we compare our predictors with the following baseline methods.

- **Multi-similarity-weighted KNN (MSWK)** (Liu et al., 2016a): The MSWK method is built based on a weighted meteorology similairty function, which takes the weighted average demands of top K most similar historical records for prediction.
- **Multi-similarity-based inference (MSI)** (Li et al., 2015a): The MSI considers the similarity of weather, temperature, wind speed and time. Its similarity function is the multiplication of these three similarities, but the weight of different factors are not studied.
- **Autoregressive Integrated Moving Average (ARIMA)**: (Szeto, Ghosh, Basu, & OMahony, 2009): The ARIMA consists of an autoregressive (AR) part and a moving average (MA) part. Here in this paper we set the parameter of ARIMA model $(p, d, q) = (7, 0, 1)$.
- **Random Forest Regressor (RF)** (Grushka-Cockayne, Jose, & LichtendahlJr., 2017): Random Forest regressor fits a number of decision trees on various sam-

ples of the original dataset and uses the average results for prediction and overfitting control.

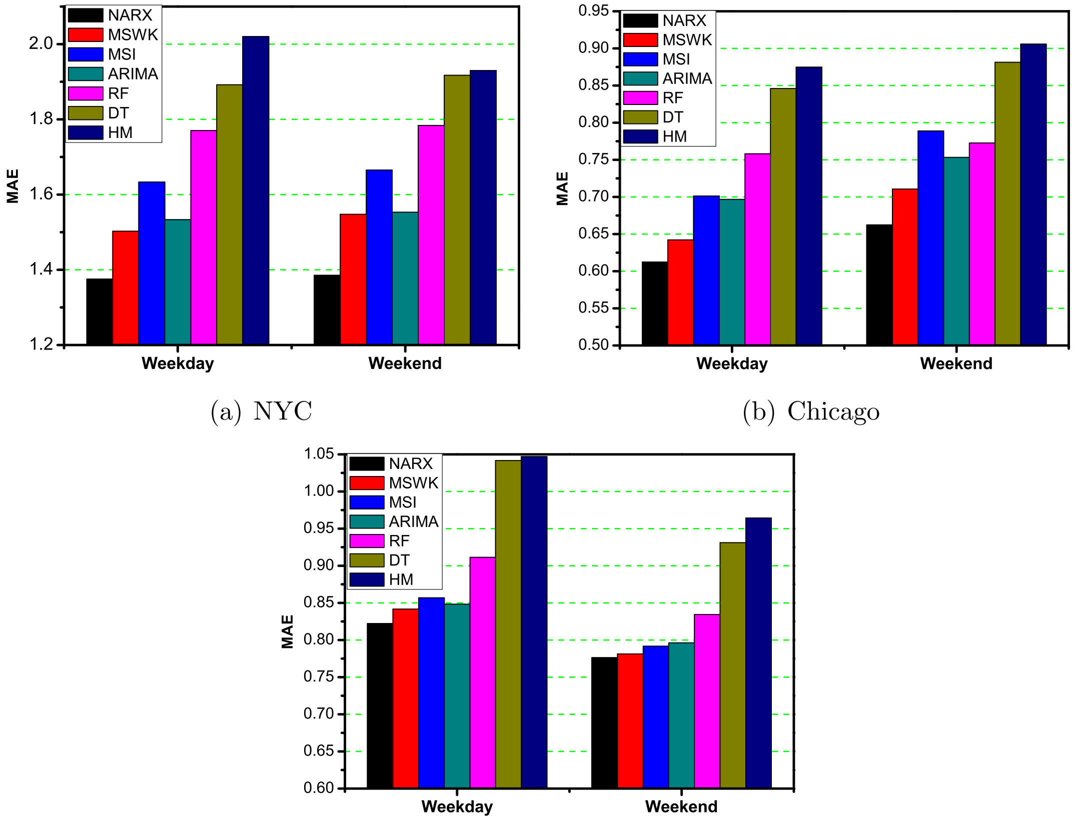
- **Decision Tree Regressor (DT)** (Olson & Wu, 2017): The Decision Tree Regressor breaks down the original dataset into smaller subsets while incrementally developing the decision rules inferred from the data features.
- **Historical Mean (HM)** (Froehlich, Neumann, Oliver, et al., 2009): The HM method takes the average bike demand of previous one-month historical records as prediction value without considering other influential factors. The one-month time period is chosen to ensure sufficient historical records with negligible meteorology difference.

The metric used to measure the prediction accuracy is the Mean Absolute Error (MAE), which measures the number of mis-estimated bikes during one hour.

For each dataset, we select the data of early 220 days as the training set and the rest as the testing set. The validation follows the rolling forcasting procedure. In addition, the business and non-business days' data for the same city are treated as separate data sets.

Bike Pick-up Demand Prediction

The performance comparison for pick-up demand prediction between the proposed NARX model and baselines is summarized in Figure 2.8. It can be seen that the MAE obtained using the proposed NARX model with the MAE=1.375 for NYC weekday, 1.386 for NYC weekend, 0.612 for Chicago weekday, 0.662 for Chicago weekend, 0.822 for Boston weekday, and 0.776 for Boston weekend, which are significantly lower than all the baselines with a significant margin. The performance comparison between NARX and MSWK indicates that we should consider the recurrent dynamics of bike demand as time series. The comparison between NARX and ARIMA validates the importance of considering the non-linear relationships. Moreover, the multi-source prediction models (NARX, MSWK, MSBI and MSEWK) are better than the signle-factor prediction models (RF, DT, and HM) by leveraging multiple meteorology factors and historical demand records.



(a) NYC

(b) Chicago

(c) Boston

Figure 2.8. Performance comparison of bike pick-up demand prediction

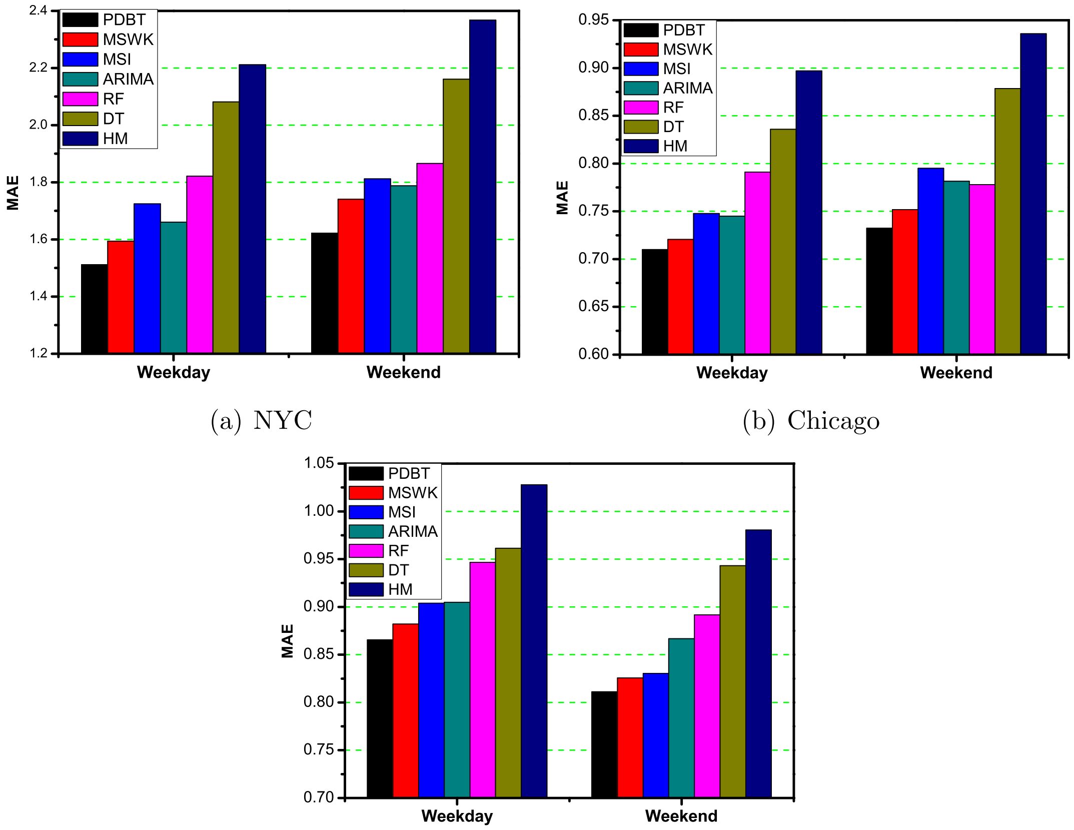
Bike Drop-off Demand Prediction

In a similar vein, Figure 2.9 displays the performance comparison for drop-off demand prediction between the proposed PDBT predictor and baselines. It is seen from the figure that PDBT has the lowest MAE among all methods tested, with an MAE=1.512 for NYC weekday, 1.622 for NYC weekend, 0.710 for Chicago weekday, 0.732 for Chicago weekend, 0.822 for Boston weekday, and 0.776 for Boston weekend. The results indicate that PDBT further improves the prediction accuracy by considering station-to-station trip transitions. We mention that the performance of PDBT can be further improved by collecting more trip data to generate better numarical analysis for inter-station transitions.

2.6.3 Results for Bike Rebalancing Optimization

Given the predicted bike pick-up and drop-off demand, the next step is to optimize the bike rebalancing operations. In order to illustrate the differences between the general v-MILP model and the proposed hierarchical optimization approaches, we first use a small instance with 2 vehicles and 15 stations. Figure 2.10 shows the different routing results among four strategies: the general v-MILP, outlier removal strategy, partially visiting strategy, and multiple vehicle visiting strategy. Note that the general v-MILP model was solved by Gurobi MILP solver to optimality, and its results are served as our baselines.

In Figure 2.10, each data box represents a bike sharing station associated with four parameters: Station Index, unsatisfied rebalancing target U, rebalancing target rt, and the number of redistributed bikes ro (see the legend on each subfigure).



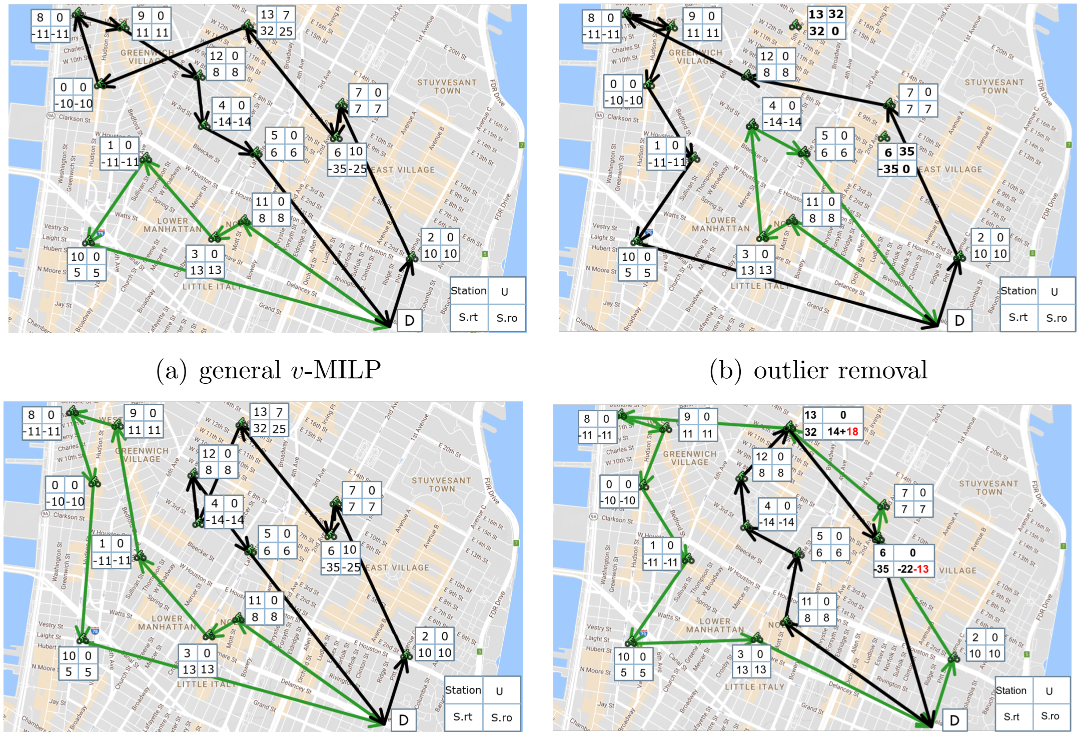
(a) NYC

(b) Chicago

(c) Boston

Figure 2.9. Performance comparison of bike drop-off demand prediction

Starting from depot D, the first vehicle follows the route drawn in green arrows and the second vehicle follows route drawn in black arrows. The optimal result of the general v-MILP is displayed in Figure 2.10(a). It is seen that all stations are visited and the rebalancing targets for most stations are met (i.e., $U = 0$). The two outlier stations with a rebalancing target larger than the vehicle capacity (i.e., $rt_6 = -35$ and $rt_{13} = 32$) have unsatisfied rebalancing targets ($U_6 = 10$ and $U_{13} = 7$). The result of the outlier removal strategy is displayed in Figure 2.10(b), where all rebalancing targets are strictly met, while the two outlier stations are left unvisited in order to ensure inner cluster optimization feasibility. Figure 2.10(c) displays the result of the partially visiting strategy, where the workload is more balanced for each vehicle and the two outlier stations have unsatisfied rebalancing target. Finally, the result of the multiple visiting strategy is displayed in Figure 2.10(d). It is seen that the outlier stations are visited by both vehicles and thus the most rebalancing targets are met.



(a) general v-MILP

(b) outlier removal

(c) partially visiting

(d) multiple visiting

Figure 2.10. Comparison of routings for different optimization strategies.

To verify the computational advantages of proposed optimization approaches, we further test several sets of instances based on real-world bike sharing data and their predicted rebalancing targets. Table 2.4–2.11 present the experimental results conducted on instances with $|\mathcal{N}|$ ranging from 20 to 35. For each instance, we compare the objective (Obj) and computational time (CT; in seconds) of different approaches: v-MILP model, Nearest Neighbor Search (NS)(Yianilos, 1993)+1-MILP model, partially visiting, and multiple visiting strategy. The objective gap (Gap%) is calculated as the percentage difference between the optimal result of the hierarchical approach and the baseline approach (i.e., the general v-MILP model). The lower case letter of each case ID indicates the bike sharing system. For example, Case “n20A” indicates the instance from NYC and Case “c20A” indicates the instance from Chicago. Note that, in Table 2.11, Gurobi was unable to solve the problems within 10 hours, and hence no result is reported for v-MILP and objective gaps.

It is observed that the computational time of the general v-MILP model is much longer than the hierarchical approaches. Specifically, it typically took several hours to solve large instances, rendering the general v-MILP model non-applicable for solving real cases. For the outlier removal strategy, although the computational time is much shorter than the baseline approach, the objective values are much larger than those of the baseline approach, due to the neglect of outlier stations. Thus, it is not recommended to be used in the real systems. The partially visiting strategy is able to obtain close results compared to the v-MILP model, while significantly reduce the computational times. Finally, the multiple visiting strategy achieves the best results as it is able to serve outlier stations using multiple vehicles. Meanwhile, the computational times are typically within 15 minutes, even for problems with $\mathcal{N} = 35$ and $\mathcal{V} = 3$.

Table 2.4. Experimental results for $|\mathcal{N}| = 20$ and $|\mathcal{V}| = 2$

Case	v-MILP		NS+1-MILP		outlier removal		partially visiting		multiple visiting	
	Obj	CT	Obj (Gap%)	CT	Obj (Gap%)	CT	Obj (Gap%)	CT	Obj (Gap%)	CT
n20A	13.66	2455	48.21(253)	5	43.24(217)	9	14.24(4.25)	12	11.46(-16.1)	14
n20B	15.54	5706	51.12(229)	72	46.6(199)	84	16.1(3.6)	96	11.43(-26.4)	234
n20C	9.19	179	25.89(181)	53	23.82(159)	76	9.82(6.9)	3	8.3(-9.7)	3
c20A	24.96	57	57.51(130)	13	40.97(64.1)	114	27.47(10.1)	117	26.23(5.1)	288
c20B	18.1	37	41.41(129)	38	36.48(101.5)	15	19.48(7.6)	28	19.22(6.2)	234
c20C	21.28	268	38.39(80.4)	11	36.95(73.6)	4	23.45(10.2)	8	23.33(9.6)	14
b20A	15.77	161	35.54(125)	8	30.83(95.5)	85	17.33(9.9)	12	15.7(-0.4)	28
b20B	19.16	5706	23.58(23.1)	27	19.45(1.5)	84	19.45(1.5)	96	19.45(1.5)	234
b20C	15.24	254	31.33(105)	4	30.4(99.5)	13	17.4(14.2)	13	15.3(0.4)	18

Finally, we present a real-world sized case study for a randomly selected dataset

Table 2.5. Experimental results for $|\mathcal{N}| = 20$ and $|\mathcal{V}| = 3$

Case	v-MILP		NS+1-MILP		outlier removal		partially visiting		multiple visiting	
	Obj	CT	Obj (Gap%)	CT	Obj (Gap%)	CT	Obj (Gap%)	CT	Obj (Gap%)	CT
n20A	16.45	9389	48.33(194)	22	46.13(180.5)	3	17.13(4.1)	5	14.67(-10.8)	31
n20B	18.33	11061	50.02(173)	62	49.13(168.1)	6	18.63(1.7)	8	14.39(-21.5)	10
n20C	10.71	126	29.73(177)	98	25.94(142.2)	32	11.94(11.5)	33	10.46(-2.3)	46
c20A	28.1	2732	48.51(72.7)	21	42.64(51.8)	9	29.14(3.7)	9	28.44(1.2)	51
c20B	20.34	132	51.16(151)	9	38.41(88.8)	6	21.41(5.2)	6	19.22(-5.5)	8
c20C	23.49	897	44.09(87.7)	31	38.49(63.8)	4	24.99(6.4)	5	24.29(3.4)	4
b20A	17.43	91	43.89(152)	4	31.62(81.4)	3	18.12(4)	3	16.3(-6.5)	16
b20B	21.48	172	25.76(20)	3	22.99(7)	23	22.99(7)	23	22.99(7)	23
b20C	17.57	593	38.23(118)	4	31.12(77.1)	6	18.12(3.1)	6	17.64(0.4)	32

