Optimizing Bike Sharing Systems: Dynamic Prediction and Rebalancing Using Machine Learning and Statistical Techniques

Thesis · April 2019

DOI: 10.13140/RG.2.2.26034.43202

1 author:

Mohammed Almannaa
King Saud University
38 PUBLICATIONS **324** CITATIONS

SEE PROFILE

Optimizing Bike Sharing Systems: Dynamic Prediction Using Machine Learning and Statistical Techniques and Rebalancing

Mohammed Hamad Almannaa

Dissertation submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy
in
Civil Engineering

Hesham A. Rakha, Chair
Mohammed M. Elhenawy
Feng Guo
Ralph Buehler
Montasir M. Abbas

April 22nd, 2019
Blacksburg, VA

Keywords: Dynamic Linear and Incremental Learning Models, Bike Sharing System, Bike Count Prediction, Machine Learning and Statistical algorithms and models, Portable Bike Stations

Optimizing Bike Sharing Systems: Dynamic Prediction Using Machine Learning and Statistical Techniques and Rebalancing

Mohammed Hamad Almannaa

ABSTRACT

The large increase in on-road vehicles over the years has resulted in cities facing challenges in providing high-quality transportation services. Traffic jams are a clear sign that cities are overwhelmed, and that current transportation networks and systems cannot accommodate the current demand without a change in policy, infrastructure, transportation modes, and commuter mode choice. In response to this problem, cities in a number of countries have started putting a threshold on the number of vehicles on the road by deploying a partial or complete ban on cars in the city center. For example, in Oslo, leaders have decided to completely ban privately-owned cars from its center by the end of 2019, making it the first European city to totally ban cars in the city center. Instead, public transit and cycling will be supported and encouraged in the banned-car zone, and hundreds of parking spaces in the city will be replaced by bike lanes.

As a government effort to support bicycling and offer alternative transportation modes, bike-sharing systems (BSSs) have been introduced in over 50 countries. BSSs aim to encourage people to travel via bike by distributing bicycles at stations located across an area of service. Residents and visitors can borrow a bike from any station and then return it to any station near their destination. Bicycles are considered an affordable, easy-to-use, and, healthy transportation mode, and BSSs show significant transportation, environmental, and health benefits.

As the use of BSSs have grown, imbalances in the system have become an issue and an obstacle for further growth. Imbalance occurs when bikers cannot drop off or pick-up a bike because the bike station is either full or empty. This problem has been investigated extensively by many researchers and policy makers, and several solutions have been proposed. There are three major ways to address the rebalancing issue: static, dynamic and incentivized. The incentivized approaches make use of the users in the balancing efforts, in which the operating company incentives them to change their destination in favor of keeping the system balanced. The other two approaches: static and dynamic, deal with the movement of bikes between stations either during or at the end of the day to overcome station imbalances. They both assume the location and number of bike stations are fixed and only the bikes can be moved. This is a realistic assumption given that current BSSs have only fixed stations. However, cities are dynamic and their geographical and economic growth affects the distribution of trips and thus constantly changing BSS user behavior. In addition, work-related bike trips cause certain stations to face a high-demand level during weekdays, while these same stations are at a low-demand level on weekends, and thus may be of little use. Moreover, fixed stations fail to accommodate big events such as football games, holidays, or sudden weather changes.

This dissertation proposes a new generation of BSSs in which we assume some of the bike stations can be portable. This approach takes advantage of both types of BSSs: dock-based and dock-less. Towards this goal, a BSS optimization framework was developed at both the tactical and operational level. Specifically, the framework consists of two levels: predicting bike counts at stations using fast, online, and incremental learning approaches and then balancing the system

using portable stations. The goal is to propose a framework to solve the dynamic bike sharing repositioning problem, aiming at minimizing the unmet demand, leading to increased user satisfaction and reducing repositioning/rebalancing operations.

This dissertation contributes to the field in five ways. First, a multi-objective supervised clustering algorithm was developed to identify the similarity of bike-usage with respect to time events. Second, a dynamic, easy-to-interpret, rapid approach to predict bike counts at stations in a BSS was developed. Third, a univariate inventory model using a Markov chain process that provides an optimal range of bike levels at stations was created. Fourth, an investigation of the advantages of portable bike stations, using an agent-based simulation approach as a proof-of-concept was developed. Fifth, mathematical and heuristic approaches were proposed to balance bike stations.

Optimizing Bike Sharing Systems: Dynamic Prediction Using Machine Learning and Statistical Techniques and Rebalancing

Mohammed Hamad Almannaa

GENERAL AUDIENCE ABSTRACT

Large urban areas are often associated with traffic congestion, high carbon mono/dioxide emissions (CO/CO2), fuel waste, and associated decreases in productivity. The estimated loss attributed to missed productivity and wasted fuel increased from $87.2 to $115 between 2007 and 2009. Driving in congested areas also results in long trip times. For instance, in 1993, drivers experienced trips that were 1.2 min/km longer in congested conditions.

As a result, commuters are encouraged to leave their cars at home and use public transportation modes instead. However, public transportation modes fails to deliver commuters to their exact destination. Users have to walk some distance, which is commonly called the “last mile”. Bike sharing systems (BSSs) have started to fill this gap, offering a flexible and convenient transportation mode for commuters, around the clock. This is in addition to individual financial savings, health benefits, and reduction in congestion and emissions. Resent reports have shown BSSs multiplying over 50 countries.

This notable expansion of BSSs also brings daily logistical challenges due to the imbalanced demand, causing some stations to run empty while others become full. Rebalancing the bike inventory in a BSS is crucial to ensure customer satisfaction and the whole system’s effectiveness. Most of the operating costs are also associated with rebalancing. The current rebalancing approaches assume stations are fixed and thus don’t take into account that the demand changes from weekday to weekend as well as from peak to non-peak hours, making some stations useless during specific days of the week and times of day. Furthermore, cities change continually with regard to demographics or structures and thus the distribution of trips also changes continually, leading to re-installation of stations to accommodate the dynamic change, which is both impractical and costly.

In this dissertation, we propose a new generation of BSS in which we assume some stations are portable, meaning they can move during the day. They can be either stand-alone or an extension of existing stations with the goal of accommodating the dynamic changes in the distribution of trips during the day. To implement our new BSSs, we developed a BSS optimization framework. This framework consists of two components: predicting the bike counts at stations using fast approaches and then balancing the system using portable stations. The goal is to propose a framework to solve the dynamic bike sharing repositioning problem, aiming at minimizing the unmet demand, leading to increased user satisfaction and reducing repositioning/rebalancing operations.

This dissertation contributes to the field in five ways. First, a novel algorithm was developed to identify the similarity of bike-usage with respect to time events. Second, easy-to-interpret and rapid approaches to predict bike counts at stations in a BSS were developed. Third, an inventory model using statistical techniques that provide an optimal range of bike levels at stations was created. Fourth, an investigation of the advantages of portable bike stations was developed. Fifth, mathematical approach was proposed to balance bike stations.

DEDICATION

To my parents (*Hamad* and *Haya*), my wife (*Asmaa*), and my two little daughters (*Maysaan* and *Kayaa*

ACKNOWLEDGEMENTS

All praise is due to Allah, we thank Him, seek His guidance and forgiveness. I'm extremely thankful to my family: my parents (Hamad and Haya), my wife (Asmaa), my brothers (Majed, Zyad, and Othman), and my lovely sisters (Eman and Huda) for their love, affection, and emotional support throughout my undergraduate and graduate career.

I would like to express my sincere gratitude to my advisory committee for their outstanding guidance, continuous support, never-ending patience, and friendly advice during my PhD journey: Dr. Hesham Rakha, Dr. Mohammed Elhenawy, Dr. Feng Guo, Dr. Ralph Buhler, and Dr. Montasir Abbas. I would like to give special thanks to my adviser, Dr. Hesham Rakha, for his continuous help, constructive guidance, and great encouragement. My sincere appreciation also goes to Dr. Mohammed Elhenawy for his mentoring during my PhD research. I have learned greatly from his knowledge in artificial intelligence and machine learning fields. My thanks also go to Dr. Feng Guo for his invaluable and immense knowledge in statistical modeling that I have benefited from during my journey. I must acknowledge the great input from Dr. Ralph Buehler in the planning side of this research.

My sincere thanks also go to my sponsor, King Saud University in Riyadh, Saudi Arabia, for funding me since I arrived in the United States of America in 2012 until writing this dissertation.

During the past years, I have been blessed with a supportive and cheerful group of colleagues in the Center for Sustainable Mobility at the Virginia Tech Transportation Institute; in particular: Dr. Mohammed Elhenawy, Dr. Ahmed Elbery, Ahmed Ghanem, Mohamed Aljamal, Dr. Mohamed Abdelmegeed, Dr. Huthaifa Ashqar, Karim Fadhloun, Dr. Osama Osman, Dr. Youssef Bichiou, Dr. Hossam Abdelghaffar, and many others. Your friendship made my PhD journey a pleasant experience.

Mohammed Almannaa
April, 22nd, 2019

{وَقُلِ اعْمَلُوا فَسَيَرَى اللَّهُ عَمَلَكُمْ وَرَسُولُهُ وَالْمُؤْمِنُونَ ۖ وَسَتُرَدُّونَ إِلَىٰ عَالِمِ الْغَيْبِ وَالشَّهَادَةِ فَيُنَبِّئُكُم بِمَا كُنتُمْ تَعْمَلُونَ}
سورة التوبة آية (105)

And say, "Do [as you will], for Allah will see your deeds, and [so, will] His Messenger and the believers. And you will be returned to the Knower of the unseen and the witnessed, and He will inform you of what you used to do." Quran, 9:105

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

1.1 Introduction

The large increase in on-road vehicles has resulted in cities facing challenges in providing high-quality transportation services. Traffic jams are a clear sign that cities are overwhelmed, and that current transportations networks and systems cannot accommodate the current demand without a change in policy, infrastructure, transportation modes, and commuters' choice of transportation mode. In response to this issue, cities in a number of countries have started putting a threshold on the number of vehicles on the road by deploying a partial or complete ban on cars in the city center. For example, in Oslo, leaders have decided to completely ban privately-owned cars from its center by the end of 2019, making it the first European city to totally ban cars in the city center (*1*). Instead, public transit and cycling will be supported and encouraged in the banned-car zone, and hundreds of parking spaces in the city will be replaced by bike lanes. As another example, in Dublin, Ireland, a proposal has been made to totally ban privately-owned cars from selected areas of the city center and push for public transit and bicycle use (*2*).

As an effort by governments to support bicycling and offer alternative transportation modes, bike-sharing systems (BSSs) have been introduced in over 50 countries (*1*). BSSs aim to encourage people to travel via bike by distributing bicycles from stations located across an area of service. Residents and visitors can borrow a bike from any station and then return it to any station near their destination. Bicycles are considered an affordable, easy-to-use, and, healthy transportation mode, and BSSs show significant transportation, environmental, and health benefits. In transportation, BSSs partially replace privately-owned car trips with bicycling, thereby mitigating traffic jams in the city. A survey conducted by McNeil at al. found that 80% or more of BSS users said they use BSSs for shopping/errands, social/recreational, trips to and from public transit, and commute trips (*3*), confirming that BSSs are becoming a reliable and convenient transportation mode for both recreational and non-recreational trips. In the environmental and health fields, the reduction in privately-owned car trips means less carbon energy consumption and carbon emissions. Qiu and He found that using BSSs in Beijing could save workers 8 minutes per day and that this saving could result in reducing fuel consumption by 225.05 thousand tons (*4*). This would contribute to an increase in Beijing's GDP of 1.2 billion Ren Min Bi (RMB), the official currency of China, and a reduction in health costs of 2,420.57 million RMB.

As the use of BSSs has grown, imbalance has become an issue and an obstacle to their further growth. Imbalance occurs when bikers cannot drop off or pick-up a bike because the bike station is either full or empty. This problem has been investigated extensively by many researchers and policy makers, and several solutions have been proposed (*5*-*9*). There are three major approaches to addressing the rebalancing issue: static, dynamic, and incentivized. The incentivized approaches employ BSS users in the balancing efforts by providing incentives for users to change their destination in favor of keeping the system balanced. Static approaches neglect demand during the rebalancing time, as rebalancing usually occurs when bike activities are at their lowest: at midnight or during the early morning hours. Dynamic approaches are more complicated, as they take into account the movement of bikes during the rebalancing efforts, and they can be done any time during the day. Regardless of the approach, the first step of rebalancing efforts is accurately predicting bike counts at any station in the BSS (Figure 1-1). This helps both bikers and operating agencies plan ahead and act accordingly. For instance, bikers could change their destination in advance if they knew that the station would be either empty or full by the time they arrived, which

would keep the BSS balanced without a need for relocating bikes. Operating agencies could use the predicted demand in the rebalancing approaches when repositioning bikes to prevent any station from running out, or having too many, bikes.

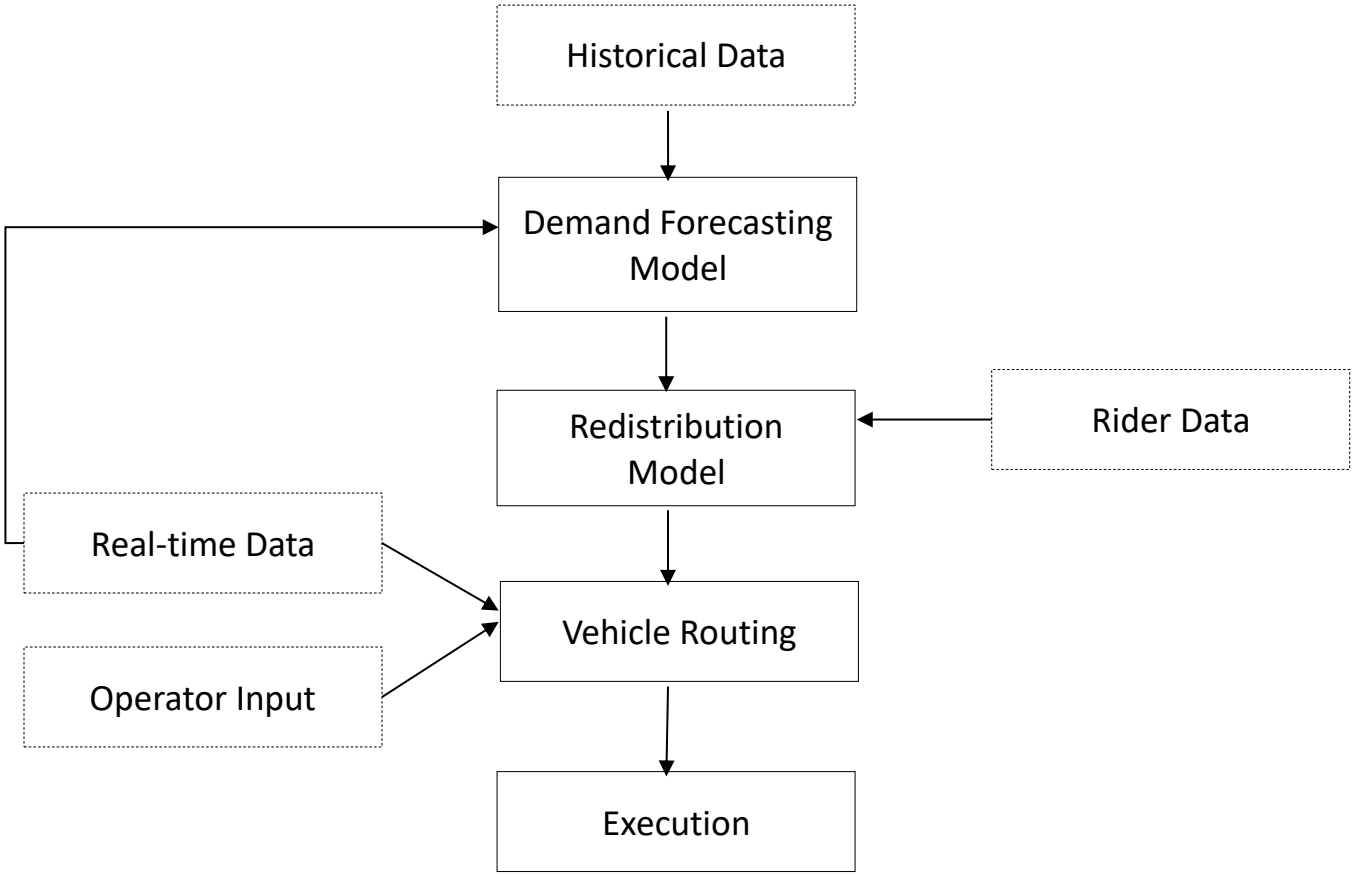


Figure 1-1 The interaction of the bike prediction model with other operational models (10)

Researchers have used different methods to predict bike counts at stations, such as regression, count models (*11*, *12*), exploring algorithms (*13*, *14*), machine learning algorithms (*15*, *16*), and time series techniques (*14*, *17*). All of these methods use many input variables, such as weather and time information, making them complex. Additionally, these models generally are static rather than dynamic, meaning that they do not adopt dynamic change over time.

The second step of rebalancing efforts is moving bikes between stations using either static or dynamic approaches. For both static and dynamic rebalancing approaches, BSS operators usually use a fleet of trucks to redistribute the bikes. Static rebalancing, or the Static Bicycle Repositioning Problem (SBRP), assumes that the number of needed or excess bikes at each station either remains the same or changes slightly during the redistribution process, and is usually done at night. The Dynamic Bicycle Repositioning Problem (DBRP) assumes that the number of needed or excess bikes at each station may change significantly. Both rebalancing approaches assume the location and number of bike stations are fixed and that only the bikes can be moved. This is a realistic assumption given that current BSSs only have fixed stations. However, cities are dynamic and their geographical and economic growth affects the distribution of trips in cities and thus constantly changes BSS users' behavior. In addition, work-related bike trips cause certain stations

to face a high-demand level during weekdays, while these same stations are at a low-demand level on weekends, and thus become useless (*18*). Moreover, fixed stations fail to accommodate big events, such football games, holidays, or sudden weather changes.

One solution for adapting to these challenges is installing and reinstalling stations; however, this is costly and impractical. Taking a different approach, a new generation of BSSs was introduced in China in 2015—the dock-less (or station-free) BSS takes an approach in which the BSS does not have stations. Rather, bikes are distributed along city sidewalks. Residents and visitors can rent a bike from anywhere and leave it within a defined zone. Although this innovative approach partially overcomes the issue of imbalance and gives bikers more flexibility, it does create other problems. First, this system has created chaotic parking problems in high-density cities where users leave their bikes in inappropriate locations, especially during rush hours and in the city center and at tourist sites (*19*). Second, in low-density cities, bikes are often left in remote locations and thus become sparse in the city, making it more difficult for users to find a bike. Eventually, the efficiency and reliability of the BSS will be affected negatively and as a result, customer satisfaction and the BSS's revenue decreases.

1.2 Problem Statement

The significant increase in the use of BSSs raises the issue of imbalance in the distribution of bikes. Some stations are at capacity and others are empty. This issue creates logistical challenges for BSS operators and may discourage bike riders, who could find it difficult to pick up or drop off a bike. To address the problem, recent research has been conducted on rebalancing the distribution of bikes at stations (*5, 6*). These approaches were built using both static prediction models and rebalancing (either static or dynamic) approaches.

Some prediction models use a statistical model to predict the demand at any given station, while others use clustering algorithms, such as traditional and non-traditional clustering (*20*). A crucial part of the prediction process is quantifying the effect of weather conditions and other factors on the bike count at stations. Extensive research efforts have been conducted using statistical and machine learning approaches to determine the correlation between bike availability and other factors and thus which factors are significant (*12, 21, 22*). Although previous approaches show promising results in predicting bike counts at stations, they suffer from three major drawbacks. First, they fail to capture dynamic changes over time, making an inaccurate assumption that users' activity will remain the same in the future and neglecting the changes that dynamic cities or new technology may bring. Consequently, these models produce constant coefficients and/or static decision rules that do not evolve with time. These models do not take into account the continuing efforts of BSS operators to keep the system balanced. For example, modern BSSs have adopted an app that can alter bikers' behavior based on the status of nearby stations. BSS operators attempt to incentivize bikers to change their origin or destination in favor of keeping the system balanced (*7, 23*). The second drawback of the existing machine learning approaches mentioned above is that they are sophisticated models using too many variables (19 variables in (*24*)) and some are difficult to interpret. Thirdly, they work poorly when encountering missing data, and thus the algorithms must rely on some sophisticated imputation techniques such as Autoclass and C4.5 (*25*). BSS datasets, as with any dataset, suffer from missing data due to malfunctions or measurement error in data collection. Additionally, it is very common for some bike stations drop out of service due to rebalancing efforts or technical issues, leading to more missing data.

As to the rebalancing approaches, the current state-of-the-practice models have extensively investigated the efficiency of both approaches: SBRP and DBRP, and DBRP has substantially advanced repositioning operations compared to SBRP. However, all existing approaches assume that stations are fixed, ignoring the dynamic spatial-temporal demand. For example, recent studies have shown the pattern of use differs significantly on weekdays and weekends, so some stations may become useless at certain times and days (*14*, *18*). In addition, (*18*) showed that some stations experience imbalance only during specific weekdays but have low-demand on the other days of the week. As a real-life example, the GoBike BSS in the San Francisco Bay Area opened in 2013 and the locations of stations have changed significantly since then (*4*). Changes such as this are due to the fact that cities change dynamically and thus the distribution of trips evolves, following new business and entertainment locations.

Assuming that stations are portable takes advantage of both types of BSS: dock and dock-less. This idea is supported by the fact that many bike stations, for example in the San Francisco Bay Area, are installed on streets (Figure 1-2), and thus can be easily linked to portable stations (*1*). The proposed portable stations can function as either individual stations (standalone) or as an extension of the existing bike stations. This concept is proposed to overcome the constraints of most current rebalancing algorithms in the following ways: (1) the locations of the docking stations are no longer fixed (2) the capacity (Q) of each station will become Q+X, where X represents the size of the portable station (3) the (un)loading time of bikes during repositioning operations would be zero, thus minimizing labor costs (4) there will be no time required for the portable stations to find parking, as they can be linked to the existing stations.

Consequently, the proposed BSS optimization framework will be built on a dynamic prediction model and rebalancing approach using portable stations.

Figure 1-2 Off-street bike station located at 594 Howard St., San Francisco (Source: Google Earth)

1.3 Research Objectives

According to the discussion above and in light of the noted limitations, the research effort presented in this dissertation intends to develop a BSS optimization framework, covering the

tactical and operational levels. Specifically, it consists of two components: predicting the bike counts at stations using online and incremental learning approaches and then balancing the system using portable stations. The goal is to propose a framework to solve the dynamic bike sharing repositioning problem. In order to fulfil this research goal, five specific objectives need to be achieved:

a) Develop a multi-objective supervised clustering algorithm to identify the similarity of bike-usage with respect to time events.
b) Propose dynamic, easy-to-interpret, rapid approaches to predict bike counts at stations in a BSS.
c) Develop a univariate inventory model using a Markov chain process that provides an optimal range of bike levels at stations.
d) Investigate the advantage of having portable bike stations with two heuristic approaches, using an agent-based simulation approach as a proof-of-concept.
e) Develop a mathematical model considering portable bike stations to be integrated with simulation as a future work.

1.4 Research Contributions

By accomplishing the aforementioned objectives, this dissertation develops a BSS optimization framework that integrates previous models into one system capable of simultaneously predicting and rebalancing. The proposed system aims to minimize the unmet demand, leading to increased user satisfaction and reducing repositioning/rebalancing operations. This is due to its ability to accommodate the dynamic change of trip distributions, allowing BSSs to accommodate dynamic demand over time without changing the infrastructure, resulting in a less-cost-and-effort solution. Different simulation scenarios in terms of the size and the number of portable stations were carried out with respect to both customer satisfaction (represented by missed bike pick-ups) and imbalanced operation (represented by deviation from the optimal status) and optimal settings were chosen. This dissertation can significantly contribute to increased revenue for BSSs operators and reduce the imbalance in BSSs compared to other DBRP approaches. Finally, this approach contributes to the development of a fifth generation of BSSs.

1.5 Dissertation Layout

Followed by the introduction, which describes the problem statement and dissertation objectives and contributions, an extensive review of the literature relevant to topics covered in the dissertation is presented in chapter 2. Thereafter, chapter 3 develops a multi-objective supervised clustering algorithm to identify the similarity of bike-usage with respect to time events. Chapter 4 introduces a univariate inventory model using a Markov chain process that provides an optimal range of bike levels at stations. Chapter 5 proposes dynamic, easy-to-interpret, rapid approaches to predict bike counts at stations in a BSS. Chapters 6 and 7 investigate the advantage of having portable bike stations, using an agent-based simulation approach as a proof-of-concept. Two heuristic approaches were proposed along with a mathematical model to be integrated with simulation as a future work. Finally, the last chapter summarizes the contributions and findings of the dissertation, followed by recommendations for further enhancements.

CHAPTER 2 LITERATURE REVIEW

This chapter goes into the main topics related to BSSs and the previous prediction and imbalance models. Section 1 presents the history of BSSs, highlighting the ups and downs through four generations, followed by a short investigation of the reasons for the remarkable explosion of BSSs after the 2000s. Section 3 discusses the main usage of BSSs and the typical users. Section 4 demonstrates the imbalance problem and its causes. Sections 5 and 6 review the previous optimal BSS inventory and prediction models. Section 7 summarizes the previous rebalancing BSS models, including the two approaches: SBRP and DBRP. Section 8 concisely covers the drawbacks of the dock-less BSSs, followed by the conclusion and summary.

2.1 Overview of the History of Bike Sharing Systems/Why Do We Need BSSs?

BSSs have been shown to be an energy-efficient and reliable transportation mode, and have been introduced in 1,139 cities in over 50 countries (*26*). According to the National Association of City Transportation Officials, in the U.S, in 2016 alone there were over 28 million bike share trips, an increase of 25% compared to 2015 (Figure 2-1). In addition, the increase continued in 2017, especially when dock-less bike share systems (BSSs) launched in the U.S. That led the number of bikes to double in 2017 compared to 2016 (from 42,500 to 100,000 bikes). This increased bike usage led many cities to either expand their existing system or launch a new one. For example, Ford, the operator of the GoBike BSS in the San Francisco Bay Area, started the system in 2013 with 700 bikes and 70 stations, and now plans to expand their system to 7,000 bikes and over 300 stations by the end of 2018 (*4*).

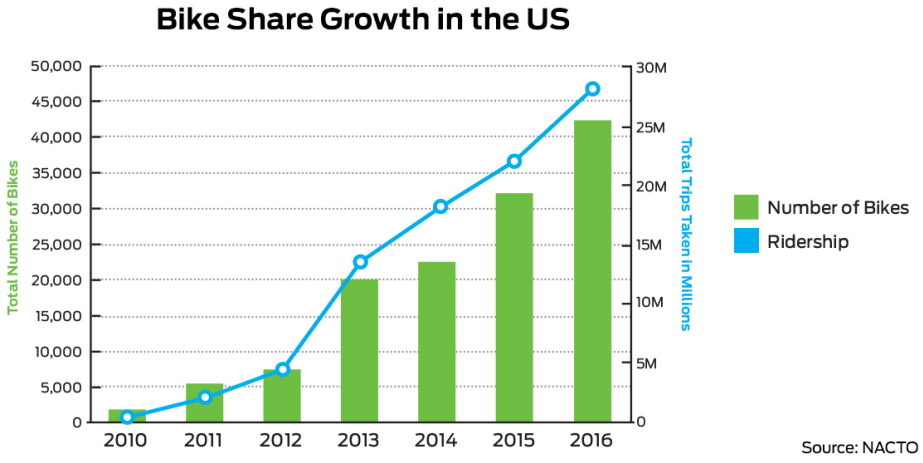


Figure 2-1 Growth rate of BSSs across the U.S. between 2010 and 2016 [NACTO]

This success did not occur overnight; BSSs have had ups and downs since the 1960s. There have been four generations of BSSs, each with its own strengthens and weaknesses (*1, 27*). In the following sections, each generation will be discussed separately.

BSSs were first proposed and implemented in July,1964 in Amsterdam (*1*). The concept of the first generation BSSs was similar to our current dock-less BSSs where a user can borrow a bike from anywhere and then return it anywhere, but bike use was free, and thus there was no lock or user ID needed, similar to Figure 2-2. These BSSs were mainly community-based initiatives. Unfortunately, users did not follow the rules and did not return the bikes after reaching their destinations. They either kept the bikes as personal property or put them in inappropriate places,

such as canals, leading the system to fail. Although this type of BSS is easy to use, it has major drawbacks: user anonymity, no deposit required, and zero revenue. The first two drawbacks do not encourage users to return the bike, while the third offers no money to improve the BSS and/or substitute any missing/damage bikes.

Figure 2-2 "White bicycles for free use, in Hoge Veluwe National Park, the Netherlands" [Wikipedia]

Thirty years later, BSSs returned to the world, this time with coin-deposit systems, meaning users had to pay to use the system but would get their coins back when returning the bike. This was an effort to overcome the theft drawback of the first generation BSSs and to incentivize users to return the bikes. In Copenhagen, Denmark, the Copenhagen "Bycyklen" was the leading BSS of the time. Given that the system was free, the company used advertising plates to generate revenue. Unlike the first generation, the second generation of BSSs had a designated place for parking bikes. However, the BSS operators did not know their users' identity, as no ID or credit card was required for system use. Therefore, these BSSs were still not secure enough from theft, leading to system collapse years later.

Another disadvantage of the first and second generations of BSSs was that the operating companies knew almost nothing about users' behaviors or needs. The bikes did not have GPS or any information technology (IT) devices. Therefore, the service did not meet users' expectations and failed to be a reliable transportation mode. The operating company had no clue as to addressing rebalancing problems, as they didn't have any information about the available bikes at each station.

Further, they didn't know the BSSs' transition matrices, so it was impossible to tell where the bikes came from or where they went.

Eight years later, a third generation of BSSs was introduced, this one much more advanced than previous generations, thanks to IT. This generation of BSSs was introduced with smart cards, which users received after registering by entering their personal information. The first BSS of the third generation was installed in Rennes, France. To incentivize people to use the BSS, they offered a free 30-minute use period. To support the huge improvements to the system, users paid to compensate the system and keep it running.

This generation overcame the main drawback of the first and second generations: user anonymity, thus protecting the system from theft. The third generation also features real-time availability and GPS tracking, which improved the BSSs significantly, providing operators more insights about the system. Operators can extract much information about the bikers' usage, such as distance traveled, routes used, and origin and destination. Further, they can determine the speed of any part of the trip, potentially improving the infrastructure of the transportation system, making it suitable for bikers.

In addition, having a large amount of historical bike availability data at stations helps predict bike counts at stations. Researchers have used different methods, such as regression, count models (*11*, *12*), clustering, and exploring algorithms (*13*, *14*), machine learning algorithms (*15*, *16*), and time series techniques (*14*, *17*), to predict bike counts at stations.

Due to the unbalanced spatial-temporal demand of bike trips, many bike stations become empty or full during the day. This significantly affects the reliability and usefulness of the BSS, which may prompt riders to return to using their personal cars or to adopt another transportation mode, consequently increasing congestion and thus auto emissions and pollution. This in turn, likely leads to a decrease in the number of BSS users, reducing the system's revenue. Operating agencies have recognized the imbalance issue and have started to establish more bike stations close to one another, aiming to keep them within no more than a 5-minute walk (*4*)[4][3][27]. However, this solution is difficult to implement, both financially and practically.

Researchers investigating the imbalance issue have recommended potential solutions to mitigate this issue with minimal cost and effort. Generally, these efforts can be categorized into three major approaches: static, dynamic, and incentivized. The underlying concept of the first two approaches is to move bikes between stations using a fleet of trucks either during or at the end of the day (*28-31*). The incentivized approach aims to encourage bikers to change either their origin or destination in favor of balancing the system (*32*).

One issue for third generation BSSs is their inability to accommodate disabled or older people, as these individuals cannot ride a regular bike for a long time. Additionally, children (< 16years old) and people without bank accounts are unable to use the system. Further, bikes are still not a convenient transportation mode for cities with steep slopes. To address two of these issues, E-bikes were introduced in this generation of BSSs. E-bikes require less energy to ride, as the bike can switch to electronic mode as needed. This addition has improved the BSS and increased its users population to include disabled and older people. The addition of e-bikes has also increased the BSSs' coverage to include steep areas. Further, BSSs have become IT-based, meaning they feature demand-responsive rebalancing. Bikers now can play a role in keeping the system balanced (self-rebalancing). Many apps on smartphones encourage bikers to be part of the rebalancing process by alerting to deviate slightly from their destination in favor of system balance. Additionally, BSSs are now being connected financially and logistically (multi-mode) with other transportation modes, such as trains or buses. People can use the same smart card to access both facilities with a minimum cost. For example, on April, 2018, the biggest commercial ridesharing company in the world (Uber)

acquired an e-bike dock-less sharing system called JUMP for approximately $200 million. Uber's plan is to dedicate short trips for the e-bike system and long trips for Uber's drivers. This intended to benefit both users and drivers. For users, e-bikes would be more affordable than cars, while drivers would benefit by increasing their income by making more long trips and fewer short trips, leading to improved mobility in the city, as traffic jams would eventually decrease.This generation of BSSs is often called "plus third" or fourth due to these significant additions.

2.2 Why Did Bikeshare Systems Increase So Dramatically After the Early 2000s?

In addition to the factors discussed above, there are other potential factors that led to the success of the current generation of BSSs:

1. **More congestion and dense cities**: Due to rapid world-wide population growth, large, dense cities are struggling with traffic congestion. Many people have migrated from rural to urban areas, creating crowded cities with limited resources. In 2007, the estimated loss attributed to missed productivity and wasted fuel was $87.2 billion. This number rose to $115 billion in 2009 in the US (*33*). Traffic jams are one of the critical issues from which urbanized areas suffer. Driving in congested areas also results in long trip times. For instance, in 1993, drivers experienced a delay of about 1.2 minutes per kilometer on arterial roads (*34*). Congestion and traffic jams were an issue before the 2000s, but has worsened significantly with the increase in vehicles and population. Taking a closer look at the bike trips occurring every day shows that 85% occur in five big and dense cities—New York; Washington, DC; Miami; Chicago; and Boston as shown in Figure 2-3 (*35*)—that congestion was a stimulus for the explosion of BSSs after the 2000s.

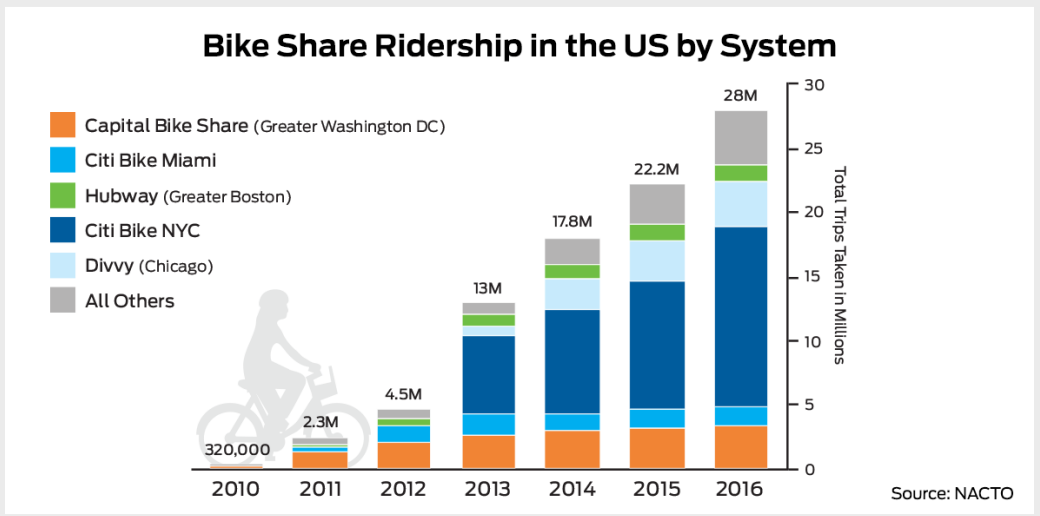


Figure 2-3 Comparison of the ridership between the biggest five cities [NACTO]

2. **Last mile problem:** Other transportation modes, such public transportation services, have attempted to mitigate congestion and accommodate the increased number of city resident trips, but only partially reduce congestion. Trains and buses generally stop at transit stations near the city center, and thus some riders need other transportation modes to get to their final destinations. This is usually referred to as "the last mile problem," which is defined as "the short distance between home and the nearest public transit or between a transit station and the workplace, which may be too far for a walk" (*27*). A well-operated and maintained

BSS can help address this issue, enabling riders to reach their final destination, while also offering the potential to relieve roadway congestion. A recent study of the BSS in New York shows the system is mainly used for short trips (< 10 min.) happening close to transit hubs (*35*). These trips are considered as "too long to walk but seem too short for a subway trip".

3. **Better quality of service**: Technology has played a role in making BSSs affordable, convenient, and easy-to-use. This includes improvements in station density, bikes per resident, and coverage area. The introduction of BSSs with smart cards, for which users register using their personal information, have also made BSSs more appealing. In addition, many BSSs offer an app for bikers that provides necessary information, such as nearby bike stations, bike dock availability, and operation hours. This app could help operating agencies mitigate the imbalance problem by sending suggestions to incentivize bikers to change their origin or destination in favor of keeping the system balanced.

2.3 Who are BBS Users and How Are They Different from Traditional Cyclists? What is the Main Use of BSSs?

The use of BSSs with regard to gender, age, car ownership and ethnicity changes from one city to another and from one type of user to another (short-term user to long-term user). Previous research efforts show different results when taking surveys or analyzing real data. For example, in Washington, DC, Buck et al. found that males use BSSs 30% more than females, yet a survey shows the distribution of males-to-females to be equal (48–52%) (*36*). Naturally, actual demographic data is more accurate than survey (self-reported) data, as it reflects real conditions. In the UK, a survey was conducted to investigate a BSSs' users. Results showed the same conclusion: that men use BSSs more than women (by 17%).

Buck et al. studied the demographic of BSS users and traditional cyclists in DC. The authors found the user demographic to be mostly white, around 80% for both traditional and BSS cyclists, as shown in Figure 2-4. Compared to the overall Washington DC population, the population of black people is slightly bigger than white people (by 5%). That can lead us to conclude that white residents in D.C. are more likely to use bikes (either BSS or their own bikes) than other races, confirming that the survey is accurate and not merely a reflection of overall city demographics. A further investigation leads us to conclude that education is a crucial factor bike use. Surprisingly, as income increases, the chance that a person will choose to ride a bike increases significantly, especially for people making more than $100k.

As expected, younger riders are more likely to use bikes than seniors, where the age group of 24–34 is the largest age group by 55%, 48%, and 24% for traditional cyclists, short-, and long-term BSS users respectively. The slight distinct difference between traditional cyclists and BSS users are Asian and African users. As for age, in general, the age distribution is similar for both BSS users and traditional cyclists, though traditional cyclists appear to be older than BSS users, as is clearly shown in the 34–44 age group (Figure 2-4).

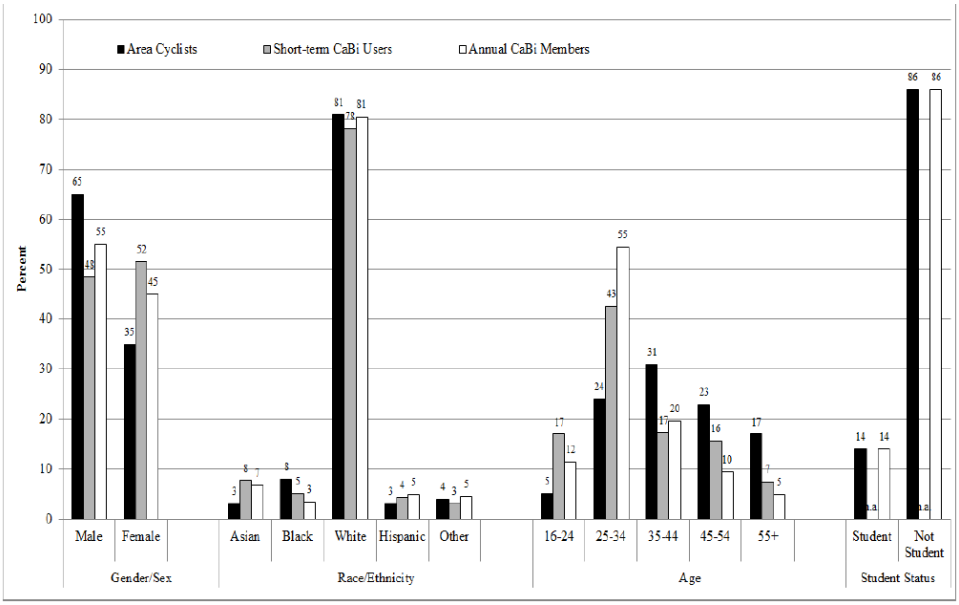


Figure 2-4 Comparison of demographics of Washington, DC area cyclists and short-term and annual Capital Bikeshare (Cabi) Members (36)

The usage of BSSs really depends on the type of user (either short or long-term). In D.C., a survey showed 43% of long-term users were work commuters while short-term users tended to have fewer work trips (*36*). Also, in London, a survey revealed the same conclusion: that long-term or annual members seems to be work commuters. This was based on a question asking the purpose of the last trip and the answer was 52% for work (*1*)[1]. Similar, another survey conducted in Brisbane, Australia showed the same results that the main purpose of the last trip was leisure for short-term members but work-related for long-term users (*37*). However, a study in 2013 shows that short- versus long-term use depends on many factors, such as gender, age, residential location, and ethnicity (*37*).

Using the Capital BSS dataset in DC, Buck et al. took a close look at the purpose of each type of user, as shown in Figure 2-5. The authors found that the main purpose of 53% of the short-term users was tourism, followed by personal. For the long-term users, the main purpose was either fitness or work.