Table 2.6. Experimental results for $|\mathcal{N}| = 25$ and $|\mathcal{V}| = 2$

Case	v-MILP		NS+1-MILP		outlier removal		partially visiting		multiple visiting	
	Obj	CT	Obj (Gap%)	CT	Obj (Gap%)	CT	Obj (Gap%)	CT	Obj (Gap%)	CT
n25A	11.94	1503	26.53(122)	68	23.47(96.5)	77	12.76(6.8)	61	11.45(-4.2)	23
n25B	17.98	1075	59.32(230)	5	53.73(198)	7	18.23(1.4)	8	15.86(-11.8)	16
n25C	18.22	987	55.18(203)	6	54.15(197)	9	18.65(2.4)	9	14.21(-22)	11
c25A	15.91	3460	39.65(149)	286	30.31(90.6)	372	17.31(8.9)	500	16.41(3.1)	988
c25B	24.94	607	48.97(96.4)	8	40.48(62.3)	15	25.48(2.2)	14	15.86(-36.4)	16
c25C	26.72	393	58.74(120)	19	56.45(111.3)	12	28.44(6.5)	24	27.28(2.1)	11
b25A	19.33	1503	22.73(17.6)	13	21.35(10.5)	61	21.35(10.5)	62	21.35(10.5)	61
b25B	14.15	1075	17.58(24.2)	3	15.59(10.2)	8	15.59(10.2)	8	15.59(10.2)	8
b25C	20.15	987	41.09(104)	6	36.3(80.1)	8	21.8(8.2)	7	20.76(3)	10

Table 2.7. Experimental results for $|\mathcal{N}| = 25$ and $|\mathcal{V}| = 3$

Case	v-MILP		NS+1-MILP		outlier removal		partially visiting		multiple visiting	
	Obj	CT	Obj (Gap%)	CT	Obj (Gap%)	CT	Obj (Gap%)	CT	Obj (Gap%)	CT
n25A	15.15	1862	36.17(139)	55	30.01(98.1)	67	17.01(12.3)	68	16.67(10)	221
n25B	19.36	1094	56.47(192)	59	55.49(186.7)	8	19.99(3.3)	10	12.4(-36)	25
n25C	20.19	1715	78.1(287)	112	56.64(180.5)	17	21.14(4.7)	18	14.59(-27.8)	39
c25A	19.98	5811	40.04(100)	11	34(70.2)	3	21(5.1)	2	20.31(1.6)	33
c25B	27.43	2745	60.28(120)	31	42.43(54.7)	9	27.43(0)	11	26.25(-4.3)	26
c25C	26.09	1355	62.79(141)	5	42.37(62.4)	8	27.37(4.9)	9	26.41(1.2)	23
b25A	21.79	2058	25.47(17)	3	23.2(6.5)	12	23.2(6.5)	13	23.2(6.5)	12
b25B	16.15	7400	17.83(9.1)	7	17.35(7.4)	11	17.35(7.4)	11	17.35(7.4)	11
b25C	21.8	12759	38.79(77.9)	2	36.78(68.7)	8	22.28(2.2)	8	22.07(1.2)	9

Table 2.8. Experimental results for $|\mathcal{N}| = 30$ and $|\mathcal{V}| = 2$

Case	v-MILP		NS+1-MILP		outlier removal		partially visiting		multiple visiting	
	Obj	CT	Obj (Gap%)	CT	Obj (Gap%)	CT	Obj (Gap%)	CT	Obj (Gap%)	CT
n30A	18.91	7627	36.23(91.6)	11	34.29(81.3)	3	19.79(4.7)	3	18.15(-4)	2
n30B	11.33	411	31.69(179)	22	26.76(107)	4	12.26(8.1)	8	11.11(-14.2)	75
n30C	16.7	42831	32.27(93.2)	42	30.11(80.3)	39	17.62(5.4)	37	17.62(5.4)	39
c30A	19.47	3491	37.34(91.8)	4	34.63(77.9)	3	21.13(8.5)	3	20.59(5.7)	9
c30B	16.55	7132	22.46(35.7)	12	17.48(5.6)	4	17.48(5.6)	4	17.48(5.6)	4
c30C	24.59	2570	70.64(187)	98	53.05(115.7)	39	25.55(3.9)	38	24.57(-0.1)	46
b30A	22.23	9676	41.92(88.6)	26	37.74(69.8)	106	23.24(4.6)	107	21.51(-3.2)	380
b30B	15.4	988	51.16(232)	17	30.83(138.1)	250	16.83(9.3)	29	15.23(-1.1)	250
b30C	18.5	22059	34.8(88.1)	13	34.8(88.1)	59	20.8(12.4)	26	17.55(-5.1)	221

Table 2.9. Experimental results for $|\mathcal{N}| = 30$ and $|\mathcal{V}| = 3$

Case	v-MILP		NS+1-MILP		outlier removal		partially visiting		multiple visiting	
	Obj	CT	Obj (Gap%)	CT	Obj (Gap%)	CT	Obj (Gap%)	CT	Obj (Gap%)	CT
n30A	21.73	3691	48.67(124)	133	36.72(69)	9	24.65(13.5)	9	22.22(2.3)	21
n30B	12.95	10319	40.31(211)	19	29(124)	498	14.5(12)	465	12.08(-6.7)	553
n30C	19.65	60266	20.97(6.7)	105	20.29(3.3)	120	20.29(3.3)	120	20.29(3.3)	121
c30A	21.62	3905	43.11(99.4)	25	35.93(66.2)	18	22.43(3.8)	16	21.93(1.4)	47
c30B	19.79	11128	26.96(36.3)	42	21.8(10.2)	28	21.8(10.2)	28	21.8(10.2)	28
c30C	27.23	36181	71.27(162)	45	57.76(112.2)	320	30.26(11.2)	321	27.62(1.5)	447
b30A	23.93	45327	45.27(89.2)	6	37.74(67.7)	7	25.64(7.1)	13	22.61(-5.5)	14
b30B	17.03	2307	33.24(95.2)	36	32.41(90.3)	40	18.41(8.1)	39	17.41(2.2)	42
b30C	20.57	14171	48.61(132)	9	35.63(73.2)	6	21.63(5.1)	6	20.3(-1.3)	15

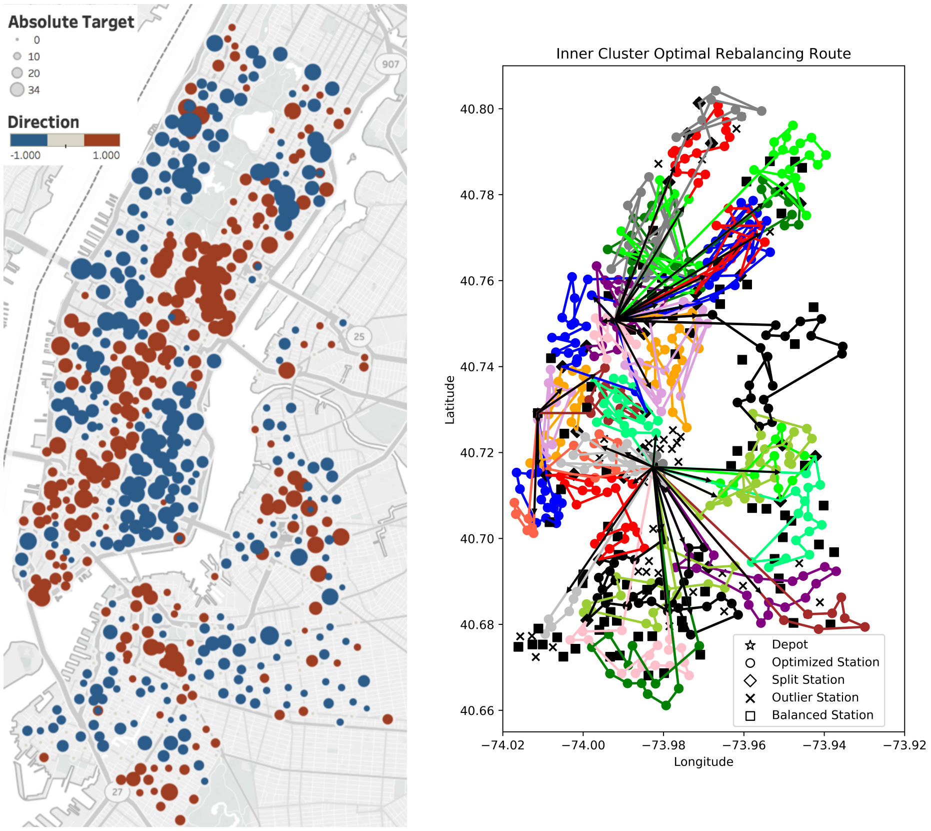
Table 2.10. Experimental results for $|\mathcal{N}| = 35$ and $|\mathcal{V}| = 2$

Case	v-MILP		NS+1-MILP		outlier removal		partially visiting		multiple visiting	
	Obj	CT	Obj (Gap%)	CT	Obj (Gap%)	CT	Obj (Gap%)	CT	Obj (Gap%)	CT
n35A	15.64	50322	33.41(114)	54	29.72(90)	68	16.73(6.9)	74	16.19(3.5)	82
n35B	22.01	16418	48.15(118)	121	46.04(109)	21	24.04(9.2)	27	15.84(-28.1)	241
n35C	13.56	13589	29.03(141)	66	28.63(111)	258	14.63(7.9)	436	14.28(5.3)	2873
c35A	19.19	13228	37.19(93.8)	52	34.02(77.3)	51	20.52(6.9)	63	18.37(-4.3)	69
c35B	19.88	51348	40.88(106)	128	37.49(88.6)	165	22.99(15.7)	236	18.26(-8.1)	251
c35C	19.01	21467	48.22(154)	279	37.04(94.9)	1777	20.54(8.1)	1800	16.53(-13)	216
b35A	24.12	50322	46.3(91.9)	133	40.58(68.2)	491	28.05(16.3)	493	23.09(-4.3)	652
b35B	26.33	16418	52.2(98.2)	65	46.04(74.8)	268	29.76(13)	23	27.17(3.2)	418
b35C	20.92	13589	48.61(132)	206	47.97(129)	571	22.01(5.2)	580	22.01(5.2)	663

Table 2.11. Experimental results for $|\mathcal{N}| = 35$ and $|\mathcal{V}| = 3$

Case	v-MILP		NS+1-MILP		outlier removal		partially visiting		multiple visiting	
	Obj	CT	Obj (Gap%)	CT	Obj (Gap%)	CT	Obj (Gap%)	CT	Obj (Gap%)	CT
n35A	-	-	46.63(-)	208	36.26(-)	467	22.26(-)	460	19.79(-)	533
n35B	-	-	68.51(-)	105	50.08(-)	224	28.08(-)	233	19.7(-)	258
n35C	-	-	44.01(-)	74	31.69(-)	496	17.69(-)	532	16.77(-)	589
c35A	-	-	52.11(-)	53	38.41(-)	90	24.91(-)	93	22.19(-)	699
c35B	-	-	55.44(-)	103	41.6(-)	25	27.1(-)	24	22.32(-)	38
c35C	-	-	50.49(-)	111	40.66(-)	80	24.16(-)	80	20.16(-)	115
b35A	-	-	53.31(-)	281	46.32(-)	407	30.11(-)	412	27.16(-)	643
b35B	-	-	59.37(-)	89	45.89(-)	433	32.88(-)	445	28.45(-)	566
b35C	-	-	41.33(-)	102	40.13(-)	329	28.13(-)	354	23.47(-)	591

for the NYC Citi Bike system. The rebalancing targets of 615 stations in NYC are displayed in Figure 2.11(a)). Particularly, the red dots represent stations that need bike drop-offs and blue dots represent stations that need pick-ups. We mention that 82 stations with zero inventory targets (marked as gray dots in Figure 2.11(a)) are self-balanced stations, and thus are filtered before optimization. Figure 2.11(b) shows the optimization results using multiple visiting strategy with 35 rebalancing vehicles. The outlier stations are detected and marked as "X". The stations that are visited by multiple vehicles are marked as "diamond". The self-balanced stations are marked as "square". The stations belonging to the same cluster are displayed in the same color and are rebalanced by the same vehicle. The depots are marked as "star" from which an arrow points toward the first visited station. The computational time for the clustering is 20 seconds and the computational cost for the largest single-vehicle inner cluster route optimization (including 34 nodes) is 3987 seconds. If we implement the 1-MILP models in parallel computers, the total time to find the optimal solution of the route optimization for the NYC Citi Bike system is merely above 4000 seconds for this case study, which is practically implementable given that such operations is run on a daily basis.



(a) Target Distribution

(b) Optimal Route

Figure 2.11. Optimal vehicle routing using multiple visiting strategy for a case study in NYC)

2.7 Conclusion

In this paper, we developed a multi-source data-driven optimization approach for addressing the bike rebalancing problem in bike sharing systems. Specifically, we first propose a nonlinear autoregressive network with exogenous meteorology factors

(NARX) model for predicting station-level bike pick-up demand, and a pick-drop bike transition (PDBT) model for trip pattern discorvery and bike drop-off demand prediction. Then, the station inventory target levels are determined based on the predicted traffic flows, and subsequently used for the rebalancing operations. For the bike rebalancing operations optimization, we developed three algorithms based on the clustering-first optimization-second idea. This idea is similar to the mathematical decomposition approaches, which decompose large-scale optimization problems into smaller, solvable ones. The decomposition used in the proposed approaches are based on data analytics and clustering. Furthermore, the clustering-based approaches consider outlier stations, which have large inventory targets exceeding rebalancing vehicle capacity. Finally, the extensive numerical experiments using real-world bike sharing system data from the NYC Citi Bike system, Chicage Divvy bike system and Boston Hubway bike system verified the advantages of our approach for bike demand prediction and large-scale bike rebalancing optimization, both in terms of solution quality and computational results.

This work still has some limitations and leaves spaces for future research. First, we do not support single vehicle-multi-visit solution. In our solution, one vehicle can only visit one station no more than once. However, it is possible to improve the rebalancing efficiency by visiting one station multiple times by the same rebalancing vehicle. Another study can be performed to examine the use of trailers that complements the rebalancing trucks. Trailers have much smaller capacities compared to the trucks, but are more flexible and thus can be used to handle outlier stations. Finally, the static bike rebalancing operations are executed once per day during midnight, and all operations should be completed before the system resumes normal operations (e.g., at 6am). Technically, a time window for each rebalancing vehicle can be imposed to guarantee the time-feasibility of the routes obtained. We do not model such time window in our MILP models, since such time-feasibility is guaranteed in our solutions given sufficient number of vehicles available. In a future study, we can impose such time window to the routing optimization model, and it would be interesting to check those cases when there are insufficient number of vehicles in service.

CHAPTER 3

FUNCTIONAL ZONE BASED HIERARCHICAL DEMAND PREDICTION FOR BIKE SYSTEM EXPANSION

Many providers of bike sharing systems are ready to expand their bike stations from the existing service area to surrounding regions. A key to success for a bike sharing systems expansion is the bike demand prediction for expansion areas. There are two major challenges in this demand prediction problem: First. the bike transition records are not available for the expansion area and second. station level bike demand have big variances across the urban city. Previous research efforts mainly focus on discovering global features, assuming the station bike demands react equally to the global features, which brings large prediction error when the urban area is large and highly diversified. To address these challenges, in this chapter, I develop a hierarchical station bike demand predictor which analyzes bike demands from functional zone level to station level. Specifically, I first divide the studied bike stations into functional zones by a novel Bi-clustering algorithm which is designed to cluster bike stations with similar POI characteristics and close geographical distances together. Then, the hourly bike check-ins and check-outs of functional zones are predicted by integrating three influential factors: distance preference, zone-to-zone preference, and zone characteristics. The station demand is estimated by studying the demand distributions among the stations within the same functional zone. Finally, the extensive experimental results on the NYC Citi Bike system with two expansion stages show the advantages of our approach on station demand and balance prediction for bike sharing system expansions.

3.1 Introduction

With the success of bike sharing system, most urban cities are planning or have been constructing bike sharing network expansion to attract more customers. For example, NYC has completed two bike sharing network expansions since its foundation in 2013. However, despite the significant benefits from bike sharing network expansion, it is very challenging to decide the expansion strategy which relies on an accurate bike demand prediction for expansion areas. An accurate bike demand prediction can help bike sharing system designers estimate how many new customers will be attracted and how much additional operation cost they need to spend on a larger system. To this end, in this chapter, I study the bike demand prediction problem for bike sharing system expansion. There are two major challenges for this problem. First, there are no historical bike transition records available in the expansion areas. This challenge makes it impractical to conduct a direct supervised learning model on the station network after expansion. Second, the station level bike demand has big variances across the city, which can be impacted by multiple factors, such as time, location, surrounding environment (Point of Interest (POI) structure), transportation network, and human mobilities.

A number of recent researchers have studied the bike demand prediction problem. Most studies on bike demand prediction are based on single factor predictors like stochastic process (Schuijbroek, Hampshire, & van Hoeve, 2013; Alvarez-Valdes, Belenguer, Benavent, Bermudez, Muoz, et al., 2016) without considering the impact of other influential factors. A promising way to improve bike demand prediction accuracy is to leverage a variety of data that is directly or indirectly related to the public bike sharing service (Liu, Sun, Chen, & Xiong, 2016b; Li, Zheng, Zhang, & Chen, 2015b). However, these methods rely on the availability of historical bike transition records to train the proposed model and are not applicable for bike sharing system expansion. Our previous work (Liu et al., 2015) proposes a global station level bike demand predictor based on a set of fine grained global features which are used for current station network redesign, however, considering the complexity of urban city structures and station demand variances across the large area of urban city, the global features may not affect bike demand equally in different regions.

Indeed, the emergence of multi-source big data enables a new paradigm for enhancing bike demand predictions. Along this line, I exploit multi-source data related to bike sharing services, such as trip records, station status records, POI dataset, taxi trip records for developing station level bike demand solutions. Specifically, starting from the existing bike sharing system (which I call it principle bike system) with its historical trip records available, I build a hierarchical prediction model to analyze bike demand from zone level to station level. The station in service area is firstly divided into different functional zones through our Bi-Clustering algorithm which considers the POI structures and station locations simultaneously. Then, a zone level bike check-in and check-out predictor is studied based on the bike trip distance preference, zone-to-zone preference, zone characteristics and the historical transitions of the principle bike system. The check-ins and check-outs are then distributed to each station within the functional zone according to the POI structures and their links to other transportation networks. To predict the station bike demand after system expansion, I re-estimate the zone level features by considering the expanded zone-to-zone network and the inner-zone demand distribution for expansion area stations.

Finally, I carry out extensive experiments on a real-world dataset of three different time periods from the NYC Citi Bike system: Stage 1. the principle bike station system consisting of 329 stations from 07/01/2013 to 07/31/2015, Stage 2. the bike system after first expansion including 486 stations from 08/06/2015 to 07/18/2016 and Stage 3. the third stage starts from 07/23/2016 to 11/30/2016 with 617 stations in service after second expansion. Figure 3.1(a) presents the three stages of station distributions with each dot representing a bike station in New York City. The red dots represent the principle bike station distribution. The orange and the blue dots represent the second and third stages of station expansions respectively. In additional, a few stations represented by different symbols are closed in different stages.