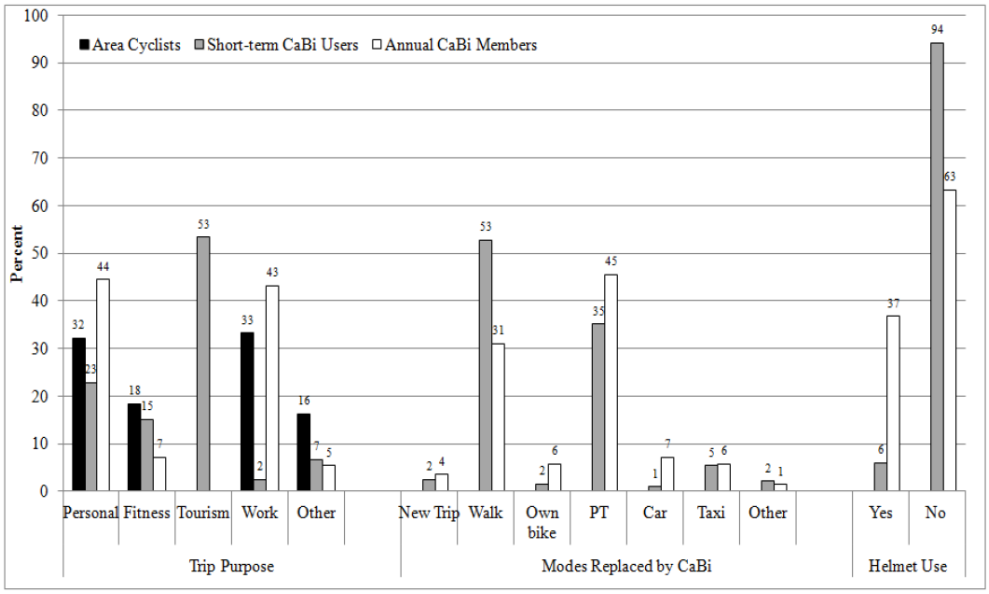


Figure 2-5 Comparison of trip purpose, transport modes replaced by capital bikeshare, and helmet usage (36)

Figure 2-5 also shows that the BSS was used a replacement for walking, followed by a replacement for public transit.

In the San Francisco Bay Area, Saltzman and Bradford found that 90% of BSS trips occur on weekdays, aggregating short- and long-term users (*38*). Of this 90%, 92% of the rides were made by annual or long-term users and only 8% were made by short-term users. In addition, the authors showed that the rides made on weekdays were work-related trips. These rides tended to be short and had less variance than the non-work related trips made by short-term users. The median of the work-related trips was 5.6, while it was 16.5 for non-work related trips.

McNeil at al. found, in the results of their survey, that 80% or more users said that they use BSSs for one of the following: shopping/errands, social/recreational, trips to and from public transit, and commute trips (*3*). Two thirds of respondents said they use BSSs for food-related trips, and half said they used BSSs for family or personal business. When asked about the frequent trips, commuting was the first answer, followed by getting to or from public transit. The authors also asked BSS riders if they had saved money by using BSSs, and about 50% noted that they did spend less on transportation.

Shaheen et al. conducted 38 interviews with experts in public bike sharing with city and regional transportation professionals, public transit agencies, community bike coordinators, policymakers, community bike organizers, and vendors (*39*). The interviews were conducted in summer 2011 and spring 2012. The experts were asked many questions regarding the use of BSSs, including the typical trip type and purpose. The main trip purpose was 42% for recreation, as shown in Figure 2-6. Interestingly, they found the largest three operators answered for non-recreational purposes. Another finding was that short-term users tended to use BSSs for recreational purposes, unlike long-term users.

Shaheen et al. also categorized the trips by point-to-point or round-trip (*39*). They found that 26% of operators indicated that point-to-point trips were the most prevalent, while 42% reported that round-trips were most common, as shown in Figure 2-6 (*39*).

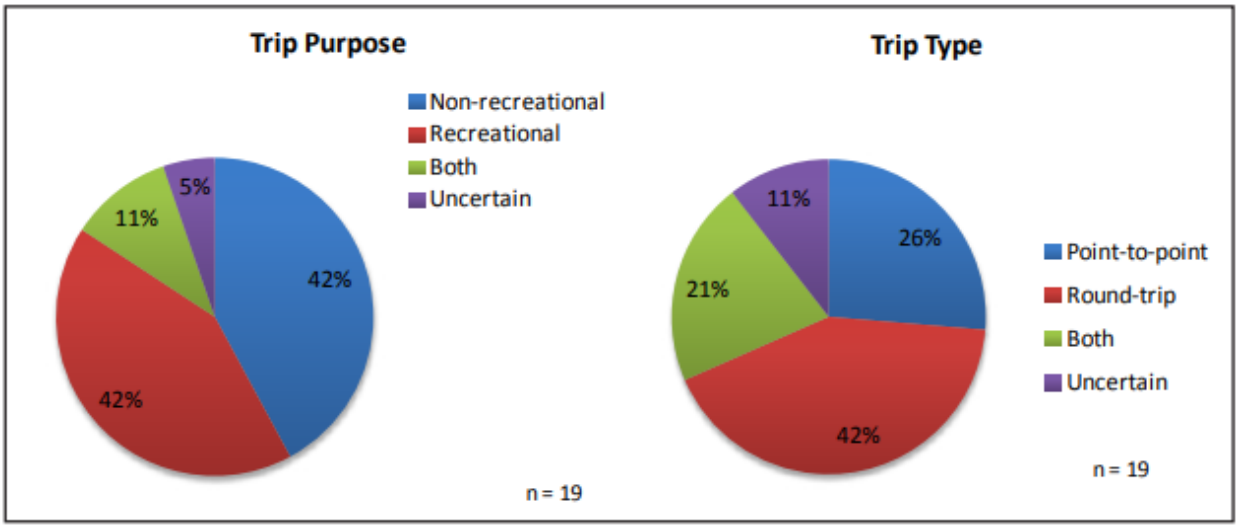


Figure 2-6 Comparison of trip purpose, transport modes replaced by capital bike share, and helmet usage (*39*)

2.4 Why Does the Imbalance Problem Occur?

This rebalancing problem is due to the unbalanced spatial-temporal demand of bike trips. This significantly affects the reliability and usefulness of the BSS, which may prompt riders to return to using their personal cars or to adopt another transportation mode, consequently increasing congestion and thus auto emissions and pollution. The causes of the imbalance problem can be summarized (in either a continuous or discrete category) as follows:

2.4.1 Unbalanced Spatial-Temporal Demand (Continuous):

A. Unbalanced temporal distribution: This happens when there are high pickups/returns in the mornings or in the late afternoon, resulting from commute trips. For example, a residential area runs empty in the early morning as people leave for work or school and becomes full later in the afternoon as users return. That can be seen clearly in BSSs operating on a college campus.
B. Unbalanced spatial distribution: This happens when there are high pickups at high elevation stations with very low returns and low elevations with very high returns. This is because users try to avoid biking on steep grades as it takes more energy. One recent study showed that, "Stations that tend to be empty are located between 80–110 meters above sea level in contrast to stations at the coast that tend to be full during the whole day." (Froehlich et al., 2009). Public transportation stations also have a great impact on the bike station status and could be a reason for the imbalance problem, as shown in (*18*).

2.4.2 Demographic or Structural Change in the City (Discrete):

Changing the infrastructure of the city (i.e., an opening/closing business or entertainment establishment) could affect the trips' distribution and thus change the demand dramatically. This leads to stations facing unexpected demand. Also, the increase in the population of the city over time results in a demand-exceeds-supply problem (which is also a function of the number of bikes relative to docking stations).

2.4.3 Others (Discrete):

This includes discrete but unexpected big events, such holidays, sudden changes in weather, one-ways trips (especially short or medium distance, as these are cheaper for users).

2.5 Optimal Inventory Models

Some research efforts have attempted to reduce the imbalance problem by determining the optimal number of bikes that should be at a station at the beginning of the day in order to minimize the probability of imbalance. Several studies have been conducted using the Markov chain process for balancing BSSs (*40-42*). Raviv and Kolka developed a single station inventory model of a BSS using a special case of the Markov chain process: the birth-death process (*42*). This particular type of Markov chain process is very limited, as it can only move one step backward or forward, meaning bike usage cannot be predicted for more than one hour (assuming the jump equals one hour). Schuijbroeka et al. modeled the inventory at each station as a finite-buffer single-server non-stationary queuing system using Kolmogorov forward equations to calculate service level requirements (*40*). Their model enabled them to introduce the concept of self-sufficient stations, which work as source or sink nodes to reduce the need for rebalancing. This proposed model has two major drawbacks. First, the authors neglected users arriving in groups. Second, the authors used the Runge–Kutta method to solve the Kolmogorov forward equations in finding the transient probabilities; the disadvantage of this method is that it increases the number of infeasible stations as the length of the observation period increases. Parikh and Ukkusuri modeled the demand at Velo Antwerpen stations in Antwerp, Belgium using the Markov chain process (*41*), but the study only used the net demand (returned minus withdrawn bikes). More importantly, the data used in the proposed model was collected for 30 minutes, making the outcome of the model vulnerable to big jumps (*42*).

2.6 Prediction Count Models

Bike prediction approaches have mainly taken one of four approaches: statistical models, exploring and clustering algorithms, machine learning algorithms, and time series models. Each approach has a different level of complexity with varying numbers of independent variables, such as time information, neighboring stations, and weather information.

2.6.1 Statistical Models

Rudloff and Lackner used three count models: Poisson, negative binomial (NB), and Hurdle models to predict bike demand using temperature, precipitation, and neighboring stations as predictors (*43*). They used bike data from the bike sharing system Citybike Wien in Vienna, Austria and concluded that the Hurdle model outperformed the other two. Wang et al. adopted log-linear and NB regression models with 13 regressors as independent parameters (*12*). These 13 regressors included socioeconomic, demographic, and geographic information. They showed that all 13 regressors were significant and fit well with both models. Rixey adopted multivariate linear regression models to predict bike ridership using demographics and built environment characteristics near the BSS (*11*). They used three bike sharing systems and concluded that the factors used were significant. Ashqar et al. investigated the significant factors on bike demand, and using Random Forest (RF) found that time-of-day, temperature, and humidity level are significant predictors in bike prediction (*44*). They adopted two count models: Poisson and NB along with RF; their results showed that RF outperforms the other two models.

2.6.2 Clustering and Exploring Algorithms

Due to the size of BSS data sets, several studies were conducted using visualization and clustering approaches and considering spatial and temporal information (*13, 45-47*). Froehlich et al. utilized a clustering approach to predict bike counts in two steps (*13*). The first step was to investigate the relationship between human behavior, geography, and time of day. The second step was to predict bike counts based on the three aforementioned factors. They divided bike stations into clusters and then predicted bike counts for each cluster. Their findings demonstrated neighboring stations were highly correlated and thus they were treated as one cluster. Similarly, Vogel et al. used clustering approaches to group stations with respect to the bike pickup and return activity (*48*). Based on the geographical information, they clustered bike stations into five groups and then provided average pickup and return rates for each hour. Kaltenbrunner et al. also attempted to improve the BSS in Barcelona using docking station data (*14*). Temporal and geographic mobility patterns were obtained and analyzed with the goal of detecting imbalances in the BSS. Subsequently, they used time series analysis techniques to predict the number of bicycles at a given station and time.

2.6.3 Machine Learning Techniques

Machine learning approaches have been shown to be promising for predictive models due to their remarkable ability to learn from the data set and account for many predictors to discover hidden data set patterns. (*15, 16*). Ashqar et al. adapted three models: RF, Least-Squares Boosting (LSBoost), and Partial Least-Squares Regression (PLSR). The authors used six weather variables, 10 nearest neighboring stations, the month, day of week, and time of day (*15*). Their analysis showed that RF outperformed the other two methods, and also that RF kept the prediction error from increasing constantly as the prediction window increased, unlike the other models. Yang et al. used deep learning (i.e., a convolution neural network) to predict the daily usage of bikes (*16*). They used weather information, neighboring stations, and day of week as inputs for the models, and showed that the convolution neural network outperformed both the neural network and the autoregressive moving integral average model.

However, the previous three approaches suffer from the following: (1) they are static models, meaning they are trained once and remain the same and thus cannot capture the dynamic change over time, (2) they require many predictors, and (3) they are computationally expensive and thus cannot be used as online models.

Machine learning algorithms can be categorized into two major approaches: batch and online (or incremental) learning approaches (*49*). The batch approach is meant to use all the observed data at once and produce fixed coefficients of the model, while the online learning approach uses the observed data once they arrive and then produces dynamic coefficients over time, leading this approach to be faster. According to the literature, the first approach (i.e., batch) has been used for bike prediction, although it suffers from the three aforementioned drawbacks. The following section will discuss the second approach.

2.6.4 Time Series Techniques

The online machine learning approach is mainly proposed to handle systems that cannot tolerate a large processing delay. Its power comes from the fact that it is flexible enough to be applied to most machine learning algorithms. A few recent studies adopted time series techniques to predict the bike counts at stations (*14, 17, 50, 51*). Although these techniques showed good performance in both explaining the past and predicting the future, they had several limitations. For example, Kaltenbrunner et al. (*14*) used an autoregressive moving average (ARMA) model to predict the bike availability at stations (*14*). However, the ARMA model is a stationary model that assumes the mean and variance of the observations are fixed over time, which is not the case in the BSS

data. Froehlich et al. proposed four models: last value, historic mean, historic trend, and Bayesian network (*13*). They showed that the Bayesian network model produces the least prediction error. Yet, the Bayesian network model was not adopted to give exact bike counts. Instead, it provided only a small number of prediction classes (in percentages); that is, the bike availability in stations was classified in even percentage intervals (for example, 25%, 50%, 75%, and 100%), and the algorithm only chose one of the four categories to describe the bike availability.

Yoon et al. proposed a spatial-temporal prediction system using an autoregressive moving integral average (ARIMA) model to overcome the non-stationary issue in the ARMA model (*17*). Seasonal trends and neighboring information were utilized in the model. A small dataset of 3 weeks was used to evaluate the model. Their results show a slight improvement in favor of ARIMA when compared to ARMA (The error is 3.47 bikes/station versus 3.50 bikes/station). However, ARIMA is considered a static model; its estimated coefficients do not evolve with time and predictions are only within even intervals. Additionally, ARIMA is a complex and hard-to-interpret model.

2.7 Rebalancing Approaches

Previous research efforts have been largely spent on two main rebalancing approaches: the SBRP and the DBRP. The SBRP neglects the bikes' movements while rebalancing the stations, so static repositioning is done overnight when there is minimal bike usage. Unlike the SBRP, the DBRP takes into consideration the bikes' movement while rebalancing, and can thus be done anytime during the day.

2.7.1 Static Bicycle Repositioning Problem (SBRP)

For the SBRP approach, researchers have typically adapted models and proposed solutions to solve the imbalance as a night problem (*29, 30, 52*). Espegren et al. proposed an optimization model aiming to minimize the deviation from the optimality for stations (*29*). The authors used a heterogeneous fleet of service vehicles to move bikes between stations that allowed for multiple visits and arrived at a non-perfect rebalancing solution. Caggiani and Ottomanelli proposed a Modular Decision Support System to rebalance the bike distribution by determining the best bike repositioning and the best relocation time horizon with the optimal carrier vehicle route (*30*). The objective function is minimizing deviation as well as the cost of repositioning using carrier vehicles. Chemla et al. used a single truck to rebalance the BSS using a branch-and-cut algorithm (*52*). The proposed algorithm aims to minimize the distance traveled by the truck. Elhenawy and Rakha proposed a rebalancing algorithm, called the deferred acceptance algorithm, based on the game theory algorithm. Their proposed algorithm had two phases: tour construction and tour improvement. The objective function was to minimize the total tour cost (*15*). Kadri and Kacem formulated the balancing problem mathematically with two lower and four upper bonds (*53*). The two lower bonds were developed based on Eastman's bound while the four upper bonds were based on a genetic algorithm. These bonds were incorporated in a branch-and-bound algorithm. The authors used a fleet of vehicles and aimed to minimize the duration of imbalanced stations.

These aforementioned studies using the SBRP approach assume the user's demand to be negligible while performing the repositioning operations. Consequently, rebalancing efforts are conducted at the end of day, making rebalancing a day-to-day operation, which means that this approach fails to prevent imbalance during the day. As a response, researchers investigated a faster approach, the DBRP, to reposition bikes dynamically by allowing repositioning decisions to be adapted over the planning horizon. The DBRP approach was shown to give a better result than the SBRP approach due to its ability to rebalance continually during the day, which will be discussed in the following section (*10, 26, 28, 31, 54-56*).

2.7.2 Dynamic Bicycle Repositioning Problem (DBRP)

The DBRP approach was shown to be more realistic and more powerful given that it considers the movement of bikes during the rebalancing operation and also the ability to rebalance the system during the day (*10, 26, 28, 54*). Brinkmann et al. attempted to solve the imbalance in a BSS by formulating the problem as a stochastic inventor routing problem (*31*). The objective function was to minimize the number of times the bike station either runs out of bikes or reaches its return limit. They proposed a short-term strategy that determines which stations are more likely to be out of service. This strategy and the long-term relocating strategies were used to compromise between the number of served stations and relocation operations with the goal of minimizing the number of times a station runs out of bike or reaches its limit. Contardo et. al proposed a dynamic approach to address the imbalance in a BSS using Danzig-Wolf and Benders decomposition (*28*). A scalable methodology to provide lower and upper bounds was developed. The objective function was set to minimize the total unmet demand (i.e., deviation). Regue and Recker proposed a framework to solve the repositioning problem using proactive dynamic vehicle routing along with machine leaning techniques. Four models were proposed: a demand for casting model, a station inventory model, a redistribution needs model, and a vehicle-routing model (*10*). First, they determined the future station inventory levels using different datasets. Then, they formulated the problem as a stochastic linear integer problem to estimate the number of needed bikes at each station at a given time. After that, pickup and drop off activities were determined and used as an input to the vehicle routing problem. It should be noted here that the output of the pickup and drop off are deterministic. Ghosh et al. used a mixed integer linear programming approach to rebalance a BBS (*26*). The objective function of the problem was maximizing the served demand and minimizing the cost caused by using balancing carrier vehicles. Then, they solved the problem using two steps: finding the repositioning solution for bikes and the finding the routing solution for carrier vehicles. To simplify the problem and speed up the solution process, they clustered the bike stations with respect to their location. BSS data from in Washington, DC and Boston, MA were used, and a significant improvement in the operational efficiency was shown. Chiariotti et al. proposed a dynamic model using birth-death processes (*55*). They predicted stations' statuses and then determined the optimal time for repositioning operations. The graph theory was used to choose the optimal path to order service vehicle destinations.

Although the DBRP has advanced repositioning operations substantially, all existing approaches assume the stations are fixed, ignoring the dynamic spatial-temporal demand. For example, recent studies have shown the pattern of use differs significantly on weekdays and weekends, making some stations useless at certain times and days (*14, 18*). In addition, (*18*) showed that some stations experience imbalance only during specific weekdays but have low-demand on the other days of the week. As a real-life example, the GoBike BSS in the San Francisco Bay Area opened in 2013 and the locations of stations have changed significantly since then (*4*). That is due to the fact the city changes dynamically and thus the trips' distribution evolves, following new business and entertainment locations.

2.8 Can Dock-less BSS Solve the Imbalance Problem?

According to the National Association of City Transportation Officials in the U.S, 44% of the BSSs in the U.S. were dock-less by the end of 2017. The dock-less BSS could help solve the imbalance issue by preventing bikers from ever finding a full station, as they would have more flexibility to leave bikes almost anywhere (i.e., within a limitless designated area instead of a specific location with a limited capacity, as in the dock-based BSS). Using dock-less a BSS, bikers would be able to minimize the walking distance to their final destination, providing a better experience than with the dock-based BSS where bikes must be returned to stations, though they

might have difficulty finding a bike, as the bikes are left by users in random places. Consequently, it is fair to say that dock-less based BSSs are better with regard to users reaching their destination but worse when the user is departing their origin point, and vice versa for dock-based BSSs. From the BSS operating agencies' standpoint, dock-less BSSs are costly to rebalance, as bikes are placed in random locations, unlike dock-based BSSs, in which bikes are left only in predetermined locations in the city. With regard to the initial conditions of bike stations (as in dock-based BSSs) or the location of the bikes (as in dock-less-based BSSs), it is much harder to find the optimal locations of bikes in dock-less BSSs than to find the optimal condition of bike stations.

To sum up the disadvantages of the dock-less BSS:

A. Dock-less BSSs create chaotic parking problems in high-density cities where users leave their bikes at inappropriate locations, especially during rush hours, in the city center, and at tourist sites (*19*).
B. In low-density cities, bikes are often left in remote locations and thus become sparse in the city, making it more difficult for users to find a bike. Eventually, the efficiency and reliability of the BSS will be affected negatively and as a result, customer satisfaction and the BSS's revenue decreases.

2.9 Summary and Conclusions

The literature review first provides an overview of the history of BSSs, describing the four generations. The reasons for the failure of the first three generations were presented, followed by a further discussion of the fourth generation, shedding light on the reasons for its success. BSS user types were also discussed and compared with traditional cyclists in regard to demographic distributions, such as gender, age, car ownership, and other factors. In addition, the causes of the imbalance problem were summarized into three categories (1) unbalanced spatial-temporal demand (2) demographic or structural change in the city (3) unexpected big incidents.

A brief literature review on bike prediction models was conducted, covering the four types of prediction count models: statistical models, exploring algorithms, machine learning algorithms, and time series models. The drawbacks of each type, in particular, were examined: (1) having constant parameters and thus failing to capture the dynamic change over time (2) being complex with too many predictors (3) being computationally expensive. To overcome these drawbacks, it is important to develop quick, easy-to-interpret, and dynamic predictive models.

The literature relative to the two main rebalancing approaches, SBRP and DBRP, reveals that both approaches assume the bike stations are fixed, ignoring the dynamic spatial-temporal demand. Consequently, this dissertation aims to overcome this drawback and provide a new generation of BSS in which some stations are portable, meaning they can move during the day. These can be either stand-alone or an extension of existing stations with the goal of accommodating the dynamic changes in the distribution of trips during the day.

CHAPTER 3 NOVEL SUPERVISED CLUSTERING ALGORITHM FOR TRANSPORTATION SYSTEM APPLICATIONS

This chapter is based on the paper listed below:
M. H. Almannaa, M. Elhenawy and H. A. Rakha, "A Novel Supervised Clustering Algorithm for Transportation System Applications," in IEEE Transactions on Intelligent Transportation Systems. DOI: 10.1109/TITS.2018.2890588

3.1 Introduction

With the growth of new technologies, smart cities and urban areas are adapting advanced devices to control and monitor transportation networks and thus provide better service to the public and private sectors. These devices collect data through many sensors in the city's infrastructure. Agencies and researchers exploring the massive amounts of collected data often find it challenging to draw meaningful conclusions due the sheer size of the datasets. One way to deal with such data is to use clustering approaches.

In the transportation field, operating agencies (such as departments of transportation) have been collecting data to improve the efficiency of the transportation network and provide a better service for all transportation modes. Clustering the travel times or speeds of transportation modes could help operating agencies to better manage the transportation network. In particular, the collected data could be reduced to find the cluster centroids (i.e., the means of the clusters) that represent the entire data with respect to a time event such as time of day, day of month, and month of the year. This could help operating agencies answer several questions related to traffic operations such as, "Can we discriminate between recurrent congestion and outliers?" and "Can we identify how many time periods we need to plan for in terms of resource and congestion management?"

Clustering is an unsupervised learning technique that identifies the underlying structure of unlabeled data. The goal of clustering is to identify intrinsic groupings in an unlabeled dataset. Meaningful clustering depends on the clustering criterion used by the clustering algorithm. Accordingly, it is crucial to find the best criterion so that the clustering results will suit the needs of researchers and agencies.

Clustering algorithms are used in many disciplines, such as computer vision to segment images (*57*), marketing to find similar customer behaviors (*58*), the insurance industry to identify fraud (*59, 60*), and in transportation to identify similar patterns in various modes of transport (*61-63*). Clustering helps develop a deep understanding of similarity in data patterns. For example, traffic engineers can use clustering algorithms to identify similar traffic patterns on a highway during the day, week, or month, and then make use of the clustered patterns in the management of the system. Clustering has also been used to analyze bike sharing system (BSS) data (*13, 48*). Some researchers have used a statistical model to predict bike availability at each station, while others have used clustering algorithms, such as traditional and non-traditional clustering (*20*). Traditional clustering approaches, such as the k-median, DBSCAN, and fuzzy algorithms are good tools for clustering data, but give narrow results, as clusters are based on only one factor (i.e., distance or similarity). These clustering algorithms are unsupervised clustering that divides the observational points into clusters based on an objective function without considering natural labels in the dataset, such as the time of events (i.e., month of year, day of week, or time of day).

Recently, supervised clustering (non-traditional) approaches have been widely embraced as powerful tools that can take advantage of other attributes (labels) in the dataset (*45, 46, 64*). Unlike traditional clustering techniques, the supervised technique clusters labeled data. Supervised algorithms use data labels to represent natural data groupings using the minimum possible number of clusters. Only the labels are used as an objective function, and distance and similarity are ignored (*64, 65*).

In this chapter, we propose a new supervised clustering algorithm based on the college admission (CA) game theory algorithm (*66*) to maximize the reciprocal of the within-cluster sum of distances (similarity) and the cluster purity simultaneously. The proposed algorithm was used to answer several transportation-related research questions, such as which days or months exhibit similar patterns.

To evaluate the proposed algorithm, we tested it using a BSS dataset in the San Francisco Bay Area, which consists of bicycle count data. We then studied how bike patterns changed within each cluster, and addressed when and where the system would be imbalanced.

3.2 Problem Statement

Operating agencies and transportation researchers have devoted significant attention to clustering approaches with the goal of clustering large datasets that contain traffic patterns (i.e., travel times or speeds) in transportation networks (*61-63, 67*). They have adopted classical approaches such as k-means, Ward's hierarchical clustering algorithms, and density-based clustering. The purpose of using these clustering approaches is to (1) cluster traffic patterns with respect to a time event so that operators can have a temporal plan for operations planning purposes, and (2) discriminate between recurrent congestion and outliers. However, the aforementioned studies used classical clustering approaches that do not take advantage of natural time event labels (such as time of day, day of week, etc.). As for unsupervised clustering algorithms, they implicitly assume that clustering the data points based on similarity or distance leads to the ground truth of the clustering, which is not necessarily true. These algorithms cannot consider both similarity/distance and other domain knowledge information in the objective function. Consequently, clustering solutions do not help operators map the clustering solution to the network demand with regard to time events (*68, 69*).

In this research, we present a supervised clustering algorithm that attempts to find similar months, days, or hours within a day that have similar traffic patterns. We sacrifice the exact centroids of traffic patterns on account of having similar time events. The proposed algorithm is scalable (polynomial order), fast, and ready for practitioners. It makes no assumptions about the dataset and requires only one parameter, namely the number of clusters, which can be found using the consensus clustering (CC) technique (Section 3.8). It compromises between distance and purity in identifying clusters within the data.

3.3 Related Work

Clustering algorithms can be categorized into three main approaches: unsupervised (i.e., traditional), supervised, and semi-supervised. Unsupervised clustering algorithms assume the data are unlabeled (i.e., the relationship is unknown between the data points) and thus try to cluster them according to similarity or distance (*70*). They implicitly assume that clustering the data points by distance or similarity leads to the ground truth of the clustering. The supervised clustering approach deals with labeled data (the relationship is known). There are a variety of supervised clustering algorithms. Some of these algorithms attempt to cluster data according to the labels (i.e., purity) and number of clusters (*64*). Another algorithm uses the labels to learn the best similarity measure that produces the desirable clustering solution (*71*). The semi-supervised clustering algorithms assume that part of the data is labeled and the rest is not. The known labels can be used to form constraints between pairs of data points in the form of must-link and cannot-link (*72, 73*) (which will not be covered in this chapter as it is very different).

Two examples of unsupervised clustering algorithms are the well-known k-means and hierarchical clustering algorithms (*74, 75*). The k-means simply partitions the data points into clusters, minimizing the distortion of each cluster (*74*). The value of the model order (k) is set by the user based on personal knowledge or is chosen to maximize some criteria such as the clustering stability. At each iteration, the k-means algorithm assigns all the observation points to the clusters and updates the centroid of each cluster. Eventually, the k-means algorithm converges when the centroids stop moving.

The hierarchical clustering algorithm is a tree-based structure. It does not require the modeler to specify *k* apriori. Moreover, the dendrogram can be utilized to select the optimum number of clusters (*75*). At every level of the tree-based structure, similar clusters are merged into one cluster. The key to this clustering algorithm is the criteria determining when and which two clusters can be merged. Different approaches are used, such as single linkage and complete linkage. The only difference between this algorithm and the k-means is the use of a similarity measure between clusters besides data points, but both use only similarity or the distance measure. More advanced unsupervised clustering algorithms have been proposed such as kernel k-means (*76*), kernel self-organizing maps (*77*), and kernel fuzzy c-means (*78*). These algorithms attempt to cluster the data points by transforming them into a higher dimensional feature space and then carry out the original clustering algorithm, which is based on the similarity or distance without considering other domain knowledge information.

Supervised clustering algorithms go a step further and endeavor to improve the unsupervised clustering algorithms by incorporating purity (i.e., labels) in the objective function (*64, 65*). Purity means using labeled data to identify clusters that have a high probability density with respect to a single class. Eick et al. proposed four different supervised clustering algorithms with the same objective function containing a linear combination of impurity and number of clusters (*64*). The aim is to minimize impurity and the number of clusters. However, these algorithms do not consider the similarity or distance measure. Spinellis proposed a supervised clustering algorithm called Box Clustering that clusters data points into specific convex polygons with a fixed cluster impurity (*65*). Similar to Eick et al.'s work, similarity was not incorporated in the objective function. Another approach to supervised clustering algorithms was given by Awasthi and Zadeh. They assumed there is access for a teacher that can help improve the purity of the clusters (*79*). Yet, this approach assumes that the teacher knows the ground truth of the data, which is not always the case in many datasets (i.e., assumes we have two datasets: training and test).

Recently, supervised clustering algorithms have been enhanced greatly by using a multi-objective approach (*80-84*). This approach aims to optimize several clustering criteria such as similarity or compactness of the clusters and connectivity of the clusters. The goal is to compromise between these objective functions and produce a trade-off solution. This has led them to be widely introduced in data mining as a powerful way to effectively classify labeled datasets. Law et al. proposed a multi-objective approach in a two-step process (*83*). In the first step, they used different clustering algorithms with different goals, and in the second step they integrated the output into a single partition. The labels of the datasets were only used for evaluating the clustering results but not in the objective function. Handi and Knowles proposed a multi-objective evolutionary algorithm, maximizing the compactness and connectivity of the clusters simultaneously (*84*). This approach (i.e., the evolution optimization algorithm) gives many possible solutions (so called population approach) at each iteration, and thus the authors used a Pareto-based approach to select the non-dominate solutions that were created by the proposed algorithm.

None of the previous approaches used both purity as well as similarity in the objective function. Only a few supervised clustering algorithms had both purity (i.e., background information) and distance or similarity in the objective function, yet they suffer from complexity and having many assumptions and parameters, making them hard to interpret [24, 25]. For instance, Marcu used the Dirichlet process prior to using a Bayesian approach to incorporate both similarity and purity (*80*). This approach is considered a generative model, meaning it estimates the joint probability distribution of the data between the observed data and the corresponding labels. This algorithm suffers from several drawbacks: (1) it is complex—one has to define the distribution of the data (which is usually unknown) and also has to use the Markov Chain Monte Carlo-based (MCMC) sampling to avoid intractability; (2) it cannot define a good distribution for the data due to its generative nature; and (3) it cannot deal with a large dataset, and thus the scalability is an issue. Forestier et al. proposed a collaborative clustering algorithm that incorporates three components: cluster quality, class label, and link-based constraints (*81*). This

approach selects a subset of the dataset as background knowledge randomly, causing it to be less stable. It also requires an expert who can tell which subset of the dataset to use as background knowledge.

In this chapter, we propose a new supervised clustering algorithm with the ability to increase both cluster purity and member similarity simultaneously. The proposed algorithm is scalable, quick, and simple, considering only one parameter—the number of clusters. It compromises between distance and purity in identifying clusters within the data. It showed promising performance when applied to the BSS dataset. It clustered the bike availability with respect to a time event, giving operators more practical clustering results for operation planning purposes.

3.4 The College Admission Algorithm

In 1962, Gale and Shapley proposed the deferred acceptance algorithm as a solution to the stable marriage problem, in which an equal number of men and women are matched such that no player has an incentive to leave his/her matched partner (*66*). The stable marriage problem involves one-to-one matching. The college admission (CA) problem is another version of the stable marriage problem, though in this case the algorithm matches many to one. In the CA problem, there are a number of colleges and applicants that need to be matched. Each college has a ranked list of students they prefer, and each student has a ranked list of colleges they prefer. The size of the ranked list of students depends on the capacity of the college. The best-qualified candidates are offered admission first, followed by the lesser-qualified candidates.

This problem includes the uncertainty of the colleges not knowing which other colleges the students have applied to, and thus not knowing the ranked list of each student, or whether the student has been offered admission by other colleges. Consequently, the colleges are in a blind position with very little information, which prevents them from making the appropriate decision. This can result in an unbalanced situation in which some students are offered many admissions, while others are not offered any at all. Gale and Shapley presented a stable solution where each student would be accepted to the best possible college with regard to his or her list, and each college would have the best possible qualified student.

The CA algorithm finds a stable matching solution through a series of iterations. At each iteration, the colleges offer admission to the best-qualified students, and the students have to reply back by either accepting the offer or not. At the end of the iteration, some students have an admission and others do not. Colleges then update their list accordingly in the next iteration and offer admission to students who did not receive an offer in the previous iterations, regardless of whether they have an admission or not. The students' lists do not change, but students can change their decision at each iteration if they are offered admission to a better college. The algorithm continues iterating until it reaches a stable matching solution.

3.5 The Proposed Algorithm

Knowing some similarities in the dataset is a great advantage to clustering algorithms. It can efficiently and effectively advance the outcome of the algorithm and create meaningful clusters. Accordingly, we developed a novel supervised clustering algorithm that is based on the CA algorithm (*66*). The proposed algorithm takes advantage of the natural labeling of the data (i.e., day of week, time of day) and models the clustering problem as a cooperative game. In this game, two disjointed sets of players join the game to identify a stable match. The first player's set consists of the centroids (clusters), and the second player's set consists of the data examples (data points). Each centroid orders the data points in its preference list based on the distance from the centroid to the data point. Alternatively, each data point orders the centroids in its preference list based on the purity. For example, a data point that has label h will give preference to the centroid that has the proportion of members with label h. In other words, a data point gives higher preference to centroids when the majority of its members have the same label as

its own label. Through a series of iterations, the proposed algorithm tries to match between the clusters, which want to minimize distances, and data points, which want to maximize purity, until it converges. It should be noted that cluster purity is the number of objects of the largest class in this cluster divided by the cardinality of the cluster, as presented in Eq. (1). The similarity measure is computed using Eq. (2). The algorithm terminates when the stopping criteria of Eq. (3) are met.

$$purity(c_i)^t = \max_m \left(\frac{n_i^m}{n_i}\right) \tag{1}$$

$$similarity(c_i)^t = \sum_{x_j \in c_i} \frac{1}{d(\boldsymbol{x}_j, \boldsymbol{c}_i)} \tag{2}$$

$$\alpha \left|\frac{\sum_{i=1}^{K} purity(c_i)^t - \sum_{i=1}^{K} purity(c_i)^{t-1}}{\sum_{i=1}^{K} purity(c_i)^{t-1}}\right| + (1-\alpha) \left|\frac{\sum_{i=1}^{K} similarity(c_i)^t - \sum_{i=1}^{K} similarity(c_i)^{t-1}}{\sum_{i=1}^{K} similarity(c_i)^{t-1}}\right| < \varepsilon \tag{3}$$

where t is the iteration number, n_i is the number of objects in cluster i (cardinality of cluster i), $i \in \{1, \dots, K\}$, n_i^m is the number of the class (m) in cluster i, $m \in \{1, \dots, M\}$, d is the distance between $\boldsymbol{x}_j$ and $\boldsymbol{c}_i$, $\boldsymbol{c}_i$ is the centroid of cluster i, $i \in \{1, \dots, K\}$, $j \in \{1, \dots, N\}$, N is the number of data points, $\boldsymbol{x}_j$ is the data vector j, α is a weighting factor (0.5 in our case), and ε is the stopping criteria threshold (0.0005 in our case).

We observe that one advantage of the proposed algorithm is that we do not need to write the entire objective function of the algorithm. Thus, we remove the normalization problem. However to stop the algorithm we normalize the purity difference by simply dividing by the previous purity and do the same with the similarity.

The following is a description of the proposed algorithm assuming the model order K is known:

1. Randomly choose K points as the initial centroids $\boldsymbol{c}_i$, $i \in \{1, \dots, K\}$.
2. Form K clusters by assigning all points to the closest centroid using $L1$ norm distance where $\boldsymbol{x}_j$ is assigned to the centroid that satisfies $\min_{\boldsymbol{c}_i} \left\|\boldsymbol{x}_j - \boldsymbol{c}_i\right\|_1$.
3. Recompute the centroid of each cluster by computing the median. The median is computed in each single dimension.
4. Find the cardinality of each cluster.
5. Compute the within-clusters class distribution matrix P.
6. $$P = \begin{bmatrix} \frac{n_1^1}{n_1} & \dots & \frac{n_1^M}{n_1} \\ \vdots & \ddots & \vdots \\ \frac{n_K^1}{n_K} & \dots & \frac{n_K^M}{n_K} \end{bmatrix}$$
7. Each centroid $\boldsymbol{c}_i$ creates its preference list of points $\boldsymbol{x}_j \; \forall \, j \in \{1, \dots, N\}$ based on $\left\|\boldsymbol{x}_j - \boldsymbol{c}_i\right\|_1 = \sum_{d=1}^{D} \left\|x_{dj} - c_{di}\right\|$, where D is the dimension of the data vector $\boldsymbol{x}_j$.
8. Each point creates its preference list based on the P matrix. For example a point from class m will create its preference list based on column m of the P matrix.
9. Find the best match using the CA algorithm.
10. Recompute the centroids and the P matrix based on the outcome of CA.
11. Evaluate the stopping criteria using Eq. 3.
12. While the stopping criteria are not satisfied, repeat steps 7–12.

To illustrate this algorithm, let us assume we have *N* data points and want to group them into three clusters as shown in Figure 3-1. The data points' labels are known. These labels could be any observed labels, such as the day of the week ($M = 7$). Moreover, we assume that the true number of clusters is three. The question we want to answer is how to partition the *N* data points such that similar data points in terms of distance and true labels are grouped together. By effectively partitioning the *N* data points, we can answer questions such as which days of the week have similar bike availability across the network.

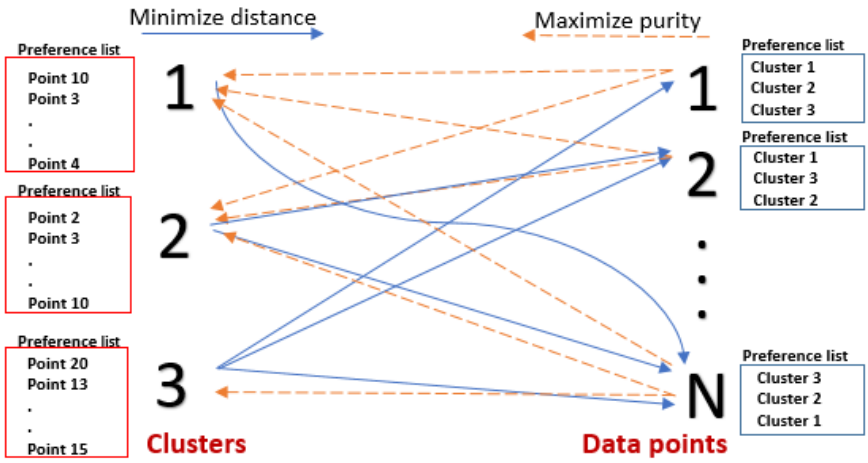


Figure 3-1 CA based clustering

In the first step, the proposed algorithm first randomly chooses three points as centroids for the three clusters. Then, it will partition the data points based on distance to get an estimate of the cardinality of each cluster and the P matrix. After that, each data point builds its preference list and each centroid builds its preference list, as shown in Figure 3-1.

In the second step, the proposed algorithm, through a series of iterations, will try to find matches between clusters and data points and provide a stable match using the CA algorithm. At the end of this step, all points should be matched with one of the three clusters.

After successfully matching the point with clusters, the centroid and P matrix of the three clusters will be recalculated. The algorithm will repeat the entire process of building new preference lists, matching, and calculating new centroids and the P matrix. The algorithm will stop when there is no significant improvement in the purity and similarity.

3.6 Datasets

We used docking station data collected from August 2013 to August 2015 in the San Francisco Bay area. The docking station data include station ID, number of bikes available, number of docks available, and time of recording. The time data included year, month, day of month, day of week, hour, and minute at which the docking station data were recorded. As the station data were documented every minute for 70 stations in San Francisco over 2 years, it was necessary to reduce the size of the dataset by sampling station data once at every quarter-hour instead of once at every 1 minute and obtaining the exact values without any smoothing process. This was done to reduce the complexity of the data and take a global view of bike availability in the entire network every 15 minutes, with the goal of finding the similarity between these views and clustering them based on this similarity and recorded time. Similarity refers to bike availability in all stations, while recorded time refers to day of week and hour of day. We discarded other time attributes such as year, day of month, and minute in the analysis as they might not have a significant impact on bike availability.