3.2 Related Work

There is an increasing interest in optimization problems arising in bike sharing systems. Below we describe some related studies that have been accomplished on demand prediction for bike sharing systems.

Station Clustering. The clustering algorithms have been proposed to discovery bike transition patterns, reduce the station demand variance and improve prediction accuracy. Patrick, etc (Vogel, Greiser, & Mattfeld, 2011) explored bike activity patterns based on temporal and spatial validation of clusters, and revealed imbalances in the distribution of bikes. Lin, etc (Li et al., 2015b) proposed a Bipartite station clustering algorithm consisting of Geo-clustering and Bike-Transit-Clustering according to the similarities of bike usage patterns and station locations. Similarly, Chen, etc. (Chen et al., 2016) proposed a Geographically-Constrained Station Clustering to group stations. However, their station clustering algorithms are based on historical bike transition records. Motivated by the Functional Zone discovery analysis (Yuan et al., 2015; Long & Shen, 2015), our Bi-Clustering algorithm is based on both of POI structures and station geographical constraints that could be applicable for the expansion areas where no historical bike transition records are available. The identification of heat-peak bike stations is motivated by a recently proposed clustering method that is published in *Science* in 2014 (Rodriguez & Laio, 2014), however, the cluster centers in our work are POI heat peaks rather than density peaks. In order to discover functional zones with distinguished POI characteristics, we proposed the HPC clustering algorithm, which can find representative stations via different POI categories. Different from most existing clustering algorithms based on predefined similarities of objects(Xu & Wunsch, 2005; Luxburg, 2007), the POI characteristics of stations fadeaway in the process of computing similarities.

Bike Demand Prediction. The early research on bike sharing system focused on the studies of bike activity patterns discovery (Kaltenbrunner, Meza, Grivolla, Codina, & Banchs, 2010a; O'Brien, Cheshire, & Batty, 2014; Zhou, 2015) or daily bike demand forecasting using data mining techniques and classical empirical statistical methods. Yutaka (Motoaki & Daziano, 2015) and Juan(Garca-Palomares, Gutirrez,

& Latorre, 2012) built multi-factor statistical models for bicycle demand prediction with the consideration of weather and geography. The hourly bike demand prediction was investigated by implementing statistical models or machine learning techniques on multi-source data (Alvarez-Valdes, Belenguer, Benavent, Bermudez, Muoz, et al., 2016; Schuijbroek et al., 2013; Liu et al., 2016b). A hierarchical bike traffic prediction model that integrating station clustering algorithm and meteorology reports were also studied (Li et al., 2015b). However, all of these prediction models require the availability of historical transition records of target stations and thus are not applicable for expansion demand prediction problem. Liu etc.(Liu et al., 2015), Wang (Wang, 2016) and Zeng etc.(Zeng et al., 2016) built station demand prediction models by extracting global features from multiple static factors of surrounding environment and public transportation networks. However, among these multi-factor prediction models, the feature bias among stations across the urban areas are neglected with the global features and predictors. Moreover, the direct analysis of station-to-station bike transition may suffer from insufficient transition records at station level.

3.3 Problem Formulation

In this section, we first introduce some preliminaries used throughout this paper, and then formally define the problem of station bike demand prediction for bike system expansion.

3.3.1 Preliminaries

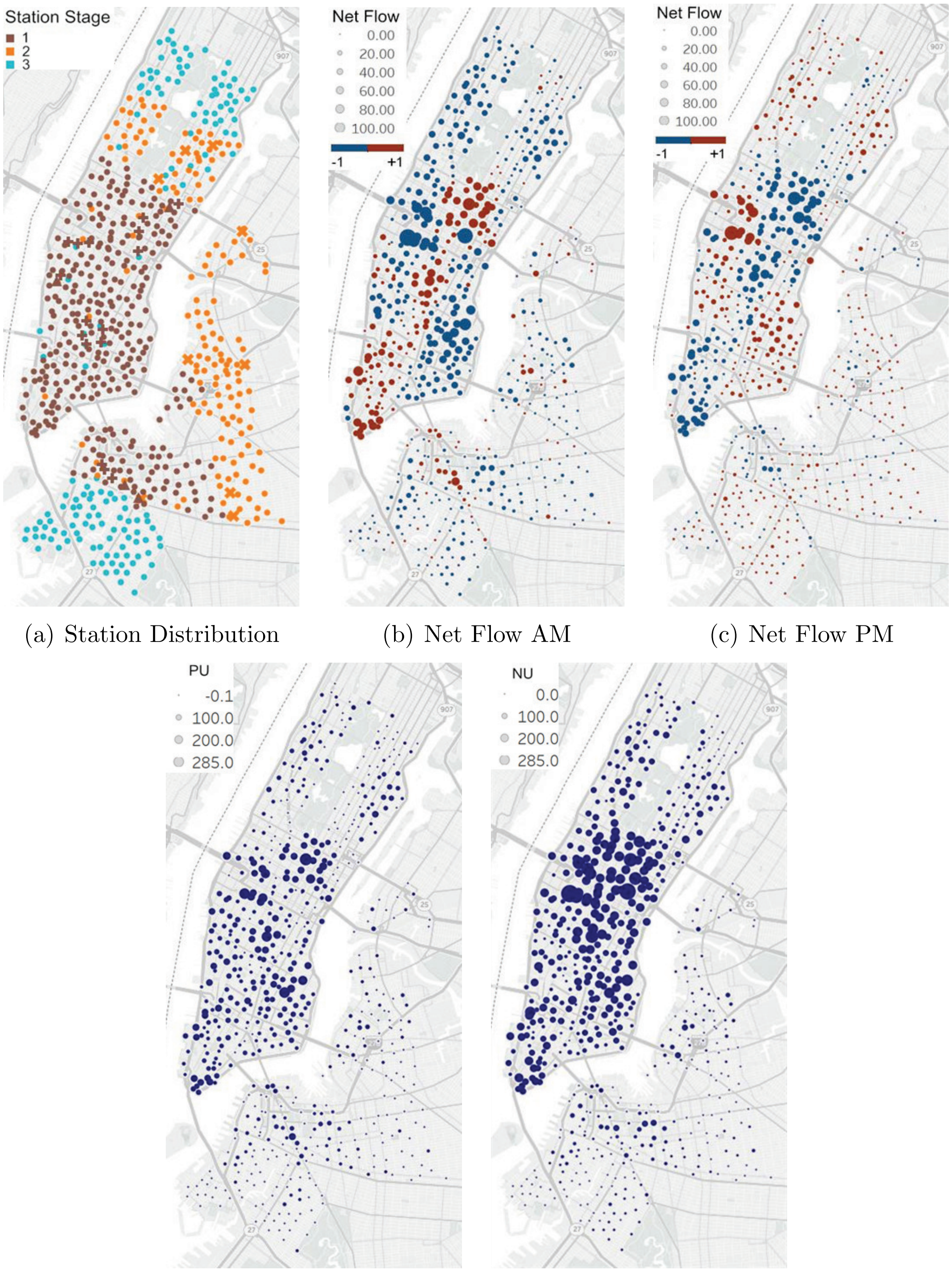
Station bike demand and unbalance

The station bike demand is defined as the pick-up (drop-off) frequency per unit time when the station is available. Station availability means the station is in service and there are bikes available for pick-up (drop-off). Station unavailability is usually due to maintenance, street block, empty dock (for pick-up) and full dock (for drop-off). We do not consider the station demand during its unavailable period.

Definition 1: Station Bike demand. Let $s_i.pf(t)(s_i.df(t))$ and $s_i.pa(t)(s_i.da(t))$ represent the pick-up (drop-off) frequency and the pick-up (drop-off) available time of station i during time slot t. Each time slot t represents a 60 minutes time duration. The bike demand during the day is split into 24 time slots: $t \in \{0, 1, ..., 23\}$. The station pick-up (drop-off) demand $s_i.pd(t)$ $((s_i.dd(t)))$ is defined as follows:

$$s_i.pd(t) = \frac{s_i.pf(t)}{s_i.pa(t)} \quad (s_i.dd(t) = \frac{s_i.df(t)}{s_i.da(t)}) \tag{3.1}$$

Due to unbalanced bike demand distribution, some bike stations can have continuous large positive bike flows (drop-off demand is much larger than pick-ups) or negative bike flows. The station bike net flow distributions during AM and PM rush hours are presented in Figure 3.1(b) and Figure 3.1(c) as an example. In Figure 3.1(b) and Figure 3.1(c), each dot represents a bike station in stage 3 with its size representing the absolute value of net flow. The red color represents a positive net flow (drop-off frequency is larger than pick-up frequency) and the blue color represents a negative net flow. As can be seen, the station demand distribution is unbalanced



(a) Station Distribution (b) Net Flow AM (c) Net Flow PM

(d) Positive Unbalance (e) Negative Unbalance

Figure 3.1. Station Distribution of NYC Citi Bike System.

both geographically and temporally. The unbalanced bike flows will cause full station or empty station status and require more operation cost to rebalance the inventory level. We investigate the station balance problem by first introducing the concept of station positive unbalance $s_i.pu$ and negative unbalance $s_i.nu$ based on accumulated station net flow.

Definition 2: Station unbalance. Let $s_i.pd(t)$ and $s_i.dd(t)$ represent the pick-ups and drop-offs at station s_i during time slot t, the station unbalance is defined as the maximum accumulate bike net flow during the day. The positive unbalance and negative unbalance are defined as the contiguous subarray of series $\{s_i.dd(t) - s_i.pd(t)\}$ whose values have the largest positive sum and smallest negative sum. The station positive unbalance $s_i.pu$ and negative unbalance $s_i.nu$ are formally defined as follows:

$$s_i.pu = \max_{t_i,t_j}\{\sum_{t=t_i}^{t_j} s_i.dd(t) - s_i.pd(t), \quad t_i < t_j\} \tag{3.2}$$

$$s_i.nu = \max_{t_i,t_j}\{\sum_{t=t_i}^{t_j} s_i.pd(t) - s_i.dd(t), \quad t_i < t_j\} \tag{3.3}$$

Ideally, a self-balanced station with balanced bike flow will make $s_i.pu$ and $s_i.nu$ close to 0. However, most bike stations in NYC are far from balanced status. The NYC Citi Bike station daily averaged positive unbalance and negative unbalance distributions are presented in Figure 3.1(d) and Figure 3.1(e) as an example.

Functional Zone

Since need-based customers will choose the station closest to their current locations or final destinations, we partition the bike station in service area using a Voronoi-based gridding method (Aurenhammer, 1991), from which the map is partitioned into regions based on walking distance to bike stations. Each grid is centered by one bike station and the points within one region is closest to its center. As a result, pick-up/drop-off points for taxi trips and POIs are mapped to the nearest bike station.

Definition 3: Voronoi Region. Let X be a space coordinate endowed with a walking distance wd extracted from Google Maps Distance Matrix API. The Voronoi region R_{s_i} associated with station s_i is the set of points in X whose distance to s_i is no greater than that to other stations:

$$R_{s_i} = \{x \in X | wd(x, s_i) \leq wd(x, s_j), \forall j \neq i\} \tag{3.4}$$

Definition 4: Functional Zone. A functional zone Z_K is comprised of a group of regions $\{R_{s_i}^K\}$ with similar urban functions identified by the distribution of socioeconomic activities (Yuan et al., 2015). Each functional zone has its major category characterized by its POI structure. For example, the commercial zones have a lot of shopping centers while the transportation junctions have many transportation centers compared to other functional zones.

3.3.2 Problem Definition

Expansion Station Bike Demand Prediction. Given a set of existing principle bike station locations S_l^p and a set of expansion station locations S_l^e, the problem of expansion station bike demand prediction is to predict the hourly pick-up (drop-off) demand $s_i.pd(t)$ $(s_i.dd(t))$ of the expanded station (including principle stations along the edge of expansion area and coverage expansion stations) during a day.

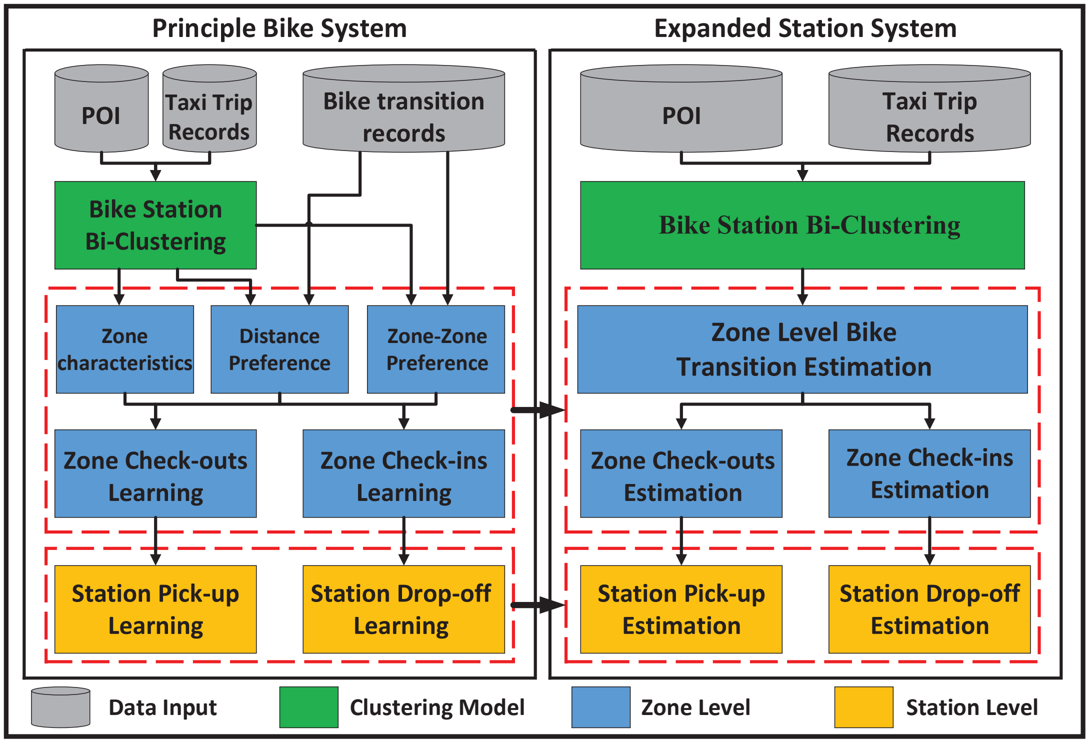


Figure 3.2. Framework Overview

Station unbalance Prediction. Once the hourly bike demand is estimated, we can further estimate the station unbalance characteristic according to definition 2.

3.3.3 Framework Overview

Figure 3.2 shows the framework overview of our proposed method which consists of three major sections: functional zone based bike station Bi-Clustering, Zone level bike transition prediction and station level bike demand prediction.

Functional zone based station clustering. We first propose a functional zone based Bi-clustering algorithm to cluster stations into different groups based on their Voronoi region POI structures (Aurenhammer, 1991; Liu et al., 2015) and station locations. The stations within one functional zone are close to each other and designed to serve for the same functional zone customers.

Zone level bike transition prediction. The zone level bike transition prediction integrates the bike trip distance preference, zone-to-zone transition preference and zone characteristics to predict zone check-ins and check-outs based on the Random Forest predictor.

Station level bike demand prediction. The station level bike demand prediction is to distribute the inner zone bike check-ins and check-outs to individual stations based on their covered resources (POI densities).

3.4 Methodology

3.4.1 Principle station network learning

The principle station network learning studies the station level bike demand prediction model by analyzing the historical bike transition records of the principle station network in 3 steps: 1. functional zone identification; 2. zone-to-zone bike transition learning and 3. inner zone station level bike demand prediction.

Functional zone identification. We first discuss how to divide the whole bike sharing system in service area into many functional zones, where a functional zone is a subregion contains several bike stations and their associated Voronoi regions, the stations in the same functional zone have similar POI distribution and close geographical locations.

Assume the POI matrix of bike stations is $\mathbf{P} = \{p_{ij}\}$, where p_{ij} is an indicator of *j-th* type of POIs of bike station i. POI matrix $\mathbf{P}$ is derived from POI counts in Voronoi regions of stations, which is essentially a POI heat matrix of the bike stations. This paper considers many types of POIs. Some types of POIs have similar geographical distribution, for example, pharmacy and convenience store, they may have a symbiotic relationship; while the other types of POIs show quite different geographical distributions, e.g., financial service or car rental service. So it is very important to study the relationships between different types of POIs before partition the studied region into functional zones.

Algorithm 5 presents a novel Bi-clustering algorithm which clusters the bike stations and POI features alternatively. In step 1, we get initial station clustering result and POI feature clustering result at the same time. Then we construct virtual bike stations in step 4∼6, and get new clustering result of POI features (or POI category) in step 7. Step 8 is a break condition, where NMI is a popular information-based evaluation metric of clustering results, which describes the coherence of two clustering results(Vinh, Epps, & Bailey, 2009). In step 12∼14, we use new POI features to represent each station, where new POI features are generated according to current clustering result of POI categories. Step 15 gets new station clustering result according to stations in new feature space. K^f and K^s are the number of station clusters and that of feature clusters respectively. From a theoretical view, the setting of the number of clusters is essentially a balance of fitting error and model complexity; while in practical applications, we set the parameters according to data volume. In this problem, K^s is set as approximately 10% number of stations, K^s is about 10% number of POI types. Table 3.1 shows the Bi-clustering result of the studied POI categories, where two POI categories are assigned into the same cluster means the two POI types have similar geographical distribution and symbiotic relationship. Take 4-*th* cluster, for example, pharmacy, grocery, and store, often locate together,

```
Require: Input: $\mathbf{P}$,$K^f$,$K^s$,$ItrMax$;
$Itr = 0$, $\mathbf{c}^s = kmeans(\mathbf{P}, K^s)$, $\mathbf{c}_0^f = kmeans(\mathbf{P}^T, K^f)$;
while $Itr < ItrMax$ do
    $Itr = Itr + 1$;
    for $i = 1 : K^s$ do
        $Idxs = find(\mathbf{c}^s = i)$;
        $\mathbf{x}_{i\cdot}^f = mean(\mathbf{P}_{Idxs\cdot}, row)$;
    $\mathbf{c}^f = kmeans(\mathbf{X}^f, K^f)$; % $\mathbf{X}^f = \{\mathbf{x}_{i\cdot}^f\}^T$.
    if $NMI(\mathbf{c}^f, \mathbf{c}_0^f) = NMI(\mathbf{c}^f, \mathbf{c}^f)$ then
        $Break$;
    else
        $\mathbf{c}_0^f = \mathbf{c}^f$.
    for $j = 1 : K^f$ do
        $Idxf = find(\mathbf{c}^f = j)$;
        $\mathbf{x}_{\cdot j}^s = mean(\mathbf{P}_{\cdot Idxf}, col)$;
    $\mathbf{c}^s = kmeans(\mathbf{X}^s, K^s)$; % $\mathbf{X}^s = \{\mathbf{x}_{\cdot j}^s\}$.
```

Algorithm 5: *BiC-POIs*($\mathbf{P}$,K^f,K^s,$ItrMax$)

while the 3-*rd* cluster indicates the geographical distribution of taxi pickups and taxi dropoffs are almost the same.