During the data processing phase, we found that numerous stations had recently been added to the network and others had been terminated, making it necessary to clean the dataset by eliminating any entries missing docking station data. This reduced the number of entries from approximately 70,000 to 48,000. Each entry included the availability of bikes at the 70 stations with the associated time (day of week and hour of day). The availability of bikes represents the coordination measure for each entry, which is used in the k-median method to determine the entry closeness measure. This resulted in each entry constituting 70 dimensions (70 stations).

3.7 Clustering Results and Discussion

In this section, we present the results of the aforementioned proposed algorithm using BSS station status dataset. We first demonstrate the technique used to select the model order, and then we show the results for each dataset with respect to month of year, day of week, and time of day.

3.7.1 Model Order Selection—Consensus Clustering (CC)

Finding clustering for similar days of the week or similar hours of the day is not straightforward, as we do not know the natural grouping for day of week or hour of day (i.e., number of clusters). In cluster analysis, determining the number of clusters is called model order selection. In this research effort, we used a well-known model order selection technique called consensus clustering to determine the number of clusters (*85*). This method looks for the model order that yields the most stable clustering solution. By stable clustering we mean that, given the model order, nearly the same paired data points are grouped together each time the clustering algorithm is run using different initial centroids (i.e., the centroids we begin the algorithm with) (*86*). The CC method begins by assuming that the number of clusters is K, and then the dataset is clustered B times (using different initial centroids). A consensus matrix (CM) which is an $N \times N$ matrix (N is the number of the data points), is built for this model order K. This matrix identifies the number of times each two data points are grouped in the same cluster divided by B. Then the algorithm increases K by one and redoes the clustering and the consensus matrix for the new model order. The algorithm continues doing this until it has scanned the whole range of model orders required. At this point, the best model order is chosen visually by drawing the cumulative distribution function (CDF) of the CM at each model order against the consensus index $c\_index \in [0,1]$ (Eq. 5). The CDF for a particular CM is defined over the range [0, 1] as follows:

$$CDF(c\_index) = \frac{\sum_{i<j} 1\{CM(i,j) \leq c\_index\}}{N(N-1)/2} \quad (5)$$

where $1\{\ldots\}$ denotes the indicator function, $CM(i,j)$ denotes entry (i, j) of the consensus matrix CM, and N is the number of rows (and columns) of CM.

The outcome of the CDF is that for the correct model order the elements of the CM will only have zeros and ones. So we estimate the CDF for different model orders and choose the cleanest CM which has the flatter CDF. In other words, every CDF curve represents a different model order (number of clusters), and the flatter the curve, the more stable the model order. To illustrate, in Figure 3-2, we give an example with regard to the time-of-day label. As the figure shows, the most stable model order for time of day was determined to be $K = 2$. Consequently, we analyzed the data in more detail for $K = 2$ in the following section. Similarly, the optimal number of clusters with regard to the day-of-week label is $K = 3$.

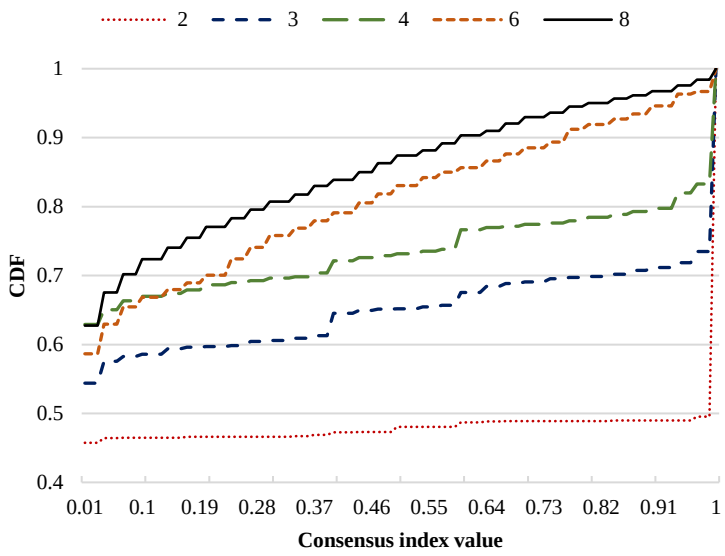


Figure 3-2 CDF against consensus index value for each cluster – time of day using BSS station status data

3.7.2 Results

First, we clustered the bike station data using the day-of-week label, and the optimal number of clusters found using the CC method was $K = 3$. The results of the three clusters are presented in Figure 3-3, which shows the probability of each day being in one of the three clusters. The three clusters are dominated by specific days: (1) Saturdays and Sundays, (2) Mondays and Fridays, and (3) finally Tuesday, Wednesday, and Thursday. This pattern differs from previous research (*14*) that showed bike patterns grouped into two clusters (weekend and weekdays). Our research shows that the weekdays can be split into groups: (a) Mondays and Fridays, and (b) Tuesdays, Wednesdays, and Thursdays. This appears to be logical, as the beginning and the end of the week are different from the rest of the weekdays.

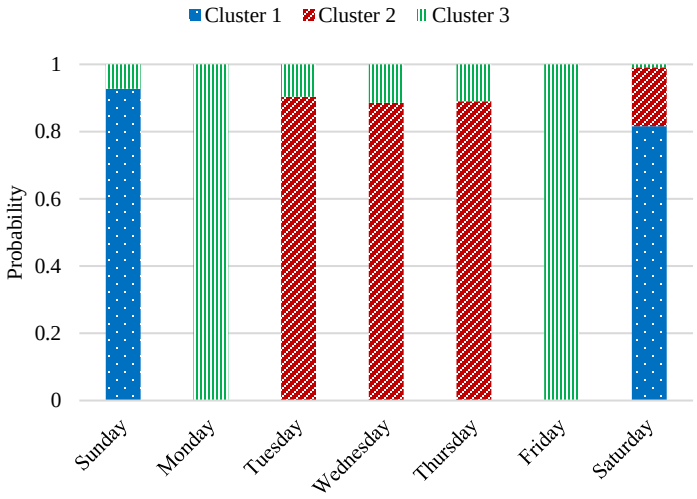


Figure 3-3 The probability of the day of week being in one of the three clusters (K = 3)

Each cluster is associated with a pattern for the availability of bikes at each station. The patterns of the ratio of the available bikes to the station capacity for the three clusters are provided in Figure 3-4.

Three observations can be made from Figure 3-4. First, the three patterns of the three clusters generally follow the pattern of the stations' capacity, which could be the result of system operators' rebalancing efforts. Second, the patterns of the three clusters show fluctuations in the bike activities; none of the days of the week has the highest activity for the entire network, which depends on both spatial and temporal factors. Third, several stations appear more likely to be empty or full on either weekdays or weekends. The difference in demand between the three clusters appears clearly for some stations, but not others. For example, the bike activities for cluster 1 (Tuesday, Wednesday, and Thursday) and cluster 3 (Saturday and Sunday) are similar for some stations in the network. That can be seen in stations 58 and 59 (San Francisco Caltrain 2–330 Townsend and San Francisco Caltrain–Townside at 4th). When taking a closer look at the location of these two stations, we found that they are located close to the Caltrain station. Accordingly, the similarity between these two clusters can be linked to the train timetable.

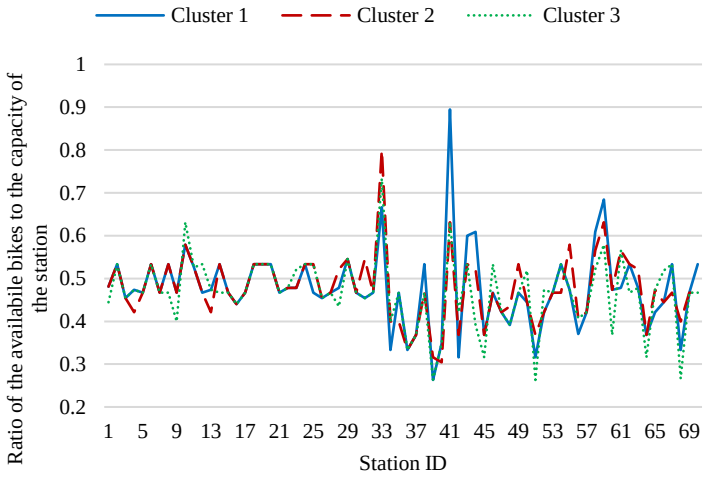


Figure 3-4 The ratio of the available bikes to station capacity for the three clusters at station in the network

Second, we clustered the bike sharing data using the hour- of-the-day label to find the hours of the day that have similar patterns. Only the station data at the beginning of each hour were considered. The optimal number of clusters was found to be two ($K = 2$). The analysis of the data reveals that the two clusters are peak (cluster 2) and non-peak (cluster 1) hours, confirming previous research. The results of the clustering are shown in Figure 3-5 and Figure 3-6, which give the probability of an hour being in one of the two clusters and the pattern of each cluster. It can be concluded from Figure 3-6 that when the patterns of the two clusters are lined up, the bike activity in the peak and non-peak hours is the same.

Generally, both clustering results for day of week and time of day are time homogeneous, making it possible for operators of the BSS to manage the bike stations and propose temporal and spatial plans. The clustering results give them a general view of the status of stations and clarify where the imbalances

would happen with respect to time of day and day of week, leading to better monitoring of the system as a whole.

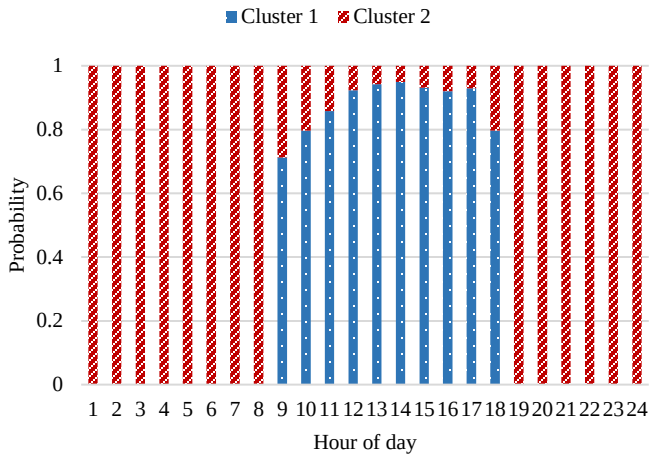


Figure 3-5 Probability of hour being in one of the two clusters (K = 2)

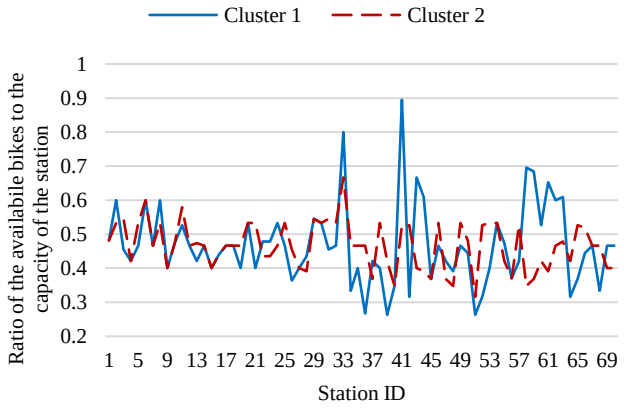


Figure 3-6 Available bikes of the two clusters for each station in the network

3.8 Conclusion

The chapter describes the development of a useful tool for agencies and researchers to cluster similar transportation patterns with respect to time-based events. A new supervised clustering algorithm was proposed to benefit from the background knowledge of the dataset along with similarity. Unlike other similar supervised clustering algorithms, the proposed algorithm is scalable given that it involves low computational times. It takes advantage of the natural labeling of the data (i.e., day of week, time of day) and models the clustering problem as a cooperative game and simultaneously clusters and identifies the stable number of clusters.

The algorithm was tested on BSS station status data from the San Francisco Bay area. Two types of background knowledge were used: day of week and hour of day. The proposed algorithm produced more meaningful clusters considering the background knowledge. The resultant clusters appear to be more time homogenous, giving the potential for operators to better manage the transportation modes per time event. Specifically, the algorithm provides insight for the clusters that operators can use to anticipate and

plan for imbalances in the BSS. We have shown that the proposed algorithm outperforms the classical k-means clustering algorithm, which did not reveal any obvious grouping of similar days.

CHAPTER 4 IDENTIFYING OPTIMUM BIKE STATION INITIAL CONDITIONS USING MARKOV CHAIN MODELING

This chapter is based on the paper listed below:
M. H. Almannaa, M. Elhenawy and H. A. Rakha, "Identifying Optimum Bike Station Initial Conditions using Markov Chain Modeling," in Transport Findings Journal.
DOI: 10.32866/6801

4.1 Introduction

Bike sharing systems (BSSs) are being deployed in many cities because of their environmental, social, and health benefits. To maintain low rental costs, rebalancing costs must be kept minimal. In this chapter, we use BSS data collected from the San Francisco Bay Area to build a Markov chain model for each bike station. The models are then used to simulate the BSS to determine the optimal station-specific initial number of bikes for a typical day to ensure that the probability of the station becoming empty or full is minimal and hence minimizing the rebalancing cost.

4.2 Research Question and Hypothesis

Bike Sharing systems (BSSs) suffer from a central recurring problem, namely: imbalance, meaning many bike stations either become empty or full during their daily operation. We hypothesize that we can reduce the cost of balancing the bike stations by optimizing the number of bikes at each station at the start of the day, thus reducing the need for a dynamic balancing system (*40, 42, 87*). We formulate our hypothesis by modeling each station using a Markov chain.

4.3 Methods and Data

This study uses Ford GoBike's BSS docking station data collected from August 2013 to August 2015 in the San Francisco Bay Area, as shown in Figure 4-1. The data provide the number of bikes at each station at one-minute intervals.

We used the discrete time-homogeneous Markov chain on a finite state space to model the system. We defined the state space as all the possible states a station could be in. Meaning, that if station s had N_s docks then the number of states for that station would be $N_s + 1$, where the "empty station" is counted as one possible state.

A matrix $X_{s,d,h}$ was constructed for each station, $s \in S$, day of the week, $d \in 7$, and hour of the day, $h \in 24$, (i.e. a total of S×7×24 X matrices were constructed of size (N_s+1)×(N_s+1)). Using a specific X matrix, the transition frequency matrix was created by computing the elements f_{ij}, where $i, j \in \{1, \dots, N_s + 1\}$. The elements f_{ij} represent the number of times a transition occurred from state *i* to state *j* over a 1-minute interval at a specific station, for a specific day of the week and within a specific hour of the day. The transition probability matrix for a specific station, *s*, hour of the day, *h*, and day of the week, *d*, was then computed as $p_{ij} = f_{ij} / \sum_{j=1}^{N_s+1} f_{ij}$. The calculated transition matrices above are the one-step transition matrices for a specific station, day of the week, and hour of the day. Each transition (i.e., the time tick) is conducted per minute, making the movement between states as smooth as possible throughout the hour.

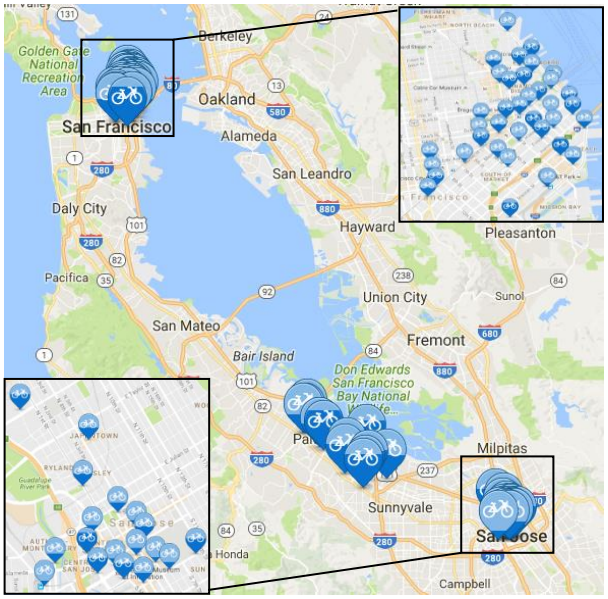


Figure 4-1 Locations of the 70 stations covering five cities: San Francisco, Palo Alto, Mountain View, Redwood City, and San Jose (88)

The probability distribution of the available bikes at a particular station at the end of the day is shown in Equation (1).

$$\mathrm{P}\left(\mathrm{x}^{\text{end of the day}} = \mathrm{q} \middle| \mathrm{x}^{\text{start of the day}} = \mathrm{m}\right) = \left(\mathrm{P}\left(\mathrm{x}^{\text{end of the first hour}} = \mathrm{q} \middle| \mathrm{x}^{\text{start of the first hour}} = \mathrm{m}\right)\right) * \prod_{\mathrm{h}=2}^{\text{last hour of the day}} \mathrm{P_h} \quad (1)$$

Here P_h is the 60-minute transition matrix obtained from simulating the corresponding one-step transition matrix.

Equation (1) finds the probability distribution of the available bikes at the end of the day given that the station started the day with *m* bikes. We count all possible paths from *m* at the very first hour of the day to all possible values of *m* at the end of the day. We use the corresponding transition matrix to simulate the Markov chains in order to produce a probability distribution that describes the likelihood of a particular state at the end of the hour. This leads to the creation of a probability distribution of available bikes at the end of the first hour. After that, we can use this probability distribution as the initial state probabilities for the following hour and create the next probability distribution, which is the next 60-step transition matrix. This procedure is repeated until we reach our target hour and draw the final probability distribution as a function of each initial condition.

When running the Markov Chain, our objective function was to find the best initial conditions that maximize the probability of the station operating at a bike-to-capacity ratio (number of bikes relative to the capacity of the station) within the range 0.25 to 0.75 at the end of each hour, as shown in Equation (2).

$$max_i \ \sum_{h=1}^{24} W_h \sum_{j=N_{min}}^{N_{max}} P_{ijh} \quad (2)$$

Where i is the initial condition of station *s*, h is the hour of the day (considered only the hours from 6:00 a.m. to 8:00 p.m. in our case), W_h is the weight assigned to hour h (assumed to be 1.0), j is the expected state of the station at the end of the hour, N_{min} and N_{max} are the upper and lower desired bounds of the station status (in our case: N_{min}=0.25×(N_s + 1) and N_{max}=0.75×(N_s+1), N_s

is the capacity of station s, and P_{ijh} is the probability of having an i initial state and a resulting j state at the end of hour h.

4.4 Findings

We used the BSS data to build the Markov chain for each station and day of the week combination to investigate the daily imbalances and identify the optimal inventory level that minimizes the probability of a station reaching an empty or full state. When analyzing the results, we first looked at all 70 stations, considering different initial conditions to identify the stations that would benefit most from optimizing the initial station state. We grouped stations into three categories: (1) have an imbalance issue but with a small probability (≤10%) for 25% of the initial conditions, (2) have an imbalance issue with a medium probability (11–25%) for 25 to 45% of initial conditions, (3) have an imbalance issue with a large probability (>25%) for >45% of the initial conditions. In Table 4-1, we present the percentage for each category for each city separately, as a previous study showed that there were close to no trips between the five cities (*15*).

Table 4-1 Percentage of stations in categories 1 through 3 for all five cities

City	Category		
	(1) Imbalance probability of ≤10% for 25% of initial conditions	***(2) Imbalance probability of 11–25% for 25 to 45% of initial conditions***	***(3) Imbalance probability >25% for >45% of the initial conditions***
San Jose	43.75	12.50	43.75
Redwood City	57.14	28.57	14.29
Mountain View	14.29	57.14	28.57
Palo Alto	80.00	20.00	0.00
San Francisco	0.00	20.00	80.00

As shown in Table 4-1, San Francisco has the highest percentage of category 3 stations, followed by San Jose. This demonstrates that San Francisco BSSs experience high bike demands, and thus are more likely to have an imbalance problem during the day. Our proposed approach would be less effective for the San Francisco BSS and more effective for the other cities given that the daily evolution of states for San Francisco varies considerably.

Our analysis shows that the optimal initial conditions vary from one day of the week to another for the same station, and thus we present the optimal initial conditions for each day of the week for only two selected stations, one in Mountain View and one in San Francisco. Note that we made two assumptions when choosing the optimal initial conditions: (1) the bikes are taken from an infinite pool, meaning we have no constraints on the available inventory (2) there is no interaction between stations. The optimal station state is assumed to occur when the bike-to-capacity ratio ranges between 0.25 and 0.75 over the entire day, thus minimizing the probability of reaching either an empty or full state. Table 4-2 presents the optimum three initial states for stations 26 and 59 that result in the highest probability of maintaining a bike-to-capacity ratio ranging between 0.25 and 0.75 for the entire day. As was demonstrated earlier, the results of Table 4-2 demonstrate that there is a lower probability of being able to maintain the San Francisco station in the optimum range over the entire day, as was discussed earlier.