Table 3.1. Clustering result of POI categories

Cluster	POI categories
1^{st}	subway, transit station, train station, finance, ...
2^{nd}	park, museum, bus station, amusement park, ...
3^{rd}	taxi pickup, taxi dropoff, ...
4^{th}	pharmacy, grocery, supermarket, store, hair care, school, shopping mall, florist, lodging, doctor, ...
5^{th}	food, cafe, bar, night club, church, spa, ATM, ...
6^{th}	parking, car rental, car wash, repair, car dealer, ...

After clustering the original POI categories into several groups, we generate new POI features based on the clustering result. The new POI heat matrix is represented by $\mathbf{H} = \{h_{ij}\}$, h_{ij} represents the heat of bike station i w.r.t. *j-th* category of

POIs. We will discover functional zones by clustering bike stations according to their geographical locations and POI heats. However, it is a very challenging clustering problem, most of the existing clustering algorithms cannot be employed in this task because: 1) this problem requires the consideration of both geographical locations and POI features of bike stations; 2) similarities between objects are required to be predefined in most of the previous methods, and the definitions of similarities usually mix all the features of objects together. Therefore, the POI characteristics of a bike station will fade away as a result of average effect. For example, if the POI heat of a bike station is $\mathbf{h}_i = [1, 0, \cdots, 0]$, it should become a representative station in a functional zone as it has the highest heat value of the first POI type. However, if we cluster stations according to similarity defined by station feature vectors like $\|\mathbf{h}_i - \mathbf{h}_j\|$, it will be very difficult to identify representative stations with distinguished POI characteristics. That is why we develop a novel Heat Peaks based Clustering (HPC) algorithm in this paper. Algorithm 6 presents the proposed HPC method.

The core of Algorithm 6 is $PeakDiscovery(\mathbf{h}, \mathbf{D}, K)$, which is used in step 3 and 25. This function finds K heat-peak stations according to distribution of POI heat $\mathbf{h}$ and geographical distances between stations $\mathbf{D}$, where a heat-peak station satisfies two conditions: 1) it has relatively larger POI heat value, and 2) there is no station with even higher heat value in its neighborhood. Besides heat value h_i, the other indicator of $i\text{-}th$ station is defined as

$$\gamma_i = \min_{j, h_j \geq h_i} D_{ij}. \tag{3.5}$$

```
Require: Input: $\mathbf{H}$,$\mathbf{D}$,$NP^0$,$NP^I$,$\delta$,$NCMin$;
 1: $PeaksAll = \emptyset$; $\mathbf{c}^s = \mathbf{0}$;
 2: for $f = 1 : N^f$ do
 3:     $PeaksI = PeakDiscovery(\mathbf{H}_{\cdot f}, D, NP^0)$;
 4:     $PeaksAll = PeaksAll \cup PeaksI$;
 5: $PeaksAll0 = PeaksAll$;
 6: while $\exists\ c_i^s = 0$ do
 7:     for $i = 1 : N$ do
 8:         if $i \in PeaksAll$ then
 9:             $c_i^s = i$;
10:         else
11:             $PeaksNI = \{j | D(i,j) \leq \delta, j \in PeaksAll\}$;
12:             if $PeaksNI \neq \emptyset$ then
13:                 $c_i^s = \underset{k \in PeaksNI}{\arg\max}\ S^f(i,k)$
14:     for $k = 1 : NP^t$ do
15:         $Cluster^s(k) = \left\{i | c_i^s = PeaksAll(k), i = 1, 2, \cdots, N\right\}$
16:         if $|Cluster^s(k)| \leq NCMin$ then
17:             $c_i^s = 0$, $i \in Cluster^s(k)$;
18:             $PeaksAll = PeaksAll \backslash PeaksAll(k)$;
19:     if $PeaksAll = PeaksAll0$ then
20:         $Break$;
21:     else
22:         $PeaksAll0 = PeaksAll$;
23:     $I = find(\mathbf{c}^s = 0)$, $\mathbf{H}^u = \mathbf{H}_{(I,\cdot)}$, $\mathbf{D}^u = \mathbf{D}_{(I,I)}$;
24:     for $f = 1 : N^f$ do
25:         $PeaksI = PeakDiscovery(\mathbf{H}^u_{\cdot f}, D^u, NP^I)$;
26:         $PeaksAll = PeaksAll \cup PeaksI$;
27: for $i = 1 : N$ do
28:     if $c_i^s = 0$ then
29:         $c_i^s = \underset{k \in PeaksAll}{\arg\max}\ S^f(i,k)$;
```

Algorithm 6: HPC($\mathbf{H}$,$\mathbf{D}$,NP^0,NP^I,δ,$NCMin$)

$PeakDiscovery(\mathbf{h}, \mathbf{D}, K)$ is to pick out K stations with largest η values, where $\eta_i = h_i \cdot \gamma_i$. It can be known that a heat-peak station selected by our method is a station with highest heat value in a relative large region.

In Algorithm 6, step 2∼4 discover the first batch of heat-peak stations, which select NP^0 heat-peak stations from each of N^f POI categories. In the following, step 7∼13 assign each bike station a cluster label by first finding heat-peak stations in its δ-neighborhood, then assigning the stations to a heat-peak station with the most similar POI mode. If there are no heat-peak stations in its δ-neighborhood, the cluster label of the station is set 0. Step 14∼18 reset the cluster label of a station as 0 if the scale of the cluster it belongs to is less than $NCMin$. In step 23∼26, we find new heat peaks from unlabeled bike stations. Repeat 7∼26 till there are no new heat-peak stations added in an iteration. The residual unlabeled bike station is finally assigned in step 27∼29.

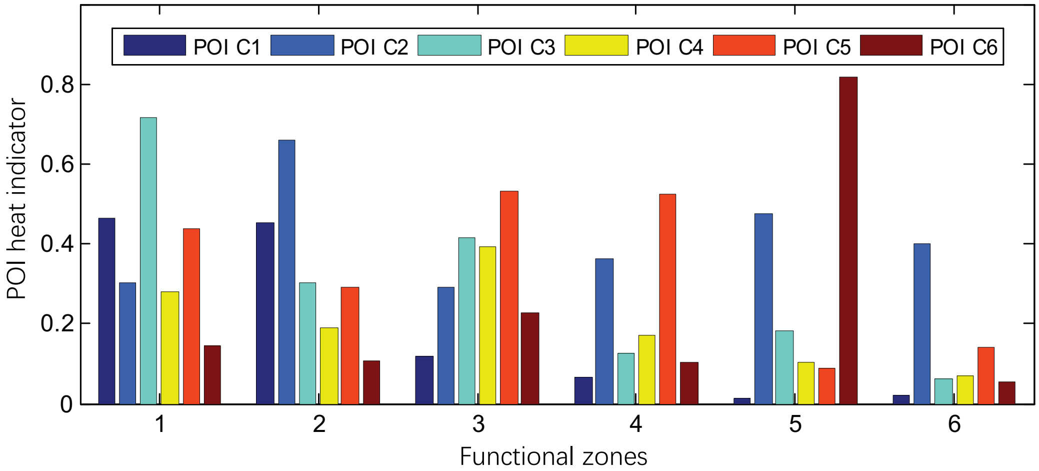
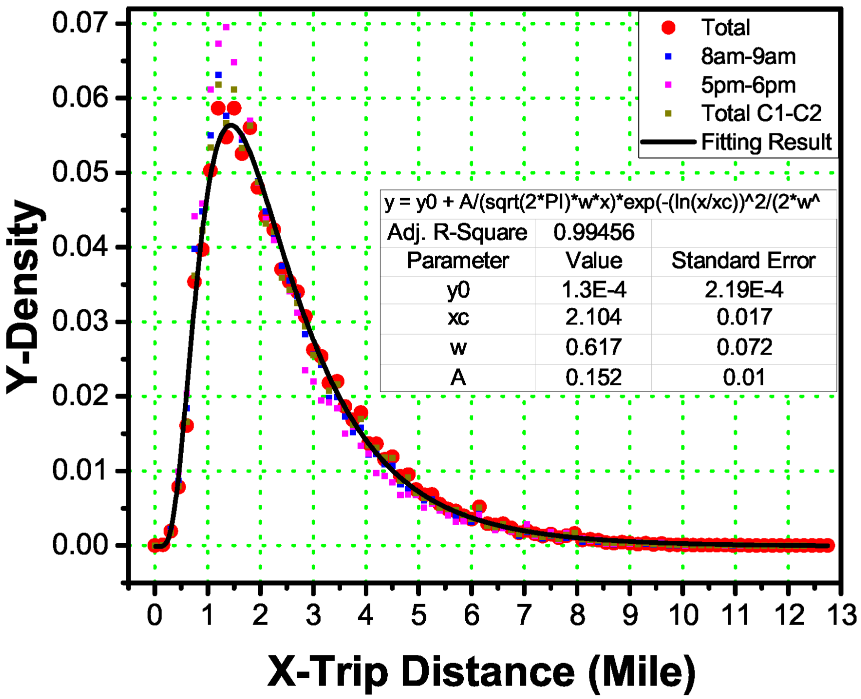


Figure 3.3. POI characteristics of 6 FZ categories.

We partition the current Citi Bike in service area into 6 functional zone categories. Figure 3.3 shows POI characteristics of the 6 functional zones categories. The first category can be defined as a mixed business zone as it contains a balanced high density of POIs from the first, third, and fifth POI category, while the third category is the residential area, which contains a large POIs density like grocery, pharmacy, and food. The other categories also have distinguished POI characteristics that can be categorized as transportation area (second functional zone category), scenic spots (fourth functional zone category), car services area (5th functional category) and education area and Park areas (the park zones that have few POIs).

It can be known that the proposed HPC algorithm can discover functional zones with the consideration of both geographical locations and POI characteristics of bike stations. The identified functional zones consist of bike stations with similar POI



y = y0 + A/(sqrt(2*PI)*w*x)*exp(-(ln(x/xc))^2/(2*w^

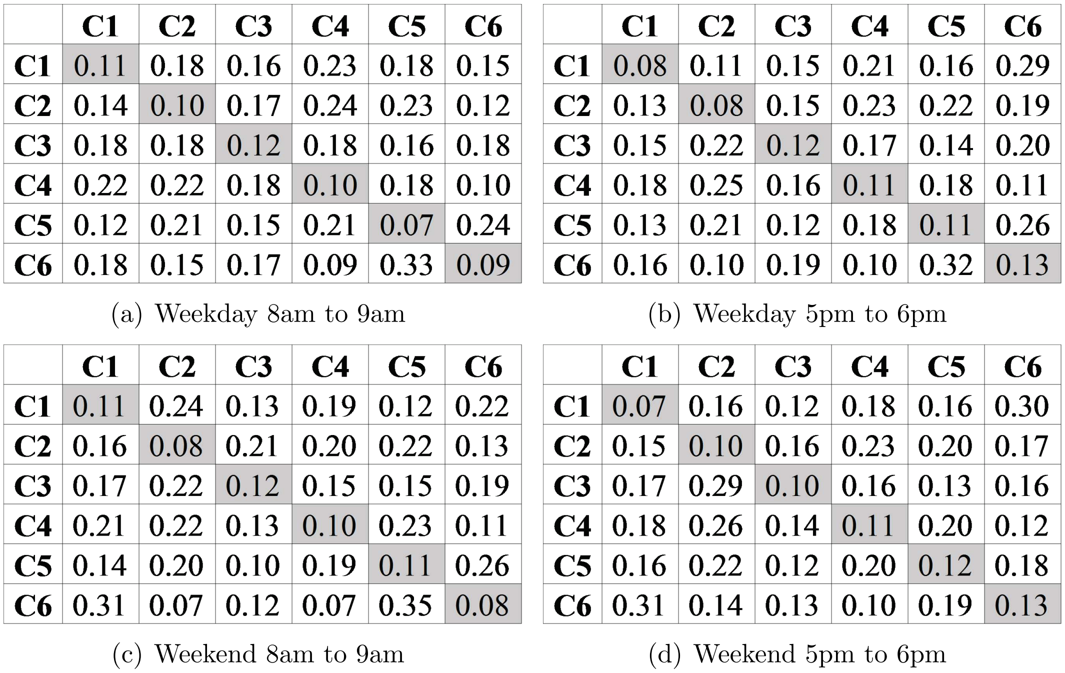
Adj. R-Square	0.99456	
Parameter	Value	Standard Error
y0	1.3E-4	2.19E-4
xc	2.104	0.017
w	0.617	0.072
A	0.152	0.01

Figure 3.4. Bike Transition Distance Preference

mode and close geographical distance. Demand prediction of bike station can not only benefit from POI features of functional zones, but also from zone-to-zone transition patterns.

Zone-to-Zone transition learning. The zone-to-zone transition learning focuses on the three major factors that would affect the check-outs and check-ins of each functional zone: trip distance preference, zone-to-zone preference, and zone characteristics.

Distance Preference Learning. The distance preference refers to the distance range that a person prefers to taking a bike other than other transportation methods such as subways or taxis. Mathematically, the pick-up frequency density versus transition distance forms a log-normal distribution (see the blue fitting line in Figure 4.2). As can be seen, the bike transition distance distributions are identical during different



	C1	C2	C3	C4	C5	C6
C1	0.11	0.18	0.16	0.23	0.18	0.15
C2	0.14	0.10	0.17	0.24	0.23	0.12
C3	0.18	0.18	0.12	0.18	0.16	0.18
C4	0.22	0.22	0.18	0.10	0.18	0.10
C5	0.12	0.21	0.15	0.21	0.07	0.24
C6	0.18	0.15	0.17	0.09	0.33	0.09

(a) Weekday 8am to 9am

	C1	C2	C3	C4	C5	C6
C1	0.08	0.11	0.15	0.21	0.16	0.29
C2	0.13	0.08	0.15	0.23	0.22	0.19
C3	0.15	0.22	0.12	0.17	0.14	0.20
C4	0.18	0.25	0.16	0.11	0.18	0.11
C5	0.13	0.21	0.12	0.18	0.11	0.26
C6	0.16	0.10	0.19	0.10	0.32	0.13

(b) Weekday 5pm to 6pm

	C1	C2	C3	C4	C5	C6
C1	0.11	0.24	0.13	0.19	0.12	0.22
C2	0.16	0.08	0.21	0.20	0.22	0.13
C3	0.17	0.22	0.12	0.15	0.15	0.19
C4	0.21	0.22	0.13	0.10	0.23	0.11
C5	0.14	0.20	0.10	0.19	0.11	0.26
C6	0.31	0.07	0.12	0.07	0.35	0.08

(c) Weekend 8am to 9am

	C1	C2	C3	C4	C5	C6
C1	0.07	0.16	0.12	0.18	0.16	0.30
C2	0.15	0.10	0.16	0.23	0.20	0.17
C3	0.17	0.29	0.10	0.16	0.13	0.16
C4	0.18	0.26	0.14	0.11	0.20	0.12
C5	0.16	0.22	0.12	0.20	0.12	0.18
C6	0.31	0.14	0.13	0.10	0.19	0.13

(d) Weekend 5pm to 6pm

Figure 3.5. Zone-to-Zone Transition Matrix

time periods and between different functional zones. Therefore, given the locations of origin $o.c$ and destination $d.c$, associated with their distance $x \equiv \|o.c - d.c\|$, we can estimate the users' distance preference of taking bicycles, which is defined by a Distance Preference Score (DPS):

$$DPS(x) = y_0 + \frac{A}{\sqrt{2\pi}wx} exp(-\frac{(ln(x/x_c))^2}{2w^2}) \tag{3.6}$$

Where y_0, A, w, x_c are fitting parameters (see fitting results in inserted table of Figure 4.2). The formula of DPS indicates that people would not like to take bikes for long term trip (larger than 4 miles) or within walking distances. People who have an origin-to-destination distance in the range of $FWHM$ (full width at half maximum) of DPS (1.5 miles $\sim$ 2.7 miles) are more willing to take bicycles.

Zone-to-zone preference learning. Besides the distance preference, customers have their functional zone preference in different periods. For example, during AM rush hour, customers will have a high preference to take bikes from subways exits to business areas even though a nearby parking lot has a higher DPS. Here we define a functional zone transition matrix $T(C_i, C_j)$ to describe the transition preference from functional zones of class C_i to functional zones of class C_j that satisfies:

$$T(C_i, C_j) \sum_{n=1}^{N} DPS(x; o_n.c \in C_i, d_n.c \in C_j) = N \tag{3.7}$$

Where N represents the total transitions from functional zone of class C_i to C_j. Table 3.5 presents the normalized FZ transition preference matrix of 4 time periods: weekday 8 am-9 am 3.5(a), weekday 5 pm-6 pm 3.5(b), weekend 8 am-9 am 3.5(c) and weekend 5 pm-6 pm 3.5(d). As can be seen, bike users are least likely to move between the functional zones of the same class (small diagonal value of T), which indicates that in order to motivate more bike users, the functional zones should be diversified. The transition matrix varies in different time periods and some classes have high links in different time periods.

Therefore, given two functional zone Z_m, Z_n with its location center $Z_m.l, Z_n.l$ and class $Z_m.C, Z_n.C$, we define the Zone Transition Score ZTS of time period t as follows:

$$ZTS(Z_m, Z_n; t) = T(Z_m.C, Z_n.C; t)DPS(|Z_m.l - Z_n.l|) \tag{3.8}$$

The functional zone check-out preference score ZPS_{out} and check-in preference score ZPS_{in} are then defined by considering the transition scores to and from all surrounding functional zones:

$$ZPS_{out}(m;t) = \sum_{n \neq m} ZTS(Z_m, Z_n; t) \tag{3.9}$$

$$ZPS_{in}(m, in; t) = \sum_{n \neq m} ZTS(Z_n, Z_m; t) \tag{3.10}$$

Zone Characteristics. The last factor considered in our prediction model that could affect the bike demands is the characteristics of each functional zone, including the densities of 6 major POI categories, historical taxi check-outs and check-ins, and the number of available docks.

Entire traffic prediction. After we extract the most influential factors from the zone characteristics, zone check-out and check-in preference scores, the entire check-outs $Z_{out}(m;t)$ and check-ins $Z_{in}(m;t)$ are predicted by feeding the factors into the Random Forest Regressor (RF) for different time periods.