Table 4-2 The optimal initial conditions for stations 26 and 59 (optimum number of initial bikes and probability of achieving the desired bike-to-capacity ratio)

	Station#26 Mountain View			Station#59 San Francisco		
	1st	2nd	3rd	1st	2nd	3rd
Saturday	6 (0.74)	5 (0.74)	7 (0.74)	7 (0.65)	8 (0.63)	9 0.63
Sunday	6 (0.74)	5 (0.74)	4 (0.73)	7 (0.62)	8 (0.62)	9 (0.62)
Monday	4 (0.70)	5 (0.69)	3 (0.69)	8 (0.42)	9 (0.42)	10 (0.41)
Tuesday	4 (0.71)	3 (0.70)	5 (0.70)	7 (0.42)	8 (0.41)	9 (0.41)
Wednesday	5 (0.71)	4 (0.71)	6 (0.69)	7 (0.38)	9 (0.37)	9 (0.37)
Thursday	4 (0.70)	5 (0.70)	6 (0.68)	7 (0.42)	8 (0.41)	9 (0.41)
Friday	5 (0.71)	4 (0.70)	6 (0.69)	7 (0.42)	8 (0.41)	10 (0.41)

CHAPTER 5 BIKE COUNT PREDICTION

This chapter is based on the papers listed below:

1. *M. H. Almannaa, M. Elhenawy and H. A. Rakha, "Dynamic Linear Models to Predict Bike Availability in a Bike Sharing System," in International Journal of Sustainable Transportation.*
2. *M. H. Almannaa, M. Elhenawy, F. Guo, and H. A. Rakha. (2018), "Incremental Learning Models of Bike Counts at Bike Sharing Systems," 21st IEEE International Conference on Intelligent Transportation Systems (IEEE ITSC 2018), Maui, Hawaii, November 4-7.*

In this chapter, we provide a toold box of prediction models that can be used for BSSs. Statistical and machine learning models were adapted and compared in terms of prediction accuracy and computational time as follows:

5.1 Dynamic Linear Models to Predict Bike Availability in a Bike Sharing System

5.1.1 Introduction

With rapid worldwide population growth, large, dense cities are struggling with traffic congestion. Many people have migrated from rural to urban areas, creating highly crowded cities with limited resources. Traffic jams are one of the critical issues that urbanized areas suffer from. A number of potential solutions have been proposed to mitigate the negative impact of this phenomenon and improve private and public transportation. One cost-effective solution is a bike sharing system (BSS), where residents and visitors to urban areas can ride from one bike station to another for a very low rental fee, making the system accessible to many people.

The BSS idea started over five decades ago in Europe, and then bloomed in 50 countries, growing to more than 37,000 stations by 2000 (*1*). This growth testifies to the significant transportation benefits that can be obtained by implementing a BSS, which are further enhanced with the use of advanced technology. For example, bike riders can borrow a bike from any bike-sharing station using a smart card and then return it to a bike station near their destination. Many BSSs offer an app for bikers that provides necessary information such as nearby bike stations, bike dock availability, and operation hours. More broadly, BSSs provide a sustainable transportation mode, especially with last-mile trips, and help to reduce congestion, emissions, and pollution. Some BSSs have successfully linked public transportation modes by filling the gaps between them and thus making it possible for residents and visitors of the city to access the restricted traffic zones that give a priority for pedestrian and cyclists over cars.

The significant increase in the use of BSSs raises the issue of imbalance in the distribution of bikes. Some stations are at capacity and others are empty. This issue creates logistical challenges for BSS operators and may discourage bike riders, who could find it difficult to pick up or drop off a bike. To address the problem, recent research has been conducted on rebalancing the distribution of bikes at stations (*5, 6, 29*). There are three major ways to address the rebalancing issue: static, dynamic and incentivized. The incentivized approaches make use of the users in the balancing efforts, in which the operating company incentivises them to change their destination in favor of keeping the system balanced. Static approaches neglect the demand during the rebalancing time because they are usually conducted when bike activities are at their lowest: at midnight. The dynamic approaches are more complicated as they take into account the movement of bikes during the rebalancing efforts, so they can be done any time during the day. Thus, a key task of dynamic rebalancing efforts is to accurately predict bike counts at any station in the BSS (Figure **5-1**).This could help both bikers and operating agencies plan ahead and act accordingly. For instance, bikers could change their origin or destination in advance if they knew that the station would be either

empty or full respectively by the time they arrive, which will eventually keep the BSS balanced without a need for relocating the bikes. Operating agencies could use the predicted demand when rebalancing to prevent any station from running out of bikes or being too full of bikes.

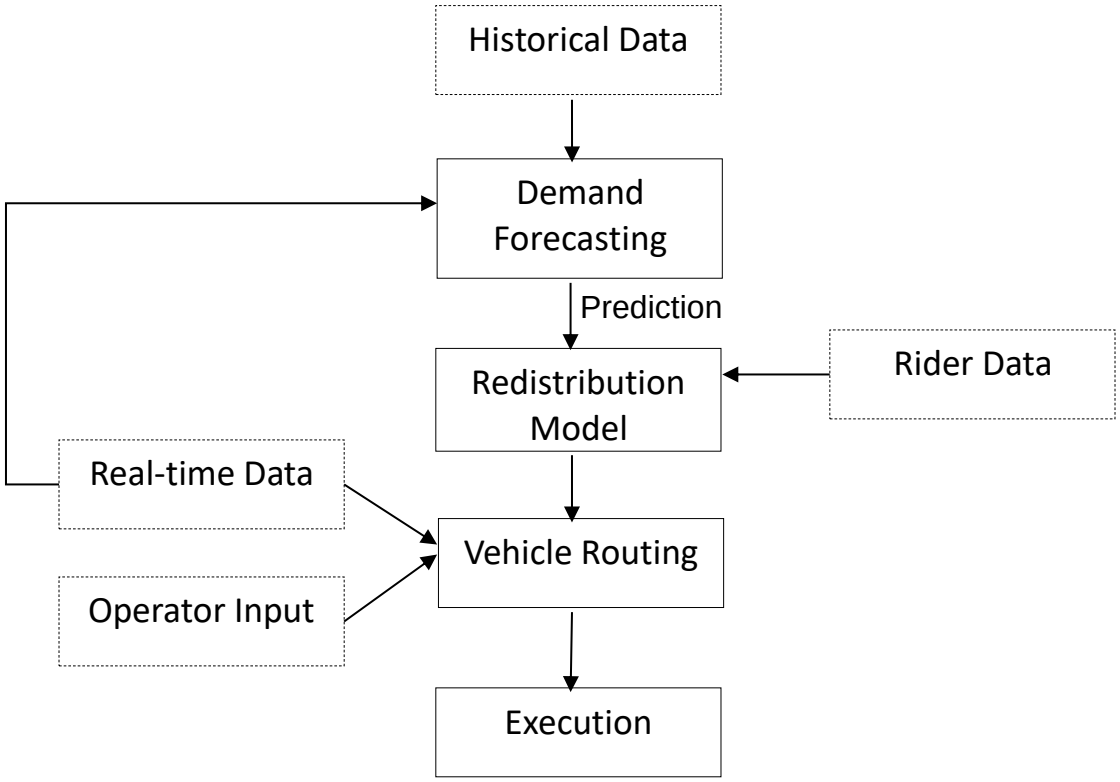


Figure 5-1 Model interactions (*10*)

Many researchers use a statistical model to predict the demand at any given station, while others use clustering algorithms, such as traditional and non-traditional clustering (*20, 89*). A crucial part of the prediction process is quantifying the effect of weather conditions and other factors on the bike count at stations. Consequently, extensive research efforts have been conducted using statistical and machine learning approaches to determine the correlation between bike availability and other factors and thus the significant factors involved (*12, 21, 22*).

In (*90*), we developed a bike count model to quantify the effect of weather conditions on the prediction of bike counts at stations using Poisson and negative binomial regression models. Random forest (RF) and step-wise regression were used, and the results show that mean temperature, mean humidity, mean visibility, mean wind speed, precipitation, and events in a day (rainy, foggy, or sunny) are significant factors. In (*24*), we used machine learning algorithms—RF, least squares boosting (LSBoost), and partial least squares regression (PLSR)—to model the number of available bikes at each station in the BSS. The input variables for these models for each station include the six weather variables mentioned above, the month, day of the week, and time of day. Univariate and multivariate regression algorithms were introduced and compared, and the results demonstrate that univariate models have lower error predictions than the multivariate model. Similarly, Wang and Kim adapted two other machine learning algorithms: long short-term memory neural networks (LSTM) and gated recurrent unit (GRU) to predict bike counts. Although their results in general show that both LSTM and GRU algorithms have similar prediction accuracy, GRU outperforms slightly the LSTM in terms of both accuracy and computational time

(*91*). Chengcheng et al. adapted the long short-term memory neural networks (LSTM NN) for predicating bike prediction and attraction at traffic analysis zones (*92*). The results show the LSTM NN shows good prediction accuracy at different prediction time intervals.

Although the previous approaches show promising results in predicting the bike counts at stations, they suffer from three major drawbacks. First, they fail to capture the dynamic changes over time, making an inaccurate assumption that users' activity will remain the same in the future and neglecting the changes that dynamic cities or new technology may bring. Consequently, these models produce constant coefficients and/or static decision rules that do not evolve with time. These models do not take into account the continuing efforts of BSS operators to keep the system balanced. For example, modern BSSs have adopted an app that can alter bikers' behavior based on the status of nearby stations. BSS operators attempt to incentivize bikers to change their origin or destination in favor of keeping the system balanced (*7*, *23*). The second drawback of the existing machine learning approaches mentioned above is they are sophisticated models using too many variables (19 variables in (*90*)) and some of them are difficult to interpret. The third drawback is that they work poorly when encountering missing data, so the algorithms must rely on some sophisticated imputation techniques such as Autoclass and C4.5 (*25*). BSS data, as data in any dataset, suffer from missing data due to malfunctions or measurement error in data collection. Additionally, it is very common that some bike stations drop out of service due to rebalancing efforts or technical issues, creating a missing data problem.

Dynamic linear models (DLMs) have gained attention due to their flexibility and ability to capture underlying changes over time, offering a powerful tool for many applications in different fields (*93*). Unlike other statistical and machine learning algorithms and models, DLM estimation and forecasting can be done recursively without a need to store the entire past history. Given the available information, they adapt themselves in a very short time as new data arrive, outperforming many advanced algorithms. In addition, DLM inference and prediction can efficiently handle the missing data problem.

The goal of this research effort is to develop a simple DLM to predict bike counts at stations in BSSs. Two DLMs are adopted: first- and second-order polynomial DLMs. Unlike other machine learning models, these two models do not use any predictors (i.e., no weather or time information) but the log of the bike count at the station being modeled. We tested the DLMs at different prediction windows: 15, 30, 45, 60, and 120 minutes. The first three short prediction windows (15, 30, and 45 minutes) were mainly tested to forecast station status for bikers. The longer prediction windows (60 and 120 minutes) are for operating agencies to rebalance the system.

The chapter is organized as follows. First, a brief summary of the related work and methodology are provided, followed by a short description of the dataset. Second, the experimental work with the obtained results are given. Before the conclusions of the chapter are drawn, a comparison with other machine learning algorithms is presented.

5.1.2 Related Work

Regression count modeling is one of the common approaches recently used to model bike counts. In Austria, Rudloff and Lackner proposed a demand model for bikes and return boxes (*43*). Poisson, negative binomial (NB), and Hurdle models were used to model the bike counts within a given hour. The weather information (in particular, temperature and precipitation) and neighboring stations were used as regressors in these three models, and the results showed that the Hurdle model outperforms the other two approaches. Wang et al. used log-linear and NB regression

models to anticipate bike availability with 13 independent variables such as socioeconomic, demographic, and geographic factors (*12*). The results showed that all 13 variables are significant with a high goodness of fit for both models. R. Rixey used a multivariate linear regression analysis to find the significant factors for the bike sharing ridership, and, then, estimate system ridership (*11*). The study found that demographics, the built environment, and access to a comprehensive network of stations were significant factors in the multivariate linear regression model.

Given the size and complexity of the BSS data, clustering analysis and visualization techniques have been discussed extensively. Various researchers have attempted to derive insights by exploring trends through visualization techniques (*13, 45-47*). For example, Froehlich et al. studied BSS patterns using 13 weeks of bicycle station usage data from Barcelona. They investigated the relationship between human behavior, geography, and time of day, and then tried to predict future bike station usage. The temporal and spatiotemporal patterns were discussed, and the results showed that there were some dependencies among the stations. The available bicycling data were used to cluster the docking stations. Neighboring stations were found to be highly correlated and therefore were clustered in one group. Kaltenbrunner et al. also attempted to improve the BSS in Barcelona using docking station data (*14*). Temporal and geographic mobility patterns were obtained and analyzed with the goal of detecting imbalances in the BSS. Subsequently, they used time series analysis techniques to predict the number of bicycles at a given station and time. Vogel et al. attempted to derive bike activity patterns by analyzing bike share data along with geographical data (*48*). Cluster analysis was used to group the bike stations with respect to pick-up and return activity. They used k-means, expectation maximization, and sequential information-bottleneck algorithms to conduct their analysis. Using the temporal activities of the stations, their results showed that the bike stations could be clustered into five groups, and, thereby, average pickup and return for each hour were given for each group. After that, the authors tried to link these five clusters with geographical information data and found that stations in the same cluster tend to be neighbors. Feng and Hillston et al. developed a novel moment-based prediction model using time-dependent rates. They used a Population Continuous Time Markov Chain (PCTMC) to derive the number of available bikes (*94*). Gast and Massonnet et al. used a queuing theoretical time-homogeneous model of BSSs to make probabilistic forecast (*95*). They also introduced a new metric to evaluate the proposed model instead of the standard root-mean-square error (RMSE). Fricker and Gast adapted a stochastic model and a fluid approximation to investigate the influence of the station capacities on the performance of homogeneous BSSs (*32*).Their proposed model helps in determining the optimal size of each station in terms of minimizing the imbalance.

A few recent studies adopted time series techniques to predict the bike counts at stations (*14, 17, 50, 51*). Although these techniques showed good performance in both explaining the past and predicting the future, they had several limitations. For example, Kaltenbrunner et al. (*14*) used an autoregressive moving average (ARMA) model to predict the bike availability at stations (*14*). However, the ARMA model is a stationary model that assumes the mean and variance of the observations are fixed over time, which is not the case in the bike station data. Froehlich et al. proposed four models: last value, historic mean, historic trend, and Bayesian network (*13*). They showed that the Bayesian network model produces the least prediction error. Yet, the Bayesian network model was not adopted to give exact bike counts. Instead, it provided only a small number of prediction classes (in percentages); that is, the bike availability in stations was classified in even percentage intervals (for example, 25%, 50%, 75%, and 100%), and the algorithm only chose one of the four categories to describe the bike availability.

Yoon et al. proposed a spatial-temporal prediction system using an autoregressive moving integral average (ARIMA) model to overcome the non-stationary issue in the ARMA model (*17*). Seasonal trends and neighboring information were utilized in the model. A small dataset of 3 weeks was used to evaluate the model. Their results show a slight improvement in favor of ARIMA when compared to ARMA (The error is 3.47 bikes/station versus 3.50 bikes/station). However, ARIMA is considered a static model; its estimated coefficients do not evolve with time and predictions are only within even intervals. Additionally, ARIMA is a complex and hard-to-interpret model.

5.1.3 Methodology

We used DLMs to model the bike counts at stations because of their ability to evolve and capture the change in users' behavior over time (*96*). The DLM is a special case of a general state space model as it is linear and Gaussian. Being linear makes it possible to extend the model and add trends, covariate, seasonality, and autoregressive components.

DLMs are based on the idea of describing the output of a dynamic system—for example, the bike count series of a bike station—as a function of a non-observable state process (which has a simple, Markovian dynamic) affected by random errors. Given that it is a dynamic model, the coefficients of the model are estimated at every Δt.

In general, the dynamic system which generates the observed station status (bike counts) can be written in the general state space model form. Therefore, it can be specified by:

1. The observation equation, $S_t = h_t(\theta_t, \nu_t)$, where ν_t is the observation error.
2. The evolution equation, $\theta_t = g_t(\theta_{t-1}, \omega_t)$, which captures the model dynamics, where ω_t is the innovation.
3. The prior distribution for the initial state, θ_0.

In the DLM, h_t and g_t are linear functions. Moreover, we assume Gaussian distributions such that any joint distribution of the states and observation will be Gaussian and we only need to estimate its mean and covariance matrix. Therefore, the bike count dynamic system can be fully specified by the following equations:

The observation equation: $Y_t = F_t\theta_t + \nu_t$, (1)

where $\nu_t \sim N(0, V_t)$ and F_t is a known matrix; and

the evolution equation: $\theta_t = G_t\theta_{t-1} + \omega_t$, (2)

where $\omega_t \sim N(0, W_t)$ and G_t is a known matrix and $\theta_0 \sim N(m_0, C_0)$ is the initial state.

Once we define the state space model for the bike station, it can be used to make inferences on the unobserved states and predict future observations using part of the observation sequence. In a DLM, the Kalman filter is used for updating our current inference on the state as new data become available. DLM computations can be done recursively and there is no need to store the entire past history. In addition, DLM inference and prediction can efficiently handle the missing data problem.

The DLM can be written in different ways based on the assumptions and information added to it (i.e., trends, seasonality, and regressors). In this chapter, we only use two simple models: first-and second-order polynomial models. The following two subsections cover them briefly.

5.1.3.1 First-Order Polynomial Model (Random Walk Plus Noise Model)

The first-order polynomial model is also called a random walk plus noise, or local level, model (*96*). It is the simplest DLM model that assumes a constant mean (i.e. a zero slope). It is similar to the first-order Taylor series approximation of a smooth function. The first-order model is used mainly for time series observations with no clear seasonal or trend variations. The observations (Y_t) are modeled as noise observations with a mean of μ_t. The mean μ_t changes over time as a function of μ_{t-1} and w_t, which leads the mean to be non-stationary. Given that it is a first-order model, F_t and G_t are equal to one. The first-order polynomial model can be formulated using the following two equations:

$$Y_t = \mu_t + v_t \qquad v_t \sim N(0, \sigma_v) \tag{3}$$

$$\mu_t = \mu_{t-1} + w_t \qquad w_t \sim N(0, \sigma_w) \tag{4}$$

where Y_t is the observation at *t*, μ_t is the state governing the mean of the observations at *t*, and v_t and w_t are independent random errors with zero mean and a variance of σ_v and σ_w, respectively. In this chapter, we assume v and w are time-invariant for the sake of simplicity.

5.1.3.2 Second-order polynomial model (linear growth model/local linear trend model)

This model is very similar to the first-order model with only one key difference, namely: it considers both the mean and slope of the observations. Unlike the first-order model, it includes a time-varying slope (denoted by B_t) in the evolution equation, representing the growth of the level of the observations. The second-order polynomial model can be defined as follows:

$$Y_t = F_t\mu_t + v_t \qquad v_t \sim \mathrm{N}(0, \sigma_{v_t}) \tag{5}$$

$$\mu_t = G_t\mu_{t-1} + B_{t-1} + w_{1,t} \qquad w_{1,t} \sim \mathrm{N}(0, \sigma_{w_{1,t}}) \tag{6}$$

$$B_t = B_{t-1} + w_{2,t} \qquad w_{2,t} \sim \mathrm{N}(0, \sigma_{w_{2,t}}) \tag{7}$$

where Y_t is the observation at *t*, μ_t is the state governing the mean of the observations at *t*, and B_t is the state governing the slope of the observations at *t*. v_t, $w_{1,t}$, and $w_{2,t}$ are independent random errors with zero mean and a variance of σ_{v_t}, $\sigma_{w_{1,t}}$, and $\sigma_{w_{2,t}}$, respectively.

The above model can be written as follows:

$$Y_t = F_t\,\theta_t + v \qquad v_t \sim \mathrm{N}(0, \sigma_{v_t}) \tag{8}$$

$$M_t = G_t\,\theta_{t-1} + w \qquad w \sim \mathrm{N}\left(0, \begin{pmatrix}\sigma_{w_1} & 0\\ 0 & \sigma_{w_2}\end{pmatrix}\right) \tag{9}$$

where $\theta_t = \begin{pmatrix}\mu_t\\ B_t\end{pmatrix}$, $F_t = (1,0)$, $G_t = \begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}$, and $w = \begin{pmatrix}w_1\\ w_2\end{pmatrix}$.

5.1.4 Dataset

This study used a publicly available dataset of docking station data. The case study dataset covers the period from September 1, 2014, to September 1, 2015, in the San Francisco Bay area for 70 stations in five different zip codes as shown in Figure 5-2. The dataset includes station ID, number of bikes available, number of docks available, and time of recording. Each row has the availability of bikes at the 70 stations with the associated time (day of week and hour). As the station data were collected at a frequency of every minute for 70 stations in San Francisco over a year of 2014–

2015, the dataset contains a large amount of recorded station data. Consequently, we derived five subsets of the original dataset by sampling station data once at 15, 30, 45, 60, and 120 minutes and obtaining the exact values without any smoothing process. This was done to reduce the complexity of the data and avoid running out of memory. Additionally, we could build a model for each different version of the dataset and do one-step-ahead forecasting to get the prediction up to 120 minutes.