Station level bike demand prediction. After we predict the check-ins and check-outs of each Functional zone, the station bike demand in the functional zone is predicted by ridge regression $f^C(s_i.F)$, indicating the number of check-ins or check-outs distributed to each station based on the station level feature vector F (Voronoi area POI densities and their distance to nearest transportation entrances, such as parking lots, subway entrances and bus stops). The logistic regressor for pick-up demand and drop-off demand are trained by the historical transition records of stations within the same functional zone of category C. For each station s_i located in Functional Zone Z_m of category C, the station level pick-up $s_i.pd$ and drop-off demand $s_i.dd$ are formally predicted as follows:

$$s_i.pd(t) = Z_{out}(m;t)\frac{f_p^C(s_i.F)}{\sum_{s_j \in Z_m} f_p^C(s_j.F)} \tag{3.11}$$

$$s_i.dd(t) = Z_{in}(m;t)\frac{f_d^C(s_i.F)}{\sum_{s_j \in Z_m} f_d^C(s_j.F)} \tag{3.12}$$

3.4.2 Demand prediction after expansion

There are two kinds of stations for expansion: station coverage expansion and complementary station expansion. Different kinds of expansion strategy have different effects on the demand related factors.

The station setup for coverage expansion is to set up new stations in the area that has no stations before. As a result, the new functional zones of the expansion areas will affect the zone-to-zone connection preference and transition distance preference of the principle system. By the definition of Zone Transition Score ZTS, which decreases fast for long distance transportation, the coverage expansion stations will have fewer effects on the functional zones located far away compared to the functional zones located near the expansion edges. The complementary station expansion aims at reducing the station workload by adding one or more stations to existing bike sharing system covered functional zones. The complementary stations have fewer effects on zone level bike check-ins/check-outs predictions but will redistribute the bike pick-ups and drop-offs within the functional zones.

Although different expansion strategies have different effects, these effects are reflected in the changes of our predictor input feature vectors. To predict the station demand prediction after expansion, we reconstruct the input features at zone level and station level after expansion and implement the predictors we have trained in the principle station network learning.

3.5 Experimental Results

To validate the efficiency and effectiveness of our proposed method, extensive experiments are performed on real world NYC CitiBike trip data of three different time periods. The first stage is the principle bike station system consisting of 329 stations from 07/01/2013 to 07/31/2015. The second stage has 486 stations from the completion of first expansion on 08/06/2015 to 07/18/2016. And the third stage starts from 07/23/2016 to 11/30/2016, with 617 stations in service after the second expansion. All experiments are conducted on a PC 7 with an Intel(R) Core i7-4790 CPU, 3.6 GHz, and 16 GB RAM running 64-bit Windows 10 system.

3.5.1 Experimental Data

We conduct our experiments with bike sharing system data, Google Place API[1] , taxi trip records from NYC with their statistics presented in Table 3.2. Citibike transition records are generated by NYC Bike Sharing System which is public available from Citibike official website [2] . This data set contains the following information: station id, bicycle pick-up station, bicycle pick-up time, bicycle drop-off station and bicycle drop-off time. In addition, the station status is crawled every 10 minutes from station status feed site [3] which contains the information of station in service status, currently

[1] https://developers.google.com/places/

[2] https://www.citibikenyc.com/system-data

[3] https://feeds.citibikenyc.com/stations/stations.json

Table 3.2. Details of the datasets

Data Source	**New York City Bike System**		
Time from	7/1/13	8/6/15	7/29/16
to	7/31/15	7/18/16	11/30/16
Weekdays	524	238	85
(Weekends)	(237)	(110)	(40)
#Stations	329	486	617
#Records	17.58 million	11.76 million	6.07 million
Data Source	**Google Place API**		
POI type	number	POI type	number
establishment	70335	car service	1088
education	2784	supermarket	4077
shopping mall	206	entertainment	996
store	28418	bus station	1981
lodging	1262	railway station	1142
home service	1166	finance	8103
convenience	9914	estate agency	5693
health center	42164	restaurant	11825
night life	4115	travel agency	1595
fitness	1357	···	···
Data Source	**New York City Taxi Trip Records**		
effective days	time Period	# of trip records	
31	08/2013	12.6 million	

available bikes and station capacity.

3.5.2 Baselines & Metric

The methods proposed in our work to predict the station level pick-up demand and drop-off demand are denoted as Functional Zone based Random Forest Regressor (**FZ+RF**). In order to confirm the effectiveness of our models, we conduct experiments to compare our methods with the following baselines:

Station Level Predictor (Li et al., 2015b): The station level predictors estimate the bike demand based on a set of global feature elements. The baselines of station level predictors we use in this paper include Random Forest **(RF)**, K-Nearest Neighbor Regressor **(KNN)**, Neural Network (NN) and Gradient Boosting Regressor **(GBR)**. The features used for station level predictors include the 19 fine grained POI densities, Voronoi region taxi check-ins and check-outs.

Hierarchical Demand Predictor: Considering our Functional Zone based station clustering is the first attempt, we use the **FZ+GBR** as a baseline. The only difference between the **FZ+GBR** and our method is that it uses GBR for zone level bike transitions prediction.

Metric: The metrics we adopt to measure the performance are the Error Rate *ER* and Root Mean Squared Logarithmic Error *RMLSE*, which are formally defined as follows:

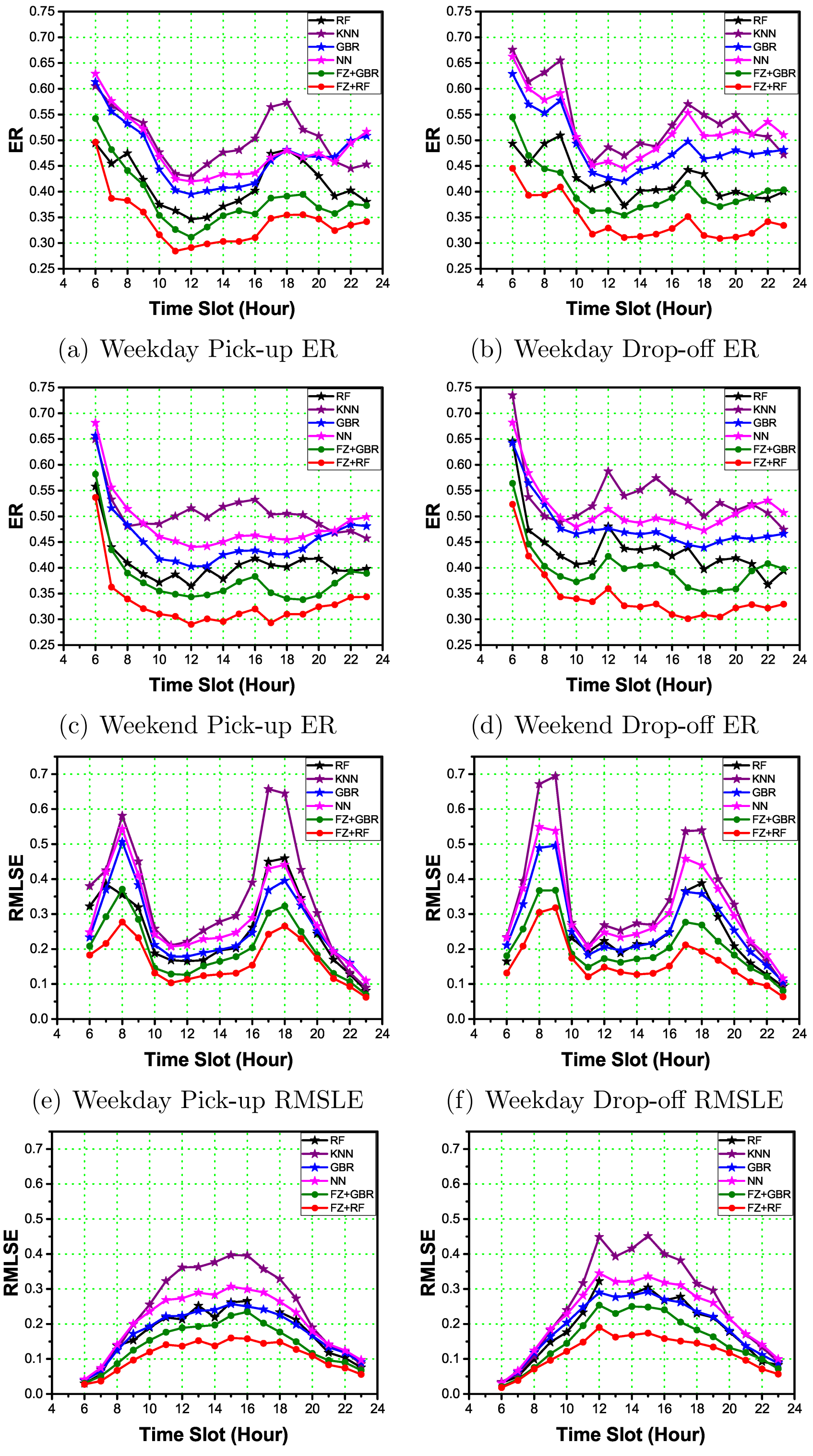
$$
\begin{aligned}
ER(t) =& \frac{\sum_{i=1}^{N} |\hat{s}_i.d(t) - s_i.d(t)|}{\sum_{i=1}^{N} s_i.d(t)} \\
RMLSE(t) =& \sqrt{\frac{1}{N}\sum_{i=1}^{N}(log(\hat{s}_i.d(t)+1) - log(s_i.d(t)+1))^2}
\end{aligned}
$$

Here $s_i.d(t)$ is our ground truth of bike pick-up or drop-off demand of station i during time slot t and $\hat{s}_i.d(t)$ is the corresponding prediction value. The principle stations in Stage 1 are used as training set. For the expansion analysis of Stage 2 and Stage 3 bike sharing systems, only the bike stations in the expansion areas, the stations located within the functional zones taht are adjacent to the expansion boundaries, and the complementary stations are included in the testing set.

3.5.3 Demand Prediction

Hourly station bike demand prediction after first expansion. The performance comparison for first expansion bike demand prediction (including weekday pick-up demand, weekday drop-off demand, weekend pick-up demand and weekend drop-off demand) between our proposed FZ+RF and baselines is summarized in Figure 3.6. From Figure 3.6, we can see that for all time periods, both of the Error Rate (ER) and the Root Mean Squared Logarithmic Error (RMSLE) obtained from our proposed method are much lower than all the baselines with a significant margin. Moreover, the hierarchical demand predictor based on functional zone station clustering (FZ+GBR and FZ+RF represented by dot lines) achieve a better performance than station level predictors based on global features (represented by star symbol lines). The high ER of early morning predictions is mainly due to the few transition records which amplify the ER but leads to a very small RMSLE.

Hourly station bike demand prediction after second expansion. Figure 3.7 presents the performance comparison for bike demand prediction after the second system expansion. As can be seen, our proposed method lower the ER and RMSLE



(a) Weekday Pick-up ER (b) Weekday Drop-off ER

(c) Weekend Pick-up ER (d) Weekend Drop-off ER

(e) Weekday Pick-up RMLSE (f) Weekday Drop-off RMLSE

(g) Weekend Pick-up RMLSE (h) Weekend Drop-off RMLSE

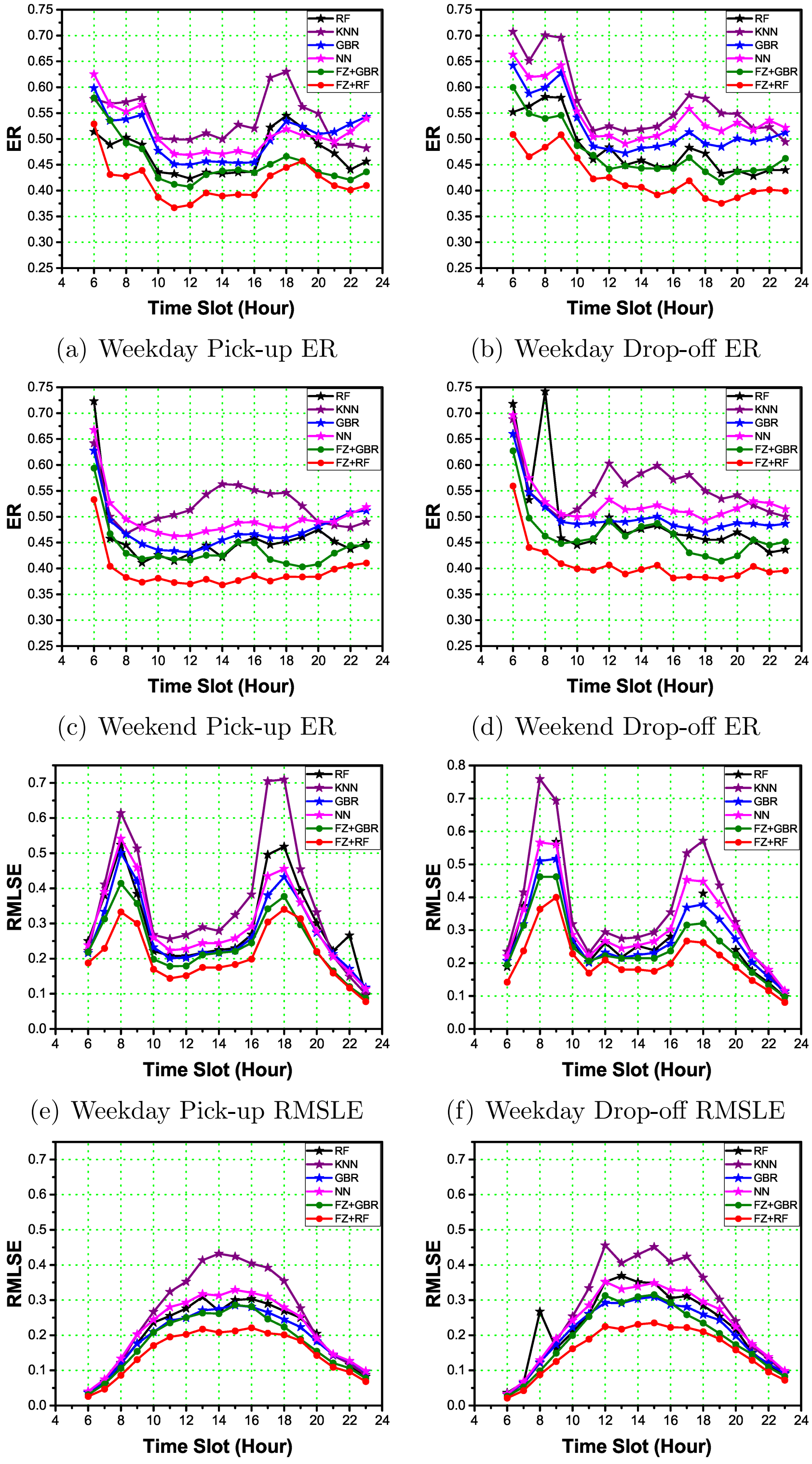
Figure 3.6. Performance comparison of station bike demand prediction after first expansion.

of bike demand predictions of different time periods. However, compared to the bike demand prediction performance after the first expansion, the ER and RMSLE increase. It might be due to the reason that the functional zones in the second expansion areas have a larger difference compared to that in the principle area.

Overall Performance Comparison. The daily averaged pick-up demand, drop-off demand, positive balance and negative balance prediction accuracy comparisons are represented in Figure 3.8. For the first stage expansion, our method achieves an overall pick-up ER of 0.3118 which is 0.0482 lower than the other hierarchical demand predictor (stage 1) and 0.0863 lower than the most competitive station level predictor RF. The overall drop-off demand ER of our method is 0.3295 which is much lower than other baselines. In terms of RMLSE, our method achieves an overall RMLSE of 0.1096, 0.1184, 0.1509 and 0.157 for stage-1 pick-up demand prediction, stage-1 drop-off demand prediction, stage-2 pick-up demand prediction and stage-2 drop-off demand prediction respectively. Moreover, an accurate hourly demand prediction can also benefit the station unbalance prediction. The Figure 3.8(c) and Figure 3.8(d) summarize the performance of the positive unbalance and negative unbalance prediction. As can be seen, our proposed method can provide a more accurate unbalance status prediction which can further help bike sharing system designers estimate the rebalancing operation cost after bike sharing network expansion.

3.6 Conclusion

In this paper, we developed a hierarchical bike demand prediction models for expansion area station level bike demand prediction. Specifically, we first partitioned



(a) Weekday Pick-up ER

(b) Weekday Drop-off ER

(c) Weekend Pick-up ER

(d) Weekend Drop-off ER

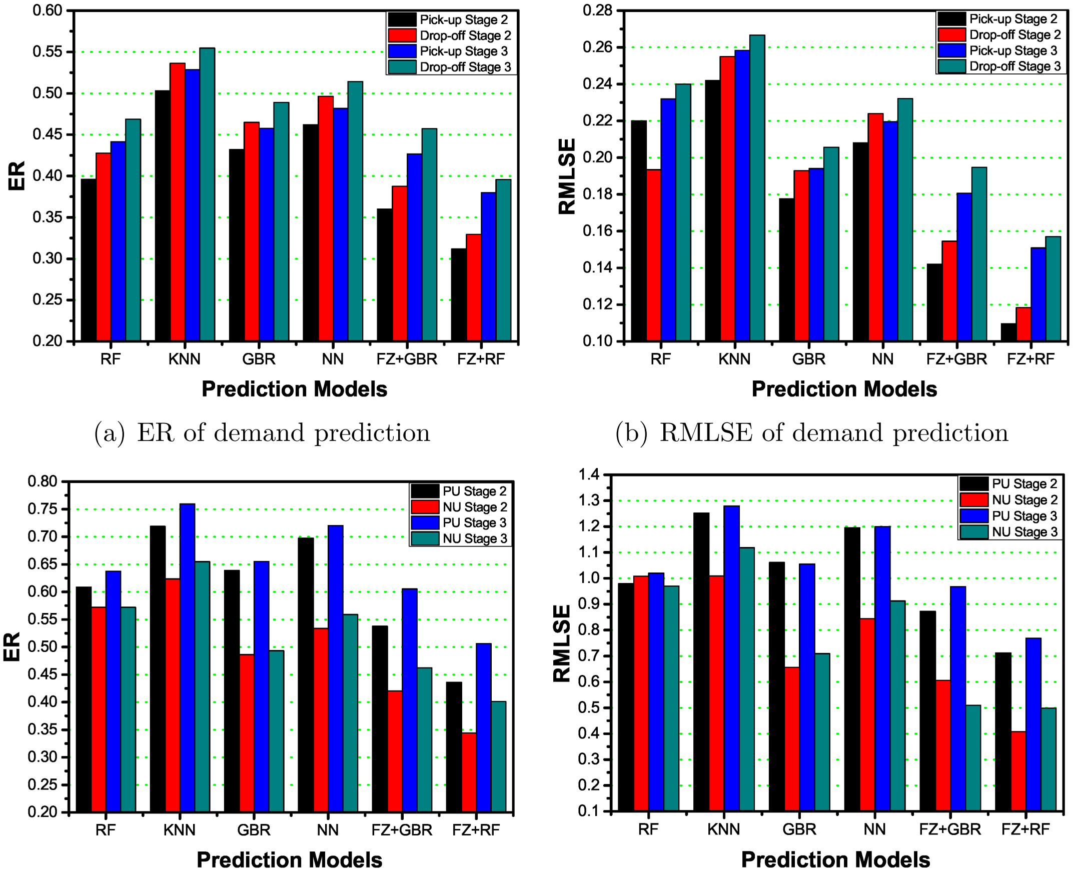
(e) Weekday Pick-up RMLSE

(f) Weekday Drop-off RMLSE

(g) Weekend Pick-up RMLSE

(h) Weekend Drop-off RMLSE

Figure 3.7. Performance comparison of station bike demand prediction after second expansion.



(a) ER of demand prediction

(b) RMLSE of demand prediction

(c) ER of balance prediction

(d) RMLSE of balance prediction

Figure 3.8. Overall Performance Comparison

the station in service area into different functional zones based on our Bi-Clustering algorithm. Then based on the functional zones, we implemented Random Forest Regressor to estimate the functional zone bike transitions by integrating the bike trip distance preference, zone-to-zone preference, and zone characteristics. The station level bike demand was predicted by distributing the zone level check-ins and check-outs to each station with the consideration of their Voronoi region POI structures. Finally, the extensive experiments on real-world data from the 3-stage NYC Citi Bike System showed the advantages of our hierarchical strategy of bike demand prediction for bike sharing system expansion.

CHAPTER 4

STATION SITE OPTIMIZATION IN BIKE SHARING SYSTEMS

In an ideal bike sharing network, the station locations are usually selected in a way that there are balanced pick-ups and drop-offs among stations. This can help avoid expensive re-balancing operations and maintain high user satisfaction. However, it is a challenging task to develop such an efficient bike sharing system with appropriate station locations. Indeed, the bike station demand is influenced by multiple factors of surrounding environment and complex public transportation networks. Limited efforts have been made to develop demand-and-balance prediction models for bike sharing systems by considering all these factors. To this end, in this paper, we propose a bike sharing network optimization approach by considering multiple influential factors. The goal is to enhance the quality and efficiency of the bike sharing service by selecting the right station locations. Along this line, we first extract fine-grained discriminative features from human mobility data, point of interests (POI), as well as station network structures. Then, prediction models based on Artificial Neural Networks (ANN) are developed for predicting station demand and balance. In addition, based on the learned patterns of station demand and balance, a genetic algorithm based optimization model is built to choose a set of stations from a large number of candidates in a way such that the station usage is maximized and the number of unbalanced stations is minimized. Finally, the extensive experimental results on the

NYC CitiBike sharing system show the advantages of our approach for optimizing the station site allocation in terms of the bike usage as well as the required re-balancing efforts.

4.1 Introduction

To offer immediate and convenient access, a network of bike docking stations are positioned throughout an urban area. However, developing an efficient bike sharing system with proper station locations is a challenging task. To construct a successful bike sharing network, we must consider the station locations in the bike sharing network and their relationship with trip demand and balance (García-Palomares, Gutiérrez, & Latorre, 2012; Martinez, Caetano, Eiró, & Cruz, 2012; Contardo, Morency, & Rousseau, 2012). Specifically, there are two major challenges for bike station site selections. First, bike sharing system is an undirected network that the performance (i.e., bicycle demand) of one station highly depends on its connection to other stations and its surrounding human activities. The multi-factor effects of surrounding environment and station network structure make it difficult to predict station demand. Second, the demand distribution is unbalanced both geographically and temporally. It is costly to dispatch bikes from full stations to empty stations for re-balancing, and the efficiency of the station usage is reduced during the unavailable period.

Recently, a number of researches on bike sharing systems analysis have been conducted from different aspects. Most of the studies have focused on the historic development of bicycle sharing system (Shaheen, Guzman, & Zhang, 2010), promotion strategies (Pucher, Garrard, & Greaves, 2011), bicycle temporal and geographical usage patterns analysis (Kaltenbrunner, Meza, Grivolla, Codina, & Banchs, 2010b), station demand related factors (Pucher, Dill, & Handy, 2010) and re-balancing bicycles among established bike stations (Rainer-Harbach, Papazek, Hu, & Raidl, 2013; Kloimüllner, Papazek, Hu, & Raidl, 2014). However, there are relatively few studies quantitatively addressing the relationship between the multiple influential factors and the station demand or its geographically imbalance distribution.

To solve the aforementioned challenges, in this paper, we first extract insightful features from human mobility data, POIs and bike station network structures. Next, we propose an Artificial Neural Network based prediction model for station demand and balance prediction according to the features extracted. Then an optimization problem aiming at maximizing station demand and minimizing the number of unbalanced stations is addressed and solved using a genetic algorithm. The performance of our prediction model and optimization strategy is comprehensively evaluated on real world bike sharing system data generated by NYC CitiBike System and the experimental results demonstrate the effectiveness and efficiency of our proposed method.

4.2 Problem Formulation

In this section, we first introduce some preliminaries used throughout this paper, and then formally define the problem of bike station network optimization.

4.2.1 Preliminaries

Station Network

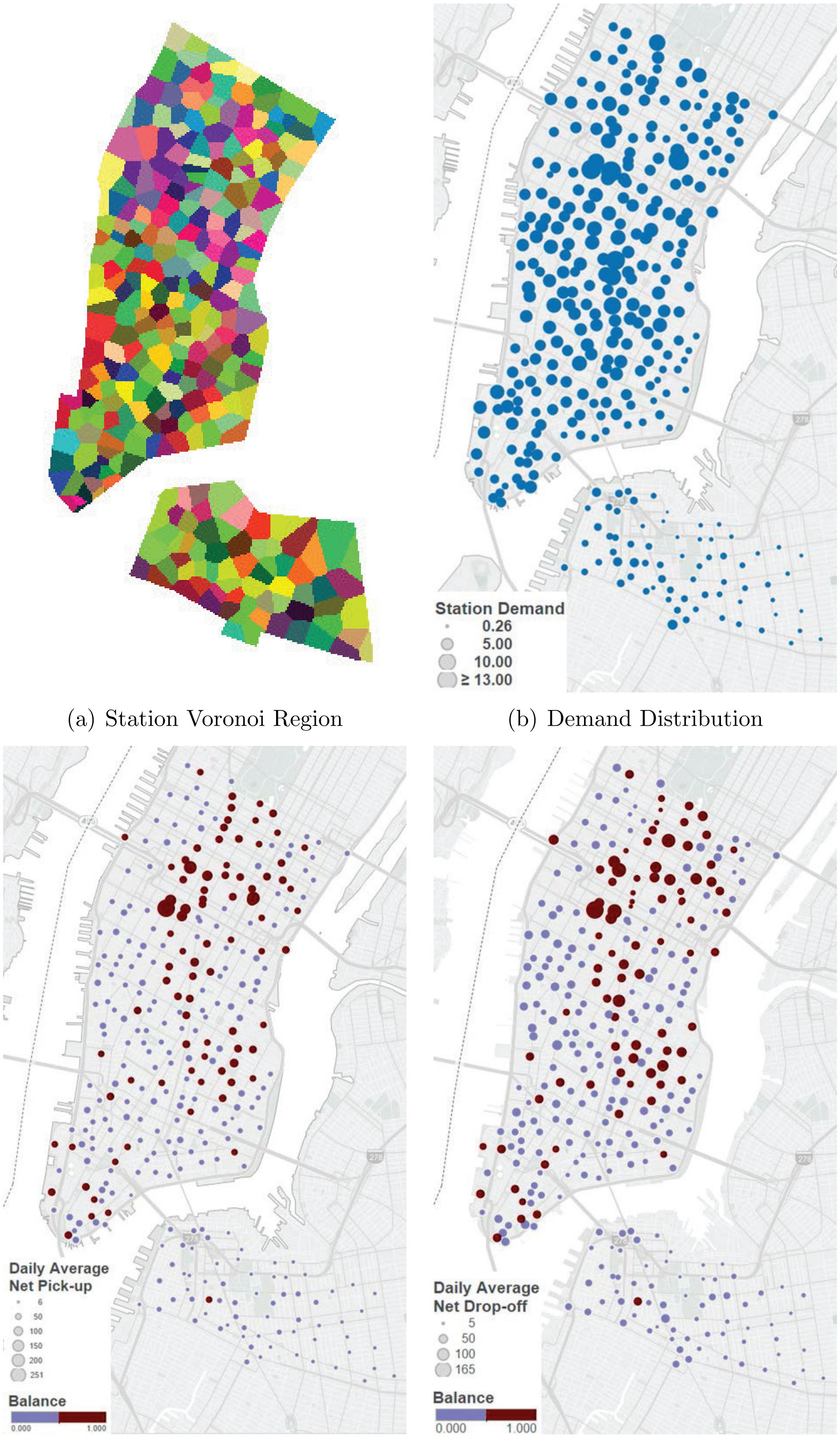
The bike station network is represented by a direct graph $G = (S, E)$. With each station $s \in S$ as a node, the edges in E are directed connections of bike stations $e_{ij} = (s_i, s_j) \in E$. Each node and edge have several attributes. For example, $e_{ij}.f$ represents the commuting frequency of pick-up at station s_i and a drop-off at station s_j.

Since need-based customers will choose the station closest to their current locations or final destinations, we partition the bike station in service area using a Voronoi-based gridding method (Aurenhammer, 1991), from which the map is partitioned into regions based on walking distance to bike stations. Each grid is centered by one bike station and the points within one region is closest to its center. As a result, pick-up/drop-off points for taxi trips and POIs are mapped to the nearest bike station.

Definition 1 (Voronoi Region) *Let X be a space coordinate endowed with a walking distance wd extracted from Google Maps Distance Matrix API. The Voronoi region R_{s_i} associated with station s_i is the set of all points in X whose distance to s_i is no greater than their distance to other stations:*

$R_{s_i} = \{x \in X | wd(x, s_i) \leq wd(x, s_j), \forall j \neq i\}$.

The NYC CitiBike in service area (Manhattan island below 61st street and western Brooklyn) is partitioned into Voronoi Regions centered by each CitiBike Station (see Figure 4.1(a)).



(a) Station Voronoi Region

(b) Demand Distribution

(c) Net-pick-up Distribution

(d) Net-drop-off Distribution

Figure 4.1. NYC CitiBike stations in service area Voronoi Region partition, bike demand distribution and balance distribution.

Station Demand

The station demand is defined as the average pick-up frequency/hour when this station is available. Station availability means the station is in service and there are bikes available for pick-up. Station pick-up unavailability is usually due to maintaining and empty dock. We do not consider the station demand during its unavailable period.

Definition 2 (Station Demand) *Let $s_i.f(T)$ and $s_i.a(T)$ represent the daily pick-up frequency and station in service time duration (hour) in day T. The station demand (SD_i) is defined as: $SD_i = \frac{1}{T} \sum_T \frac{s_i.f(T)}{s_i.a(T)}$.*

The station demand distribution of stations of NYC CitiBike sharing system is presented in Figure 4.1(b) as an example. In Figure 4.1(b), each circle represents a current in service bike station in NYC with its size representing its bike demand defined by Defintion 2.

Station Balance

Due to unbalanced bike demand distribution, bikes from full stations are dispatched by truck to empty stations, which greatly increases the operation cost of bike sharing systems and affects customers' conveniences. We investigate the station imbalance problem by first introducing the concept of station net pick-up/drop-off frequency from the daily transaction records. Let $\{s_i.pd(t_j)|j = 0, 1...\}$ represents the pick-up/drop-off events of station s_i at time t_j, where $s_i.pd(t_j) = 1$ for pick-up record and $s_i.pd(t_j) = -1$ for drop-off record. The net pick-up (net drop-off) $s_i.np$ ($s_i.nd$) is defined as the contiguous subarray of series $\{s_i.pd(t)\}$ whose values have the largest positive sum (smallest negative sum).

In our study, if the average net pick-up or net drop-off of station exceeds a threshold γ (decided by the tolerance of a station vacancy rate), we discriminate this station as unbalanced. For the situation of NYC CitiBike system, γ equals to the average dock numbers of the CitiBike stations.

Definition 3 (Station Balance) *Let* $(s_i.np(T_j), s_i.nd(T_j))$, $j = 1, ..., n$ *represents the net-pick/drop frequency of station* s_i *from day* T_1 *to* T_n*, the station balance is identified as a binary variable discriminated by a delta function* $\delta(x) = 1$ *if* x *is* *TRUE and* 0 *otherwise:*

$SB_i = \delta(\frac{1}{n}\sum_{j=1}^{n} s_i.np(T_j) \geq \gamma \textit{ or } \frac{1}{n}\sum_{j=1}^{n} s_i.nd(T_j) \geq \gamma)$

For the example of NYC CitiBike system, the station net pick-up/drop-offs and their balance patterns are presented in Figure 4.1(c) and Figure 4.1(d) with the unbalanced station highlighted in red.

4.2.2 Problem Formulation

The bike station network optimization problem for bike sharing systems can be separated into two stages: station demand and balance prediction; station network optimization.

Station Demand and Balance Prediction

Given a set of bike station locations and their surrounding features ($\mathbf{F}$), the problem of station demand and balance prediction is to predict the station demand defined in Definition 2 and to identify if the station is unbalanced according to Definition 4. In our study, we feed multi-factor features extracted from human mobilities, POIs and station network structures into prediction models based on neural network $NN_{SD}(s_i;\mathbf{F})$ for station demand prediction and neural network $NN_{SB}(s_i;\mathbf{F})$ for station balance prediction.

Station Network Optimization

Given the well trained neural network prediction models NN_{SD} and NN_{SB} from stage 1 and a set of bike station location candidates SC of size $|SC| = m$, the problem of station network optimization is to find an optimal subset OC of the location candidates SC such that the total demands from all chosen stations are maximized while the number of unbalanced stations are minimized. Formally, our objective function for station network optimization is defined as follows:

$$\max \mathcal{F}(\mathbf{y}) = \sum_{i=1}^{m} y_i(\frac{1}{n}NN_{SD}(s_i) - \lambda NN_{SB}(s_i)) \tag{4.1}$$

$$s.t. \quad \sum_{i=1}^{k} y_i = n_1 \tag{4.2}$$

$$\sum_{i=k+1}^{m} y_i = n_2 \tag{4.3}$$

$$\|s_i - s_j\| \geq y_i y_j d \quad \forall i \neq j \tag{4.4}$$

$$y_i \in \{0,1\} \quad j = 1,2,...,m \tag{4.5}$$

where $\mathbf{y} = \{y_1, y_2, ..., y_m\}$ is a binary variable vector. y_i=1 indicating location candidate s_i is chosen to be an optimal station site locationn otherwise $y_i = 0$. "$\|a - b\|$" is spherical earth (ignoring ellipsoidal effects) distance calculation according to the coordinates of two points. λ is a penalty parameter representing the additional cost and demand losing for unbalanced stations. Constrain (4.2) and Constrain (4.3) specify the limits of total number of stations in different areas respectively and the total number of stations n is pre-determined. Constrain (4.4) specifies the minimum distance between any optimal stations. Different from other optimization problems which treat station candidates independently, the station demand and balance are non-functionally decided by the chosen stations indicated by indicator vector $\mathbf{y}$. For the same selected candidate s_i, a different network will have different Voronoi Regions and different network structures which will affect the station demand and balance pattern for station s_i.

4.3 Feature Extraction

In this section, we introduce 10 fine-grained features extracted from station network, bicycle trajectories, taxi trajectories and POIs for station demand balance prediction.

4.3.1 Transportation Related Features

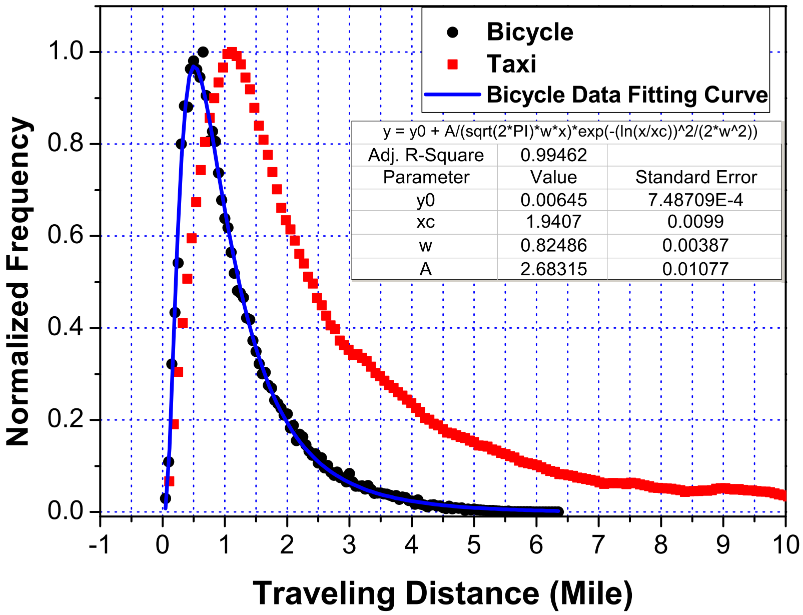
Public bicycles are widely used for short-term distance traveling and transportation missing link connection. It is very common that people will take bikes to nearby locations with more convenient accesses to other long-distance transportation like subways, taxis, etc. In our study, we extract the walking distance from each bike station to its nearest parking lot, the walking distance to the nearest subway station, the taxi pick-up densities and the number of faster bicycle routes as our transportation related features. Taxi pick-up density mapped to station s_i is the number of taxi pick-up in Voronoi Regin R_{s_i} divided by the region size: $s_i.TP = \sum_{k_t} \delta(TP_{k_t} \in R_{s_i})/|R_{s_i}|$. Because of traffic jams and vehicle detours, bicycling is faster than vehicles in some areas. For the same origins and destinations, people are more willing to take bikes if it is faster, cheaper and more convenient than vehicles. By tracking bicycles and taxies as speed sensors, we are able to define the feature of number of faster bicycle routes as follows: Let $e_{ij}.vt, e_{ij}.vb$ represents the average transportation time of taxis and bicycles from station s_i to station s_j. The feature number of faster bicycle route is defined as the number of edges taking a bicycle is faster than a taxi: $s_i.FR = \sum_{j \neq i} \delta(e_{ij}.vb - e_{ij}.vt > 0)$

4.3.2 POIs Features

POIs provide us various information about the city from different aspects. The density of POIs is an indicator of human crowd intensity. A high population density means a high probability of bicycle demand. On the other hand, people tend to take bicycles to go to/from their POIs. In terms of station balance, the stations near schools and restaurants are more likely to have a large net pick-up/drop-off during after-school time period and dining time. In this study, we use the density of 4 major categories of POIs within the Voronoi region surrounding each bicycle station, which are entertainment, restaurant, shopping center and education.