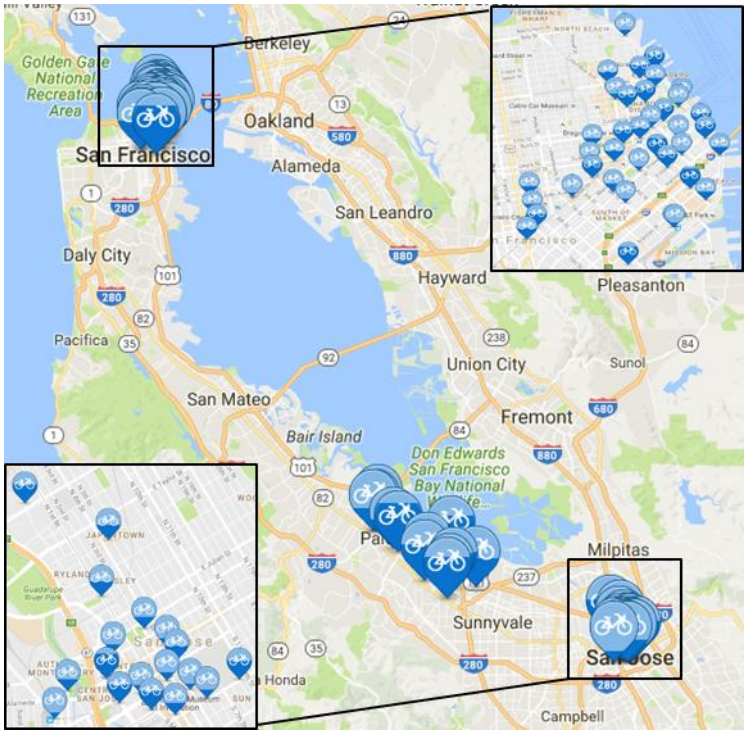


Figure 5-2 Locations of bike stations in San Francisco Bay area (97)

5.1.5 Model Testing

The first- and second-order polynomial models were coded using R (*96*). These two models were applied to create univariate models for 70 stations in the San Francisco Bay area. The response of the models (denoted by Y_i) is the log of the number of predicted available bikes at station i. Different prediction windows were used: 15, 30, 45, 60, and 120 minutes. The first three short prediction windows (15, 30, and 45 minutes) were mainly tested to forecast station status for bikers while the longer prediction windows (60 and 120 minutes) are for operating agencies to rebalance the system.. All year-round data were utilized, including weekends and weekdays, on-peak and off-peak hours, and summer and non-summer months. The DLMs returned the anticipated log of the number of bikes at every prediction window. To ensure the prediction did not exceed the size of the bike station, we set the prediction equal to the maximum capacity if the prediction was larger than the station's capacity.

We used two different approaches when applying the DLMs for prediction windows longer than 15 minutes: (1) we modeled the sample at an interval equal to the prediction horizon and did one-step-ahead forecasting and (2) we modeled the 15-minute sampled dataset and used multiple-steps-ahead forecasting techniques. In the following subsections, we present the evaluation criteria used with the results for each approach, followed by a comparison of these two approaches with other machine learning algorithms.

5.1.6 Evaluation Criteria

To measure the predictive accuracy of the two models, two different measurements were used: the mean absolute forecast error (MAE) and the symmetric mean absolute percentage error (SMAPE). The MAE (well-known as prediction error) was calculated by taking the average of the absolute difference between the anticipated and actual number of the bike counts for all 70 stations in the entire year (Equation 8). The SMAPE is an accuracy measure and is calculated as shown in Equation 9.

$$\text{MAE} = \frac{\sum_{i=1}^{n} |Y_t - A_t|}{n} \tag{8}$$

$$\text{SMAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \frac{|Y_t - A_t|}{(|A_t| + |Y_t|)/2} \tag{9}$$

where n is the number of observations, and Y_t and A_t are the predicted and actual number of bike counts respectively.

The third measurement that was used is the MAE relevant to the capacity of the station (MAE/C) in which we divide the MAE for each station over its capacity. This is to make the prediction error more informative by considering the capacity of stations.

5.1.7 DLM Using Single-Step-Ahead Forecasting Technique

This approach used a one-step forecasting technique, meaning we built a model of each version of the dataset. For instance, when applying the DLMs for a prediction window of 120 minutes, we used the reduced (sampled) dataset at 120 minutes, then did the forecast one step ahead using the observation and evolution equations mentioned above for both first- and second-order models.

5.1.8 DLM Using Multiple-Steps-Ahead Forecasting Technique

This approach built only one model using the 15-minute sampled data and did the forecast using a multiple-steps-ahead forecasting technique. If we want to estimate the bike counts at time $t + k$ (k is sometime in the future) and the available data are only up to time t, the multiple step forecasting of the bike count can be estimated as following:

For the first-order model, we need to know only the mean of the observations at time t (the level for the observation), so the estimated bike count at time $t + k$ can be determined as follows:

$$\mu_t = E\left(\frac{Y_{t+k}}{Y_{1:t}}\right)$$

where μ_t is the known status space at time *t*, and $y_{1:t}$ is the observed data from time 1 until time t.

For the second-order model, we need to know two parameters: the mean μ_t and slope B_t of the observations at time *t*, and then estimate the bike counts as follows:

$$E\left(\frac{Y_{t+k}}{Y_{1:t}}\right) = \mu_t + K \times B_t$$

5.1.9 Results

Table 5-1 shows the performance comparison of the two DLMs considering five different prediction windows and two approaches. It was unsurprising that the first and second-order models for both approaches had quite similar results (up to the ten-thousandths place) over different

prediction windows. That can be explained by the fact that our bike count data do not show any clear trend, so there is no benefit of adding a term for the slope (i.e., using the second-order polynomial model). The DLMs clearly show a high accuracy in predicting the bike counts, especially at a short prediction window for both the single- and multiple-step approaches. The DLMs were able to predict the bike count precisely at a 15-minute window with a small prediction error of 0.37 bikes/station (2% with respect to the station capacity), corresponding to a percentage error (SMAPE) of 5%. The DLM using a multiple-step approach outperforms the single-step approach under all the prediction windows. The difference between these two approaches increases as the prediction window increases, with the 120-minute prediction window having the biggest difference: 0.6 bikes/station (9.3% with respect to the station capacity).

Table 5-1 Performance comparison of the two DLMs at different prediction windows, using one-step-ahead and multiple-steps-ahead forecast techniques

Prediction window (minutes)	**First-order model, single step**			**Second-order model, single step**			**First-order model, multiple step**			**Second-order model, multiple step**		
	MAE	**SMAPE**	$\frac{MAE}{C}$	**MAE**	**SMAPE**	$\frac{MAE}{C}$	**MAE**	**SMAPE**	$\frac{MAE}{C}$	**MAE**	**SMAPE**	$\frac{MAE}{C}$
15	0.37	0.05	2.07	0.37	0.05	2.07	0.37	0.05	2.09	0.37	0.05	2.09
30	0.65	0.08	3.59	0.65	0.08	3.59	0.52	0.07	2.89	0.52	0.07	2.89
45	0.90	0.11	4.99	0.90	0.11	4.99	0.65	0.08	3.58	0.65	0.08	3.58
60	1.13	0.13	6.22	1.13	0.13	6.22	0.76	0.09	4.19	0.76	0.09	4.19
120	1.70	0.18	9.32	1.70	0.17	9.32	1.10	0.12	6.06	1.10	0.12	6.06
Average	0.95	0.11	5.24	0.95	0.11	5.24	0.68	0.08	3.76	0.68	0.08	3.76

Given that the first- and second-order DLMs yield almost the same results, we will discuss only the first-order DLM in the rest of the chapter for both the singe-step and multiple-step approaches. In Figure 5-3, we present the pattern of the prediction error in percentages (MAE/C) of the first-order DLMs of the single-and multiple-step approach over different prediction windows. The figure clearly shows that the prediction error increases as the prediction window increases for both approaches, with the 120-minute window being the least-effective prediction window and producing a prediction error of 6% and 9.3% bikes/station for the single- and multiple-step approaches, respectively.

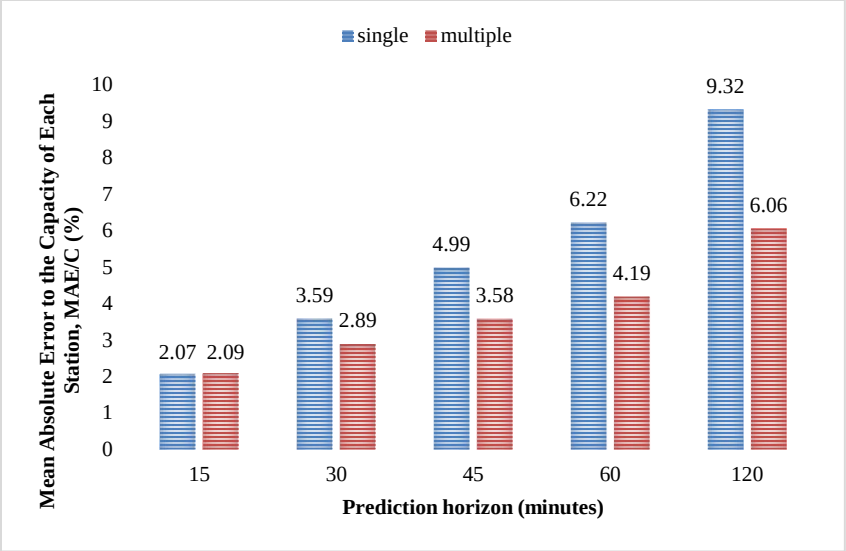


Figure 5-3 Prediction error with respect to the capacity for the single- and multiple-step approaches for the first-order DLM for different prediction horizons

The MAE/C of the single- and multiple-step approaches at different prediction windows helps to explain why the multiple-step approach outperforms the single step (Figure 5-4). Generally, the behavior (pattern) of the two approaches is similar at each prediction window. Surprisingly, the patterns of both approaches are lined up at stations 1–32, and then a gap favoring the multiple-step approach begins (i.e., the prediction error decreases in favor of the multiple-step approach). This gap becomes larger at the 60-minute and 120-minute prediction windows.

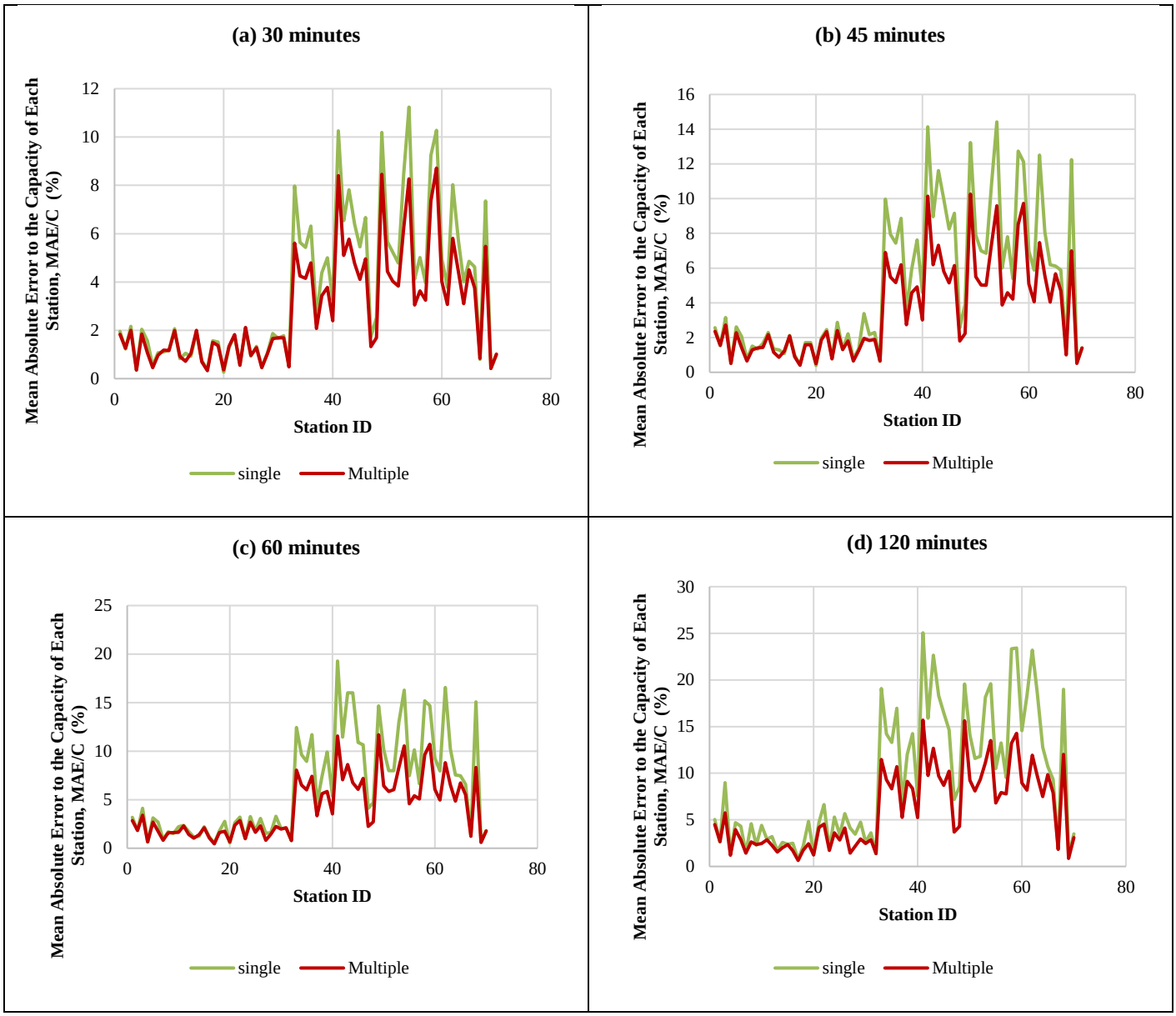


Figure 5-4 MAE/C per station for single-step and multiple-step forecasts at different prediction windows

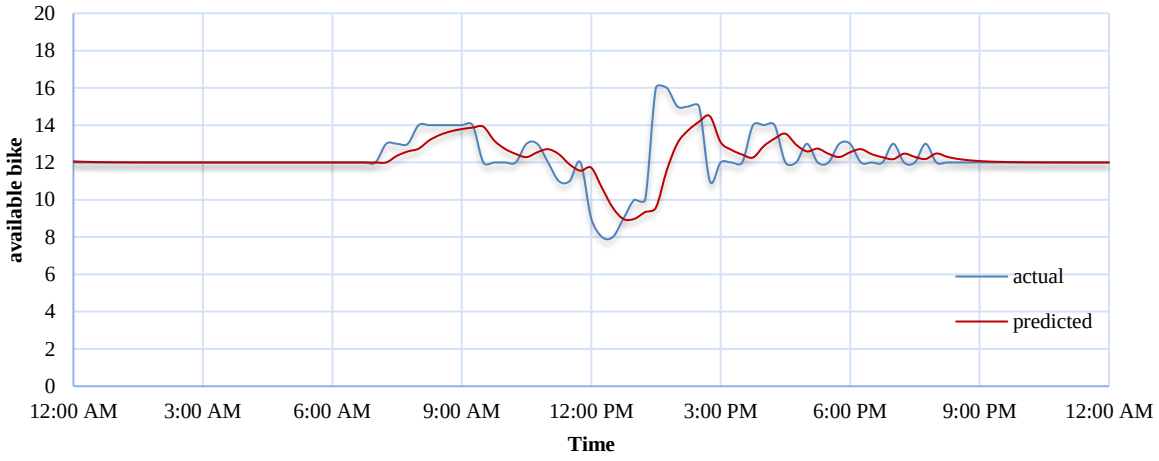
Investigating the difference between stations 1–32 and the other stations, we found that the usage patterns for stations 1–32 tends to be smoother than the patterns of the other stations. Stations 33–70 are more likely to become empty (or get full) in a short period. We investigated the spatial characteristics of the stations and found that, for the most part, stations 33–70 are located in downtown San Francisco and thus are exposed to high demand, unlike the other stations, which are located in four smaller cities (San Mateo, Mountain View, Palo Alto, and San Jose). In fact, we recently learned that the operating company (Ford GoBike) has deactivated most of these stations and we believe this could be due to the low demand.

With regard to performance, the single-step approach performs well when encountering a smooth pattern in which the bike activity changes slightly, but it fails with irregular patterns (i.e., sudden

changes in bike activity within a short period). The multiple-step approach is always powerful when facing either a smooth or uneven pattern. This can be explained as follows. The DLM using a single-step approach uses sampled data at a larger interval, hence less information is used to build the model. In addition, because the multiple-step approach uses 15-minute sampled data, it is updated more frequently than the single-step approach.

For the sake of completeness, in Figure 5-5**,** we give three examples for the prediction obtained using the first-order DLM at the 15-minute prediction window for three different days: Saturday, Sunday, and Friday. We chose these days with the goal of demonstrating the performance of the adopted first-order DLM for three common patterns: (a) low-demand, (b) medium-demand, and (c) high-demand stations. The blue curve corresponds to the actual number of bikes in the station, while the red curves indicates the predicted number. In Figure 5-5(a), the bike counts in the station only changed slightly during the day, so the difference between the actual and predicted curves is relatively low, and, thus, we achieve a low prediction error: 0.32 bikes/station. In Figure 5-5(b), the station had more bike activity than Figure 5-5(a), but generally the expected pattern follows the actual curve with a small delay in responding to the jumps. In Figure 5-5(c), the station experienced a high demand and went out of service (i.e., bike inventory dropped to zero between 8:00 a.m. and 9:45 a.m.). Also, at 4:45 p.m. the station received 14 bikes, which caused inventory to jump from 5 to 19 bikes within 15 minutes. Consequently, the predicted curve could not follow these sudden changes in the actual curve, leading to a high prediction error: 1.86 bikes/station.

(a) Saturday - 9/6/2014 (prediction error = 0.32)



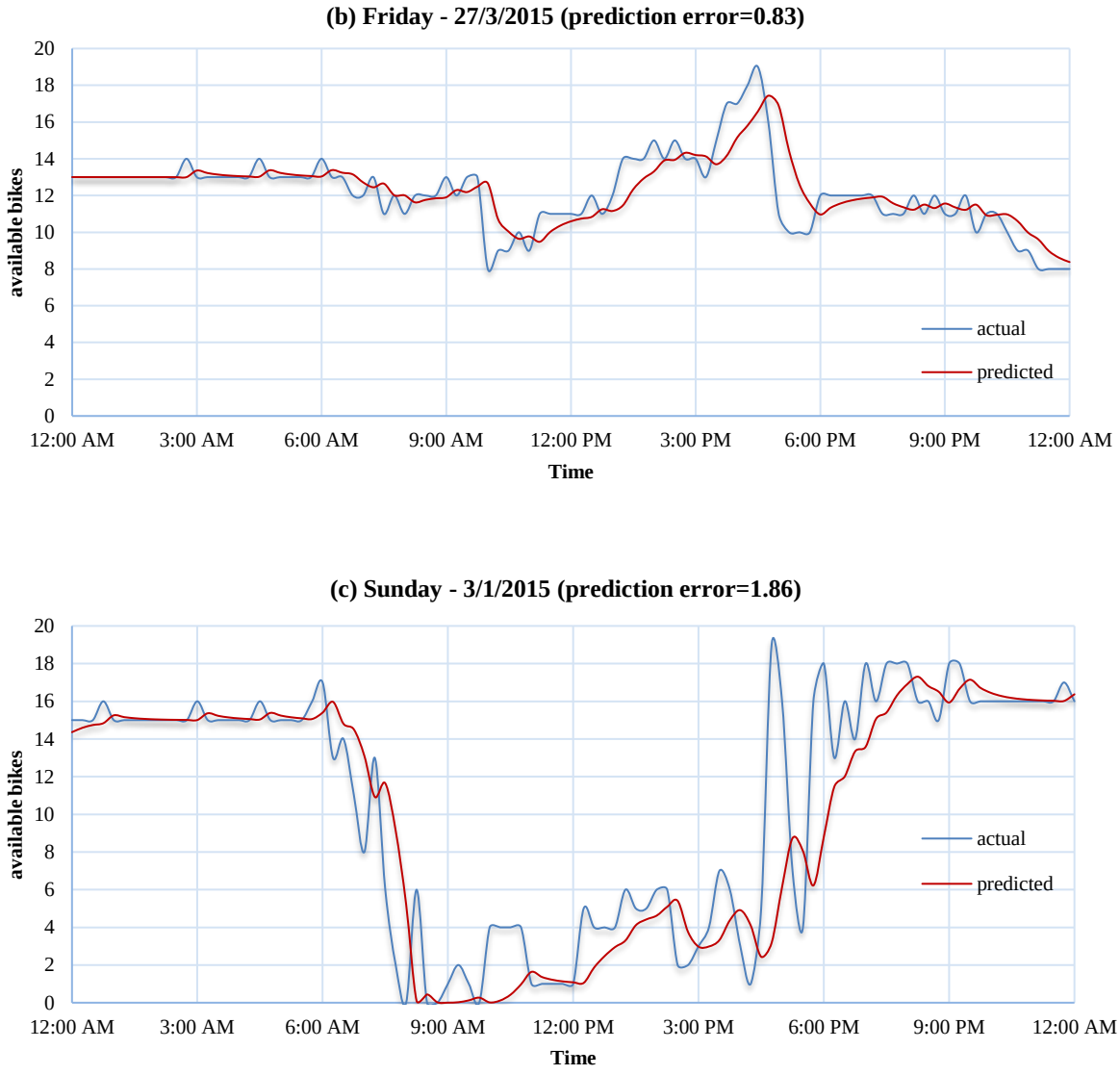


Figure 5-5 Pattern of expected and actual bike availability for three different days of the week at 15-minute prediction window of one station for first-order DLM, multiple-step technique

5.1.10 Comparison With Other Machine Learning Algorithms

In (*24*), we adopted two machine learning algorithms: RF and LSBoost using the same dataset to model the number of available bikes at each station. The input variables for these two models for each station were six weather variables (mean temperature, mean humidity, mean visibility, mean wind speed, precipitation, and events in a day), the available bikes at the 10 nearest neighboring stations, and the month, day of week, and time of day. These input variables were chosen based on subject-matter expertise, previous studies (*51*, *98*), and also were found to be significant. We compared the best result of these two models (140 trees for RF and 180 trees for LSBoost) to the multiple-step approach of the first-order DLM result (Figure 5-6). Although the LSBoost and RF algorithms were adopted using 19 variables and the DLM does not use regression components (no predictors), the first-order DLM of the multiple-step approach outperforms the LSBoost at all the

prediction windows. It gives the same prediction error as RF at the 15-minute and 30-minute prediction windows.