4.3.3 Station Network Profile

Station Scale. The station scale is represented by the total number of docks. Although the pick-up frequency is restricted by the station scale since a small station is more likely to be empty, the station demand from our definition is not restricted by this situation because the empty time period is not counted. In addition, because of the bike sharing re-balance system, the stations with size smaller than the threshold



y = y0 + A/(sqrt(2*PI)*w*x)*exp(-(ln(x/xc))^2/(2*w^2))

Adj. R-Square	0.99462	
Parameter	Value	Standard Error
y0	0.00645	7.48709E-4
xc	1.9407	0.0099
w	0.82486	0.00387
A	2.68315	0.01077

Figure 4.2. Distance preference comparisons between bicycles and taxis

γ defined in Definition 3 can still have a net pick-up/drop-off larger than γ and our definition of station balance is not restricted.

Nearby Station Score. Bicycle users will have a different traveling distance preference compared to vehicle users. From the historical traveling distance records of bicycles and taxis, we calculate the statistic frequencies of trips in different trip distance intervals and represent the frequencies as distance preference. The difference of the transportation distance preference between taxis and bicycles are presented in Figure 4.2. From Figure 4.2, we can see that people usually take bicycles for 0.5-1.5 mile distance transportation while most people take taxis for long distance destinations where the bicycles can hardly reach. Mathematically, the normalized pick-up frequency versus station distance forms a log-normal distribution (see the blue fitting line in Figure 4.2). Therefore, given the locations of two stations $s_i.c$, $s_j.c$ associated with their distance $x \equiv \|s_i.c - s_j.c\|$, we can estimate the users' preference of taking bicycles from s_i to s_j, which is defined by a single nearby station score ($SNSS_{ij} = y_0 + \frac{A}{\sqrt{2\pi}wx} exp(-\frac{(ln(x/x_c))^2}{2w^2})$). y_0, A, w, x_c are fitting parameters (see fitting results in inserted table of Figure 4.2). The feature of nearby station score is then defined as $NSS_i = \sum_{j \neq i}^{n} SNSS_{ij}$. From the definition of NSS, we can see that a station should not be located too close or too far away from other stations and the station demand should be positively correlated to the NSS.

4.4 Methodology

4.4.1 Prediction Model

We propose an artificial neural network (ANN) to predict the station demand and station balance based on the features extracted. The comparative advantage of ANN over most conventional prediction models is that it can implicity detect complex nonlinear relationships between the features from different domains and the targets without any prior assumptions about the underlying data generating process (Benediktsson, Swain, & Ersoy, 1990). The details of the specification and estimation of our M-layer ANN model is summarized below.

Layer Input. The net input to unit i in layer $k+1$ is the linear combinations of the outputs α^k in layer k. The network input α^0 is the feature vector normalized within [0,1] ranges by mapping $x = \frac{x - x_{min}}{x_{max} - x_{min}}$.

Layer Output. The output of unit i in layer $k+1$ is mapped from l^{k+1} using a sigmoid activation function $a^{k+1}(i) = \frac{1}{1+e^{-l^{k+1}}}$. The output layer is a linear layer for regression problem of station demand prediction and the final output a^M is t_{sd} (continue variable). For station balance prediction, a threshold output layer is trained and the final output a^M is binary variable t_{sb}.

Training Algorithm. Our training task is to learn the associations between the inputs and outputs of our training set which aims at minimizing the prediction error: $V = \frac{1}{2}\sum_{i=1}^{nt}(t_i - a_i^M)^2$. The Levenberg-Marquardt algorithm (Hagan, Demuth, Beale, et al., 1996) is applied for parameter training in our study. Moreover, a testing set is used for monitoring validation error and overfitting control without affecting training parameters during the training process.

4.4.2 Optimization Model

The station network optimization problem is to find a binary indicator vector $\mathbf{y}$ that maximizes our objective function (4.1). We first simulate $k = 1702$ and $m - k = 634$ locations as candidates in Manhattan and Brooklyn areas. Among which, we select $n_1 = 252$ optimal stations from Manhattan and $n_2 = 68$ optimal stations from Brooklyn. The candidates are simulated with equally distanced interval which cover the NYC CitiBike in service area and the docks number of each candidate is simulated to be 35 (the average number of docks of current bike system).

A genetic algorithm (GA) can be understood as a probabilistic search algorithm which is applicable to our combinational optimization problem (Reeves, 1993). In our case, each possible solution (an optimal station network) represented by our indicator vector $\mathbf{y}$ is identified by a chromosome as an individual with each element y_i representing one piece of gene. The process for solving the bike station network optimization problem starts by randomly initializing 1000 individuals as the first generation, which are transformed to the next generation through the designed tournament selection (Miller & Goldberg, 1995), recombination and mutation (Gen & Cheng, 2000). The termination criteria is setup by identifying if best objective is varying within 0.2% for 5 continuous generations.

In the tournament selection process of our study, 3 individuals are selected randomly from the large population and the selected individuals compete against each other. The individual with the highest value of objective function among the three is selected as one of the next generation population. This procedure is repeated 100 times and 100 individuals are selected for genetic operation of recombination and mutation to generate next generation. In recombination process, a multiple points crossover specified by a binary vector $S = (s_1, s_2, ..., s_m)$ is applied to determine the genes inherited from the two parents. In general, the crossover point marker S can be arbitrarily decided. However, we limit the structure of S to guarantee the constrain (4.2) and (4.3) in our optimization problem. Mutation is applied to explore newly possible offsprings for diversified generation. Two pieces of gene of offsprings from crossover are randomly selected to have 2 genes mutated.

4.5 Experiment

To validate the efficiency and effectiveness of our proposed method, extensive experiments are performed on real world NYC CitiBike trajectory data of 320 stations in Manhattan and Brooklyn area (see Figure 4.1(b)). The stations are randomly split into 80% (256) for training and 20% (64) for validation. Their demand and balance information are extracted from the CitiBike system historical data as our ground truth.

4.5.1 Data Description

Citibike Transactions. Citibike transactions are generated by NYC Bike Sharing System which is public available from Citibike official website. 11.3 million transactions are extracted from July 2013 to November 2014 with winter session from December 2013 to March 2014 excluded because the demand for bicycle during the winter is very low. This data set contains the following information: station id, bicycle pick-up station, bicycle pick-up time, bicycle drop-off station and bicycle drop-off time.

Taxi GPS Transactions. Taxi GPS transaction dataset is generated by taxis in New York City in August 2013 which is public available. 11.3 million taxi transactions are collected with each record containing the information of trip distance, taxi pick-up coordinate, taxi pick-up time, taxi drop-off coordinate and taxi drop-off time.

4.5.2 Feature Analysis

Correlation Analysis

We first perform a correlation analysis investigating the correlation relationship between our targets (station demand and station balance) and the features extracted from real world data (see Figure 4.3). The Pearson correlation coefficient is applied for station demand and features. For the correlation of station balance and features, we use Point-Biserial correlation. From Figure 4.3 we can see that all features are correlated to the targets we investigate, compared to a simulated random noise feature (RN). Moreover, the features of distance to subway entrance and parking lot are negative correlated, which indicate the bike station is treated as an transportation

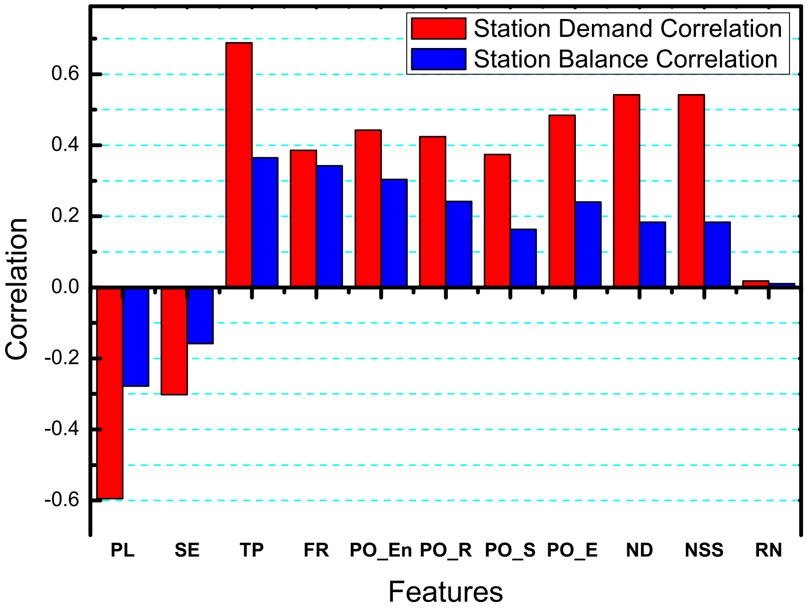


Figure 4.3. Correlation of features and station demand & balance patterns

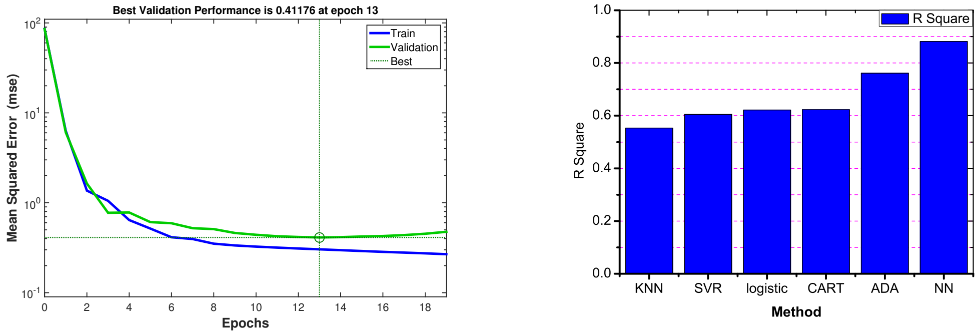
missing link connection.

4.5.3 Station Demand Prediction

Evaluation Metrics. To show the effectiveness of our proposed method for station demand prediction, we use the coefficient of determination for the prediction error measurement.

Training Progress. Figure 4.4(a) shows how fast the ANN converges using Levenberg-Marquardt algorithm. Although the training error continues decreasing, the optimized ANN is chosen at epoch 13 of minimum validation error.

Baseline Algorithm. We evaluate the effectiveness of our model for station demand prediction with a set of baselines, including (1)K-Nearest Neighbor; (2)Logistic Regression (3)SVR with RBF kernal; (4)Decision Tree; and (5) Adaboost Decision Tree Regression. All baseline algorithms are trained on the same training data set and their performance are compared using the same validation set as our Regression



(a) Training progress of ANN

(b) Performance comparison

Figure 4.4. Station demand prediction training progress and performance

Neural Network. All the baselines are implemented by a python machine learning library named Scikit-Learn (Pedregosa et al., 2011).

Overall Performance. The overall performance comparison of different methods is summarized in Figure 4.4(b). Our proposed Neural Network achieves an R^2 of 0.88168, which obviously outperforms the baseline algorithms with a significant margin. Among the 5 baseline algorithms, only AdaBoosted decision tree can achieve a relatively high R^2 of 0.76152. The algorithms of KNN (0.55322) logistic regression (0.62134), Suport Vector Regressor (0.60479) and CART (0.62261) are not able to predict station demand based on the features extracted.

4.5.4 Station Balance Prediction

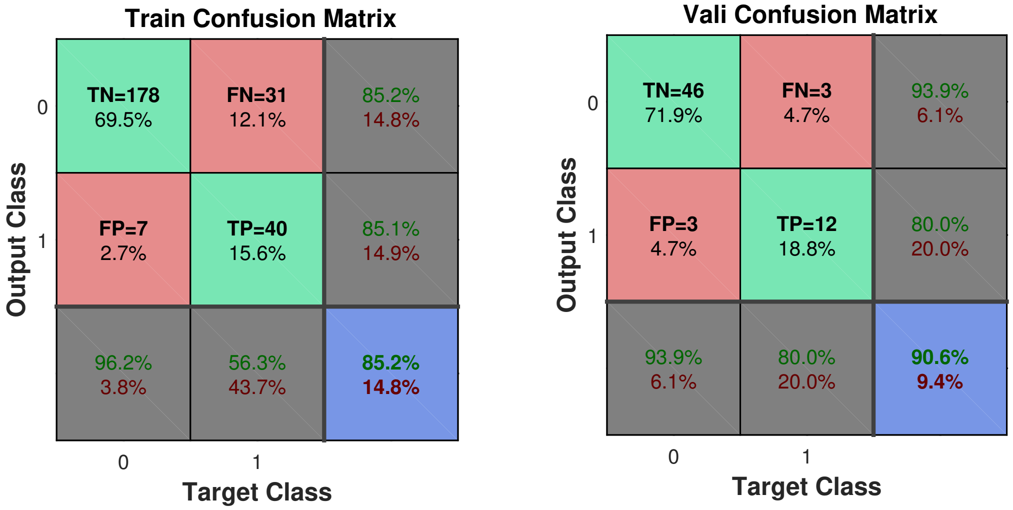
Evaluation Metrics. The classification performance of the optimized artificial neural networks for station balance prediction is evaluated using evaluation metrics including overall accuracy, precision, recall and F-measure.

Baseline Algorithm. We evaluate the effectiveness of our model for station demand prediction with a set of baselines: (1)K-Nearest Neighbor Classifier (KNN); (2)Support vector classifier (SVC) with linear kernal; (3)Gaussian Naive Bayes (GNB) classifier; (4)classification and regression tree (CART) and (5)Adaboost Decision Tree Classifier.

Overall Performance. The training and validation performance of artificial neural network based station balance prediction is presented by two confusion matrixes in Figure 4.5. Our proposed prediction model can achieve an accuracy of 85.2% for the 256 stations (185 balanced and 71 unbalanced stations) in training set and the validation accuracy reaches 90.6% for the rest 64 stations (49 balanced and 15 unbalanced stations). The overall performance comparison of different methods is summarized in Figure 4.6. As can be seen from Figure 4.6(a), our proposed method achieves the highest prediction accuracy compared to the 5 most commonly used classification algorithms. The overall validation accuracy of AdaBoost is above 84% and the Gaussian Naive Bayes has the lowest accuracy of 76.6%. Moreover, from Figure 4.6(b), 4.6(c) and 4.6(d), our method outperforms other baseline algorithms in terms of precision, recall and F-measure.

4.5.5 Station Network Optimization

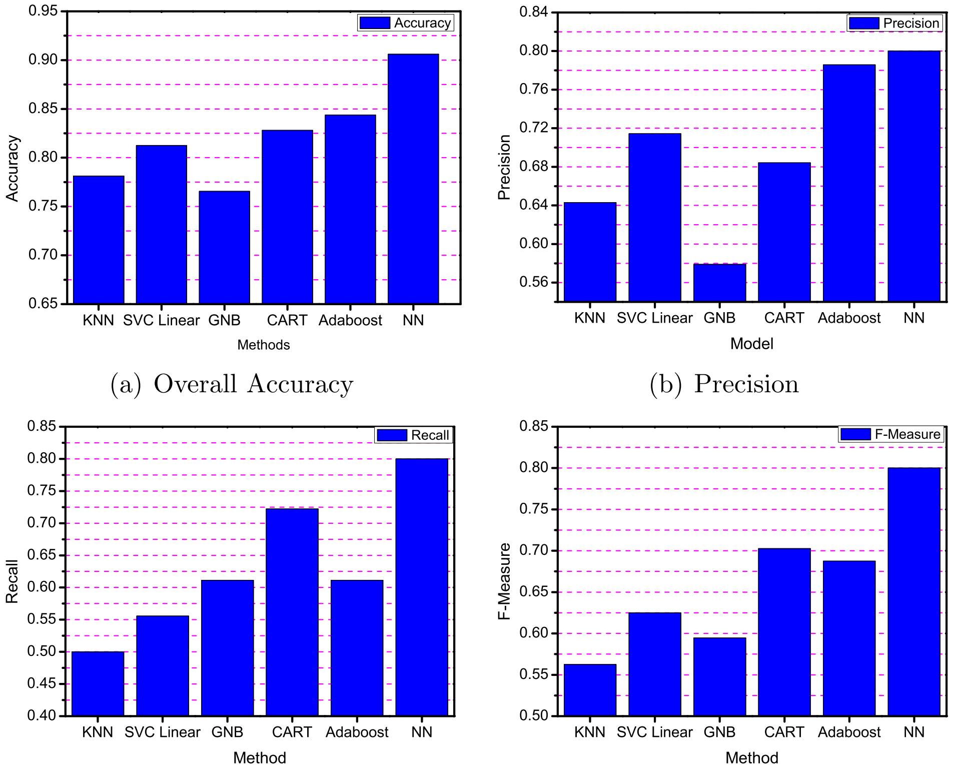
Based on our prediction models, a bike sharing network optimization is conducted to find 252 optimal stations from 1720 station candidates in Manhattan area and 68 optimal stations from 967 station candidates in Brooklyn. Figure 7(a) shows the progress of searching best station network. It can be seen, the optimization converges at 109th generation with the best objective of 3.42323, significantly higher than the current station network that has the same number of stations but obtains a much



(a) Training Confusion Matrix

(b) Validation Confusion Matrix

Figure 4.5. Confusion Matrix of ANN training and validation outputs



(a) Overall Accuracy

(b) Precision

(c) Recall

(d) F-Measure

Figure 4.6. Performance comparison of models for staion balance prediction

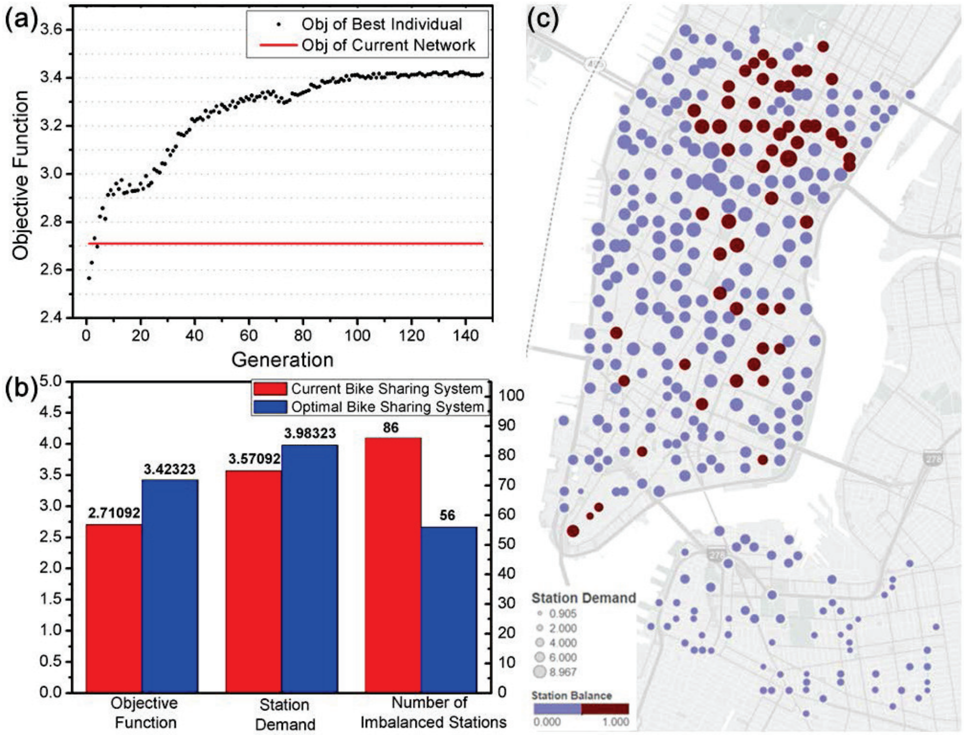


Figure 4.7. Distribution and statistics of optimum bike stations

lower objective of 2.71. The optimum stations achieve a high objective from two respects: the average station demand is 3.98323 compared to current stations with an average demand of 3.57092; the number of unbalanced stations decreases from 86 to 56 (see Figure 7(b)).The distribution of optimum stations is presented in Figure 7(c). The potential station demands are represented by the circle sizes and the red circles indicate unbalanced stations.

4.6 Conclusion

In this paper, we developed a comprehensive bike station network optimization approach by selecting bike station locations with high demand and balanced pick-ups/drop-offs . To the best of our knowledge, this paper is the first attempt to integrate multiple factors from human mobilities, surrounding POIs and station network structures for station demand prediction and balance evaluation in bike sharing systems. Specifically, artificial neural network based prediction models was developed to build the complex nonlinear relationships between the features extracted from different factors and the patterns of station demand and balance. Evaluated by bike sharing system data generated by NYC CitiBike System, our proposed model manifested the best prediction performance among other state of the art algorithms. Moreover, an genetic algorithm based optimization strategy aiming at maximizing station network demand as well as minimizing number of unbalanced stations was conducted by selecting optimal station locations from a large set of station locations.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

In this dissertation, I integrate data mining techniques and optimization algorithms for the design and operaions management in bike sharing systems, by effectively modeling and efficientyly computing with various bike system data, human mobility data, meteorology data, and other POI data.

First, I developed a data-driven bike station inventory reblancing model by exploiting the station-level bike pick-up and drop-off demand. I first develop a metorology similarity K-nearest Neighbor regressor and a nonlinear autoregressive network with exogenous meteorology factors (NARX) to predict bike pick-up demand, and a pick-drop bike transition (PDBT) predictor for transition patterns discovery and bike drop-off demand prediction. Then, a Mixed Integer Linear Programming (MILP) model is formulated to redistribute bikes using rebalancing vehicles. To address the challenge of computational efficiency, we propose a data-driven hierarchical optimization methodology to decompose the multi-vehicle routing problem into smaller and localized single-vehicle routing problems. Further, we propose two advanced rebalancing strategies: partial target satisfying strategy and multi-vehicle visiting strategy to deal with outlier stations while ensuring the feasibility of the route optimization solution.

Second, I developed a hierarchical bike demand prediction models for expansion area station level bike demand prediction. Specifically, we first partitioned the station in service area into different functional zones based on our Bi-Clustering algorithm. Then based on the functional zones, we implemented Random Forest Regressor to estimate the functional zone bike transitions by integrating the bike trip distance preference, zone-to-zone preference, and zone characteristics. The station level bike demand was predicted by distributing the zone level check-ins and check-outs to each station with the consideration of their Voronoi region POI structures.

Third, I developed a comprehensive bike station network optimization approach by selecting bike station locations with high demand and balanced pick-ups/drop-offs . This model attemptted to integrate multiple factors from human mobilities, surrounding POIs and station network structures for station demand prediction and balance evaluation in bike sharing systems. Specifically, artificial neural network based prediction models was developed to build the complex nonlinear relationships between the features extracted from different factors and the patterns of station demand and balance. Evaluated by bike sharing system data generated by NYC CitiBike System, our proposed model manifested the best prediction performance among other state of the art algorithms. Moreover, a genetic algorithm based optimization strategy aiming at maximizing station network demand as well as minimizing number of unbalanced stations was conducted by selecting optimal station locations from a large set of station locations.

BIBLIOGRAPHY

Aguayo, M. M., Sarin, S. C., & Sherali, H. D. (2018). Solving the single and multiple asymmetric traveling salesmen problems by generating subtour elimination constraints from integer solutions. *IISE Transactions*, *50*(1), 45-53.

Alvarez-Valdes, R., Belenguer, J. M., Benavent, E., Bermudez, J. D., Muoz, F., Vercher, E., & Verdejo, F. (2016). Optimizing the level of service quality of a bike-sharing system. *Omega*, *62*, 163 - 175.

Alvarez-Valdes, R., Belenguer, J. M., Benavent, E., Bermudez, J. D., Muoz, F., Vercher, E., & Verdejo, F. (2016). Optimizing the level of service quality of a bike-sharing system. *Omega*, *62*, 163 - 175.

Applegate, D. L., Bixby, R. E., Chvatal, V., & Cook, W. J. (2011). *The traveling salesman problem: a computational study.* Princeton university press.

Aurenhammer, F. (1991). Voronoi diagrams, a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, *23*(3), 345–405.

Basu, S., Bilenko, M., & Mooney, R. J. (2004). A probabilistic framework for semi-supervised clustering. In *Sigkdd 2004* (pp. 59–68).

Benediktsson, J. A., Swain, P. H., & Ersoy, O. K. (1990). Neural network approaches versus statistical methods in classification of multisource remote sensing data.

Chemla, D., Meunier, F., & Calvo, R. W. (2013). Bike sharing systems: Solving the static rebalancing problem. *Discrete Optimization*, *10*(2), 120 - 146.

Chen, L., Zhang, D., Wang, L., Yang, D., Ma, X., Li, S., … Jakubowicz, J. (2016). Dynamic cluster-based over-demand prediction in bike sharing systems. In *Proceedings of the 2016 acm international joint conference on pervasive and ubiquitous computing* (pp. 841–852). New York, NY, USA: ACM.

Contardo, C., Morency, C., & Rousseau, L.-M. (2012). *Balancing a dynamic public bike-sharing system* (Vol. 4). CIRRELT.

Corcoran, J., Li, T., Rohde, D., Charles-Edwards, E., & Mateo-Babiano, D. (2014). Spatio-temporal patterns of a public bicycle sharing program: the effect of weather and calendar events. *Journal of Transport Geography*, *41*, 292 - 305.

Dell'Amico, M., Hadjicostantinou, E., Iori, M., & Novellani, S. (2014). The bike sharing rebalancing problem: Mathematical formulations and benchmark instances. *Omega*, *45*(Supplement C), 7 - 19.

dell'Olio, L., Ibeas, A., & Moura, J. L. (2011). Implementing bike-sharing systems. *Proceedings of the Institution of Civil Engineers - Municipal Engineer*, *164*(2), 89-101.

DeMaio, P. (2009). Bike-sharing: History, impacts, models of provision, and future. *Journal of public transportation*, *12*(4), 3.

DeMaio, P., & Meddin, R. (2018). *The bike-sharing world map.*

Erdoğan, G., Laporte, G., & Calvo, R. W. (2013). *The one commodity pickup and delivery traveling salesman problem with demand intervals.* CIRRELT, Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation.

Erdoan, G., Battarra, M., & Calvo, R. W. (2015). An exact algorithm for the static rebalancing problem arising in bicycle sharing systems. *European Journal of Operational Research*, *245*(3), 667 - 679.

Faghih-Imani, A., Hampshire, R., Marla, L., & Eluru, N. (2017). An empirical analysis of bike sharing usage and rebalancing: Evidence from barcelona and seville. *Transportation Research Part A: Policy and Practice*, *97*, 177 - 191.

Forma, I. A., Raviv, T., & Tzur, M. (2015). A 3-step math heuristic for the static repositioning problem in bike-sharing systems. *Transportation Research Part B: Methodological*, *71*, 230 - 247.

Freund, D., Henderson, S. G., & Shmoys, D. B. (2017). Minimizing multimodular functions andallocating capacity in bike-sharing systems. In F. Eisenbrand & J. Koenemann (Eds.), *Integer programming and combinatorial optimization* (pp. 186–198).

Froehlich, J., Neumann, J., & Oliver, N. (2009). Sensing and predicting the pulse of the city through shared bicycling. In *Proceedings of the 21st international jont conference on artifical intelligence* (pp. 1420–1426).

Froehlich, J., Neumann, J., Oliver, N., et al. (2009). Sensing and predicting the pulse of the city through shared bicycling. In *Ijcai* (Vol. 9, pp. 1420–1426).

García-Palomares, J. C., Gutiérrez, J., & Latorre, M. (2012). Optimizing the location of stations in bike-sharing programs: a gis approach. *Applied Geography*, *35*(1), 235–246.

Garca-Palomares, J. C., Gutirrez, J., & Latorre, M. (2012). Optimizing the location of stations in bike-sharing programs: A {GIS} approach. *Applied Geography*, *35*(1C2), 235 - 246.

Garca-Palomares, J. C., Gutirrez, J., & Latorre, M. (2012). Optimizing the location of stations in bike-sharing programs: A gis approach. *Applied Geography*, *35*(1), 235 - 246.

Gebhart, K., & Noland, R. B. (2014, Nov 01). The impact of weather conditions on bikeshare trips in washington, dc. *Transportation*, *41*(6), 1205–1225.

Gen, M., & Cheng, R. (2000). *Genetic algorithms and engineering optimization* (Vol. 7). John Wiley & Sons.

Grushka-Cockayne, Y., Jose, V. R. R., & LichtendahlJr., K. C. (2017). Ensembles of overfit and overconfident forecasts. *Management Science*, *63*(4), 1110-1130.

Gurobi Optimization, I. (2016). *Gurobi optimizer reference manual.* Retrieved from `http://www.gurobi.com`

Hagan, M. T., Demuth, H. B., Beale, M. H., et al. (1996). *Neural network design.* Pws Pub. Boston.

Kaltenbrunner, A., Meza, R., Grivolla, J., Codina, J., & Banchs, R. (2010a). Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system. *Pervasive and Mobile Computing*, *6*(4), 455 - 466. (Human Behavior in Ubiquitous Environments: Modeling of Human Mobility Patterns)

Kaltenbrunner, A., Meza, R., Grivolla, J., Codina, J., & Banchs, R. (2010b). Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system. *Pervasive and Mobile Computing*, *6*(4), 455–466.

Kaspi, M., Raviv, T., & Tzur, M. (2014). Parking reservation policies in one-way vehicle sharing systems. *Transportation Research Part B: Methodological*, *62*, 35 - 50.

Kloimüllner, C., Papazek, P., Hu, B., & Raidl, G. R. (2014). Balancing bicycle sharing systems: An approach for the dynamic case. In *Evolutionary computation in combinatorial optimisation* (pp. 73–84). Springer.

Kloimüllner, C., Papazek, P., Hu, B., & Raidl, G. R. (2015). A cluster-first route-second approach for balancing bicycle sharing systems. In R. Moreno-Díaz, F. Pichler, & A. Quesada-Arencibia (Eds.), *Computer aided systems theory – eurocast 2015* (pp. 439–446).

Laporte, G., Meunier, F., & Wolfler Calvo, R. (2015, Dec 01). Shared mobility systems. *4OR*, *13*(4), 341–360.

Li, Y., Zheng, Y., Zhang, H., & Chen, L. (2015a). Traffic prediction in a bike-sharing system. In *Proceedings of the 23rd sigspatial international conference on advances in geographic information systems* (pp. 33:1–33:10).

Li, Y., Zheng, Y., Zhang, H., & Chen, L. (2015b). Traffic prediction in a bike-sharing system. In *Proceedings of the 23rd sigspatial international conference on advances in geographic information systems* (pp. 33:1–33:10). New York, NY, USA: ACM.

Lin, J.-R., Yang, T.-H., & Chang, Y.-C. (2013). A hub location inventory model for bicycle sharing system design: Formulation and solution. *Computers and Industrial Engineering*, *65*(1), 77 - 86. (Intelligent Manufacturing Systems)

Liu, J., Li, Q., Qu, M., Chen, W., Yang, J., Xiong, H., … Fu, Y. (2015, Nov). Station site optimization in bike sharing systems. In *2015 ieee international conference on data mining* (p. 883-888).

Liu, J., Sun, L., Chen, W., & Xiong, H. (2016a). 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* (pp. 1005–1014).

Liu, J., Sun, L., Chen, W., & Xiong, H. (2016b). 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* (pp. 1005–1014). New York, NY, USA: ACM.

Liu, J., Sun, L., Li, Q., Ming, J., Liu, Y., & Xiong, H. (2017). Functional zone based hierarchical demand prediction for bike system expansion. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining* (pp. 957–966). New York, NY, USA: ACM.

Long, Y., & Shen, Z. (2015). Discovering functional zones using bus smart card data and points of interest in beijing. In *Geospatial analysis to support urban planning in beijing* (pp. 193–217). Springer.

Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, *17*(4), 395–416.

Martinez, L. M., Caetano, L., Eiró, T., & Cruz, F. (2012). An optimisation algorithm to establish the location of stations of a mixed fleet biking system: an application to the city of lisbon. *Procedia-Social and Behavioral Sciences*, *54*, 513–524.

Miller, B. L., & Goldberg, D. E. (1995). Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, *9*(3), 193–212.

Motoaki, Y., & Daziano, R. A. (2015). A hybrid-choice latent-class model for the analysis of the effects of weather on cycling demand. *Transportation Research Part A: Policy and Practice*, *75*, 217 - 230.

O'Brien, O., Cheshire, J., & Batty, M. (2014). Mining bicycle sharing data for generating insights into sustainable transport systems. *Journal of Transport Geography*, *34*, 262 - 273.

Olson, D. L., & Wu, D. (2017). Regression tree models. In *Predictive data mining models* (pp. 45–54). Singapore: Springer Singapore.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *J Machine Learning Research*, *12*, 2825–2830.

Pucher, J., Dill, J., & Handy, S. (2010). Infrastructure, programs, and policies to increase bicycling: an international review. *Preventive medicine*, *50*, S106–S125.

Pucher, J., Garrard, J., & Greaves, S. (2011). Cycling down under: a comparative analysis of bicycling trends and policies in sydney and melbourne. *Journal of Transport Geography*, *19*(2), 332–345.

Rainer-Harbach, M., Papazek, P., Hu, B., & Raidl, G. R. (2013). *Balancing bicycle sharing systems: A variable neighborhood search approach.* Springer.

Reeves, C. R. (1993). Modern heuristic techniques for combinatorial optimization. *Alfred Waller Ltd*.

Rodriguez, A., & Laio, A. (2014). Clustering by fast search and find of density peaks. *Science*, *344*(6191), 1492–1496.

Schuijbroek, J., Hampshire, R., & van Hoeve, W.-J. (2013). Inventory rebalancing and vehicle routing in bike sharing systems.

Schuijbroek, J., Hampshire, R., & van Hoeve, W.-J. (2017). Inventory rebalancing and vehicle routing in bike sharing systems. *European Journal of Operational Research*, *257*(3), 992 - 1004.

Shaheen, S. A., Guzman, S., & Zhang, H. (2010). Bikesharing in europe, the americas, and asia. *Transportation Research Record: Journal of the Transportation Research Board*, *2143*(1), 159–167.

Shekhar, S., Jiang, Z., Ali, R. Y., Eftelioglu, E., Tang, X., Gunturi, V. M. V., ... Zhou, X. (2015). Spatiotemporal data mining: A computational perspective. *International Journal of Geo-Information*, *4*(4).

Shu, J., Chou, M. C., Liu, Q., Teo, C.-P., & Wang, I.-L. (2013). Models for effective deployment and redistribution of bicycles within public bicycle-sharing systems. *Operations Research*, *61*(6), 1346-1359.

Singhvi, D., Singhvi, S., Frazier, P. I., Henderson, S. G., O'Mahony, E., Shmoys, D. B., & Woodard, D. B. (2015). Predicting bike usage for new york city's bike sharing system. In *Aaai workshop: Computational sustainability.*

Singla, A., Santoni, M., Bartók, G., Mukerji, P., Meenen, M., & Krause, A. (2015). Incentivizing users for balancing bike sharing systems. In *Aaai* (pp. 723–729).

Szeto, W., Ghosh, B., Basu, B., & OMahony, M. (2009). Multivariate traffic forecasting technique using cell transmission model and sarima model. *Journal of Transportation Engineering*, *135*(9), 658–667.

Vinh, N. X., Epps, J., & Bailey, J. (2009). Information theoretic measures for clusterings comparison: Is a correction for chance necessary. In *Proceedings of the 26th icml* (pp. 1073–1080).

Vogel, P., Greiser, T., & Mattfeld, D. C. (2011). Understanding bike-sharing systems using data mining: Exploring activity patterns. *Procedia-Social and Behavioral Sciences*, *20*, 514–523.

Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S., et al. (2001). Constrained k-means clustering with background knowledge. In *Icml* (Vol. 1, pp. 577–584).

Wang, W. (2016). Forecasting bike rental demand using new york citi bike data.

Waserhole, A., & Jost, V. (2016, Aug 01). Pricing in vehicle sharing systems: optimization in queuing networks with product forms. *EURO Journal on Transportation and Logistics*, *5*(3), 293–320.

Wolsey, L. A. (1998). *Integer programming.* Wiley, New York.

Xu, R., & Wunsch, D., II. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, *16*(3), 645–678.

Yianilos, P. N. (1993). Data structures and algorithms for nearest neighbor search in general metric spaces. In *Soda* (Vol. 93, pp. 311–321).

Yuan, N. J., Zheng, Y., Xie, X., Wang, Y., Zheng, K., & Xiong, H. (2015). Discovering urban functional zones using latent activity trajectories. *IEEE Transactions on Knowledge and Data Engineering*, *27*(3), 712–725.

Zeng, M., Yu, T., Wang, X., Su, V., Nguyen, L. T., & Mengshoel, O. J. (2016). Improving demand prediction in bike sharing system by learning global features. *Machine Learning for Large Scale Transportation Systems (LSTS)@ KDD-16*.

Zhou, X. (2015, 10). Understanding spatiotemporal patterns of biking behavior by analyzing massive bike sharing data in chicago. *PLOS ONE*, *10*(10), 1-20.