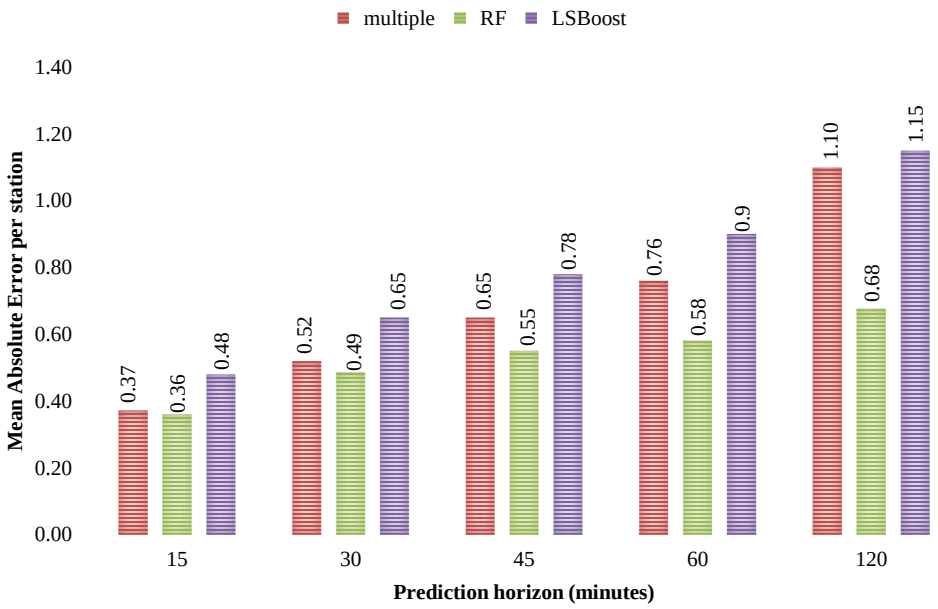


Figure 5-6 Multiple-step approach of the first-order DLM, RF, and LSBoost MAE at different prediction windows

This comparison reveals the good performance of the DLMs using the multiple-step approach compared to other statistical and sophisticated machine learning algorithms. Given that DLMs are linear, they can be easily extended to incorporate external factors such as weather information, seasonality, etc., that might improve the prediction much more than the results presented here.

5.1.11 Conclusions

BSSs are expanding and becoming a reliable transportation mode across the world, yet suffer from logistical challenges in which some stations run out of bikes and others become full of bikes. The first step to solve this issue is to predict bike demand in advance to help both bikers and operating agencies be part of the solution. Bikers could use the predicted demand to plan ahead and change their destination, while BSS managers could relocate bikes using service trucks from saturated to non-saturated stations. This research makes use of two well-known DLMs: first-and second-order polynomial models to predict the bike counts at stations in a BSS in the San Francisco Bay area. The two DLMs were adopted to create univariate models for 70 stations. Different prediction horizon windows were used: 15, 30, 45, 60, and 120 minutes to investigate the effect of the length of the prediction horizon on the prediction accuracy. Short prediction windows (15, 30, and 45 minutes) can be used to inform bikers of a station's status in advance (and thus mitigate the impact of logistical challenges), while the longer ones (60 and 120 minutes) enable operating agencies to relocate bikes.

Results reveal that both DLMs predicted the bike counts at stations with high accuracy and errors as low as 0.37 bikes/station (corresponding to a percentage error of 2% using the third measurement, MAE/C) for 15-minute prediction horizons. The prediction error increased as the time horizon increased, with a prediction error of 1.1 bikes/station for a 2-hour prediction horizon

(corresponding to a percentage error of 6% using the third measurement, MAE/C). Although the DLMs that were adopted in this chapter did not use any other external variables such as weather or spatiotemporal information, our results show they outperform the LSBoost algorithm for both short and long prediction horizons.

In the future, we will extend our work by incorporating more predictors in the DLM model such as weather information, seasonality, and availability and location of the other public transportation modes (bus or metro) and their schedules. In addition, we will investigate the benefit of clustering the months or days, and, then, adopt a DLM for each cluster

5.2 Incremental Learning Models of Bike Counts at Bike Sharing Systems

5.2.1 Introduction

Many cities have realized the negative effects of the increasing number of vehicles on the roads, such as greater congestion, emissions, and pollution rates. In response, various cities have discussed methods to reduce these rates. For example, in South Korea, a massive, first-of-its-kind 100 million square foot city is being designed to reduce or even eliminate the need for cars. At a cost of $35 billion, completion of this district is expected by 2020 (*99*).

Bike sharing systems (BSSs) have also been shown to be an energy-efficient and reliable transportation mode, and have been introduced in 1,139 cities and over 50 countries (*26*). In the San Francisco Bay Area, Saltzman and Bradford found that 92% of all weekday trips using BSSs were made by daily commuters going to and from work, showing significant faith in the BSS's reliability (*38*). According to the National Association of City Transportation Officials in the U.S, in 2016 alone there were over 28 million bike trips, an increase of 25% compared to 2015. This increased usage of bikes led many cities to either expand their existing system or launch a new one. For example, Ford, the operator of the GoBike BSS in the San Francisco Bay Area, started the system in 2013 with 700 bikes and 70 stations, and now plans to expand their system to 7,000 bikes and over 300 stations by the end of 2018 (*4*).

Due to the unbalanced spatial-temporal demand of bike trips, many bike stations become empty or full during the day. This significantly affects the reliability and usefulness of the BSS, which may prompt riders to return to using their personal cars or to adopt another transportation mode, consequently increasing congestion and thus auto emissions and pollution. This in turn, would lead to a decrease in the number of BSS users, reducing the system's revenue. Operating agencies have recognized the imbalance issue and have started to establish more bike stations close to one another, aiming to keep them within no more than a 5-minute walk (*4*). However, this solution is difficult to implement, both financially and practically.

Researchers have been investigating the imbalance issue and have recommended potential solutions to mitigate this issue with minimal cost and effort. Generally, these efforts can be categorized into three major approaches: static, dynamic, and incentivized. The underlying concept of the first two approaches is to move bikes between stations using a fleet of trucks either during or at the end of the day (*28-31*). The incentivized approach aims to encourage bikers to change either their origin or destination in favor of balancing the system (*32*).

An essential part of the rebalancing efforts is to predict the bike counts at stations accurately and quickly so that an imbalance can be discovered in advance and plans can be made accordingly. Predictions can be either used as an input for the three rebalancing approaches or can simply be given to bikers using a smartphone app to help them organize their trips. A good predictive model

can improve the rebalancing efforts and thus increase the reliability and efficiency of the system.

Researchers have used different methods to predict bike counts at stations, such as regression, count models (*11, 12*), clustering, and exploring algorithms (*13, 14*), machine learning algorithms (*15, 16*), and time series techniques (*14, 17*). All of these methods use many input variables, such as weather and time information, making them complex. Additionally, these models generally are static rather than dynamic, meaning that they do not adopt dynamic change over time.

This year, a study was published in the field of crime predictive models showing that a very simple model (i.e., a linear model) with only two features has almost the same predictive accuracy as other machine learning algorithms (ML) with up to 137 features (*100*). This raises the question of why so many factors are needed in a model when the same accuracy (or close to it) can be achieved using simple (and thus fast) models. A quick and simple predictive model for bikers would allow them to be informed and adjust their routes before heading to their destination, and would also help keep the system balanced.

In this chapter, we adopted two dynamic, easy-to-interpret, rapid approaches to predict bike counts at stations in a BSS: mini-batch gradient descent for the linear regression (MBGDLR) and locally weighted regression (LWR). These two approaches were built using an incremental learning concept based on previous knowledge (i.e., the previous status of the station) with neither weather nor time information. The two proposed models were applied to a BSS data set for one year (2014–2015) in the San Francisco Bay Area at different prediction windows: 15, 30, 45, 60, and 120 minutes. Our results show that both MBGDLR and LWR algorithms perform well, with high accuracy and errors as low as 0.30 bikes/station for a 15-minute prediction window and as low as 1.1 bikes/station for a 120-minute window.

5.2.2 Related Work

Bike prediction approaches have mainly taken one of four approaches: statistical models, exploring and clustering algorithms, machine learning algorithms, and time series models. Each approach has a different level of complexity with varying numbers of independent variables, such as time information, neighboring stations, and weather information.

Rudloff and Lackner used three count models: Poisson, negative binomial (NB), and Hurdle models to predict bike demand using temperature, precipitation, and neighboring stations as predictors (*43*). They used bike data from the bike sharing system Citybike Wien in Vienna, Austria and concluded that the Hurdle model outperformed the other two. Wang et al. adopted log-linear and NB regression models with 13 regressors as independent parameters (*12*). These 13 regressors included socioeconomic, demographic, and geographic information. They showed that all 13 regressors were significant and fit well with both models. Rixey adopted multivariate linear regression models to predict bike ridership using demographics and built environment characteristics near the BSS (*11*). They used three bike sharing systems and concluded that the factors used were significant. Ashqar et al. investigated the significant factors on bike demand, and using Random Forest (RF) found that time-of-day, temperature, and humidity level are significant predictors in bike prediction (*44*). They adopted two count models: Poisson and NB along with RF; their results showed that RF outperforms the other two models.

Due to the size of BSS data sets, several studies were conducted using visualization and clustering approaches and considering spatial and temporal information. Froehlich et al. utilized a clustering approach to predict bike counts in two steps (*13*). The first step was to investigate the relationship

between human behavior, geography, and time of day. The second step was to predict bike counts based on the three aforementioned factors. They divided bike stations into clusters and then predicted bike counts for each cluster. Their findings demonstrated neighboring stations were highly correlated and thus they were treated as one cluster. Similarly, Vogel et al. used clustering approaches to group stations with respect to the bike pickup and return activity (*48*). Based on the geographical information, they clustered bike stations into five groups and then provided average pickup and return rates for each hour.

Recently, machine learning approaches have been shown to be promising for predictive models due to their remarkable ability to learn from the data set and account for many predictors to discover hidden data set patterns. (*15, 16*). Ashqar et al. adapted three models: RF, Least-Squares Boosting (LSBoost), and Partial Least-Squares Regression (PLSR). The authors used six weather variables, 10 nearest neighboring stations, the month, day of week, and time of day (*15*). Their analysis showed that RF outperformed the other two methods, and also that RF kept the prediction error from increasing constantly as the prediction window increased, unlike the other models. Yang et al. used deep learning (i.e., a convolution neural network) to predict the daily usage of bikes (*16*). They used weather information, neighboring stations, and day of week as inputs for the models, and showed that the convolution neural network outperformed both the neural network and the autoregressive moving integral average model.

However, the previous three approaches suffer from the following: (1) they are static models, meaning they are trained once and remain the same and thus cannot capture the dynamic change over time, (2) they require many predictors, and (3) they are computationally expensive and thus cannot be used as online models.

Machine learning algorithms can be categorized into two major approaches: batch and online (or incremental) learning approaches (*49*). The batch approach is meant to use all the observed data at once and produce fixed coefficients of the model, while the online learning approach uses the observed data once they arrive and then produces dynamic coefficients over time, leading this approach to be faster. According to the literature, the first approach (i.e., batch) has been used for bike prediction, although it suffers from the three aforementioned drawbacks. To the best of our knowledge, the second approach has not been adapted for bike prediction.

The online machine learning approach is mainly proposed to handle systems that cannot tolerate a large processing delay. Its power comes from the fact that it is flexible enough to be applied to most machine learning algorithms. For the sake of simplicity, we chose two simple machine learning algorithms: stochastic gradient descent for linear regression and locally weighted regression. These two algorithms are dynamic and use no predictors aside from previous knowledge and both have a small computational time.

5.2.3 Methods

5.2.3.1 Mini-batch Gradient Descent for Linear Regression (MBGDLR)

In 1972, Nelder and Wedderburn developed a non-Bayesian approach to improve the classical static regression models by proposing generalized linear models (*101*). One of the proposed generalized linear models was the incremental learning linear regression model. This model is a stochastic approximation of the gradient descent optimization and is an iterative method for minimizing an objective function. It is built based on the classical linear regression model in which we make the coefficients (β) dynamic, meaning that they change over time.

For the multiple linear regression (MLR), we have input-output pairs: $(x_1, y_1) \ldots (x_n, y_n)$ where $x_i \in R^m$ and $y_i \in R$ for $i = 1, \ldots, N$. Assuming the relationship between $x's$ and $y's$ are linear with $E[y_i] = x_i^T \beta$ and the loss function for any x_i (i.e., the objective function) is the squared loss, then $f(y_i, x_i^T \beta) = (y_i - x_i^T \beta)^2$ where β denotes the regression coefficient's vector. The gradient of the loss function is $-2(y_i - x_i^T \beta)x_i$. The negative of the gradient helps move the β in a direction that decreases the loss function to find the optimal values of the coefficients (i.e., minimizing the current loss function will lead to minimizing the error and providing a better fit for the model).

The dynamic linear regression coefficients are estimated using a stochastic gradient descent. At time t, we receive the t-th observation and thus we predict the output using the previous dynamic coefficient (i.e., B_{t-1}) as follows:

$$\widehat{y_t} = x_t^T \beta_{t-1} \quad (1)$$

Once we receive the true value of the output (y_t), we can update the dynamic coefficient (B) considering the previous observations (as β_{t-1}) plus the new data point as follows:

$$\beta_t = \beta_{t-1} + 2\,\alpha(y_t - x_t^T \beta_{t-1})x_t \quad (2)$$

Where α is the learning rate in which we determine how much weight we want to give to this new arrival point. The higher the value is, the more stochastic the observations are.

As shown in (2), we update the regression coefficient immediately every time we receive the true value of the response. A better way to update the regression coefficients is the mini-batch approach, which calculates the gradient of a selected number (W) of data points and updates the regression coefficients as shown in Eq. 2 and (3).

$$\beta_t = \beta_{t-W} + 2\,\sum_{j=0}^{W} \alpha\left(y_{t-j} - x_{t-j}^T \beta_{t-W}\right)x_{t-j} \quad (3)$$

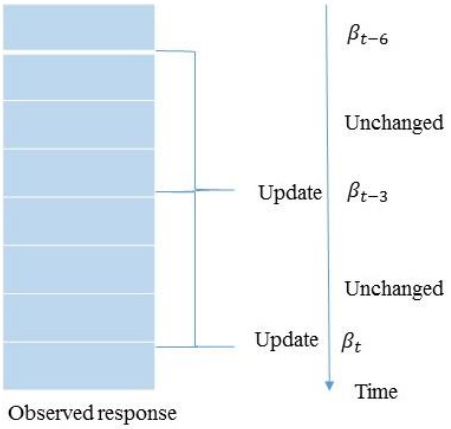


Figure 5-7 Illustration of the regression coefficients updating process (W=3)

As shown in Figure 5-7, the number of coefficient updates are fewer and hence the coefficients are more stable.

Note that this approach applies to both single and multiple linear regression and the same approach applies for each coefficient separately. Although we assume the relationship between $x's$ and $y's$ (globally) are linear, the predicted line does not have to be linear, as we calibrate the coefficients locally not globally.

The MBGDLR algorithm needs two parameters—the learning rate (α) and the mini-batch size (W)—to be tuned, and thus a sensitivity analysis must be carried out to find the optimal values, as shown in the "Model Testing" section. Moreover, the initial values of the β need to be set up and then continuously updated based on the new arrival of data points. To accomplish this, it is necessary to first determine the size of the sample to be used in calculating the initial coefficients. Again, more details are provided in the "Model Testing" section.

5.2.3.2 Locally Weighted Regression (LWR)

LWR is a form of memory-based or lazy-learning algorithm for learning continuous non-linear mappings from real-data to predicted vectors. It is considered a local learning approach due the fact that it considers only a particular moving window (W_L) when calibrating model parameters. When predicting a new data point at time $t+1$ (e.g. $\hat{y}_{t+1}$), the model uses only the data points that are inside the moving window (e.g., if the moving window is 5, then we would use $\{(x_{t-4}, y_{t-4}), \dots (x_t, y_t)\}$) and then gives them weights based on a weight function. The weight function assigns weights to each of the five points inside the window W_L based on the distance between x_{t+1} and each x inside the window. There are different weighting functions (i.e., kernel functions), and here we used the most common function: Gaussian (4). The distance (d) between the point of estimation (x_{t+1}) and other points inside the moving window are squared and then used in the Gaussian function as shown in (4).

$$K = diag(e^{-\frac{d_i^2}{2\sigma^2}}) \quad (4)$$

Where k_i is the weight of the i^{th}data point in the moving window, d_i is the distance between the i^{th}data point inside the moving window and the point of estimation (x_{t+1_t}), and σ^2 is the variance of the kernel.

Then, the weight $(K)\ matrix$ will be used in the Hat matrix to estimate the new coefficient regression $\beta_{t+1}$to predict $\hat{y}_{t+1}$ as shown in (5) and (6).

$$\beta_{t+1} = inv(X' * W_i * X) * X' * W_i * Y \quad (5)$$

$$\hat{y}_{t+1} = x_{t+1}^T \beta_{t+1} \quad (6)$$

Where X is the design matrix and consists of the x's of points inside the W_L. and Y is the vector of the corresponding responses. Note that there are two tuning parameters that need to be determined when adapting LWR: the size of the moving window (W_L) and the variance of the kernel (σ^2). More details are provided in the "Model Testing" section regarding the optimal values used in this research.

5.2.4 Data Set: Case Study of San Francisco

This study used a publicly available BSS docking station data set. The case study data set covers the period from September 1, 2014, to September 1, 2015, in the San Francisco Bay area for 70 stations across five different zip codes. The data set includes station ID, number of bikes available, number of docks available, and time of recording. Each row has the number of available bikes at the 70 stations with the associated time (day of week and hour). As the station data were collected at a frequency of every minute for 70 stations in San Francisco over a year of 2014–2015, the data set is quite large. Consequently, we derived a subset of the original data set by sampling station

data once at every 15 minutes and obtaining the exact values without any smoothing process. This was done to reduce the complexity of the data and avoid running out of memory. The subset was tested to make sure it represented the population and the analysis showed it to be representative of the entire data set. When analyzing the data set, we noticed big jumps in the numbers of returned and taken bikes at specific times at some stations; we suspect these indicate periods of rebalancing operations. However, we did not exclude these jumps when making predictions, as we could not get confirmation of this suspicion from the operating agency (Ford GoBike).

5.2.5 Results and Discussion

5.2.5.1 Model Testing

Given that there are tuning parameters in the two models, we conducted a sensitivity analysis to find the optimal values. For the MBGDLR algorithm, we found that all prediction horizons behaved in the same way when changing the two tuning parameters: mini-batch size (W), and the learning rate (σ). The optimal values for W and α for all prediction horizons were 1.5-hours (6 steps) and 0.0055 respectively. We found also that the prediction accuracy starts to decrease significantly at $W =$ 24-hour (96 steps) and α = 0.01. For the sample size used to calculate the initial coefficients ($\beta's$), our analysis shows a seven-day window is sufficient to calibrate the coefficients.

For the LWR algorithm, there are two tuning parameters: the size of the moving window (W_L) and the variance of the kernel (σ^2). Our analysis shows that W_L starts giving reasonable results after a length of around half a week and then the prediction accuracy starts improving very slightly as W_L increases. Therefore, we had to compromise between obtaining good prediction accuracy and achieving a small computational time (i.e., increasing W_L would make the model slower as the data got bigger) Choosing a 1-week window as the optimal W_L allowed us to achieve both of our goals for all prediction windows.

For σ^2, the optimal value is based on the prediction horizon. The smaller prediction horizons (15 and 30 minutes) tended to behave slightly better with large variance (16) while the longer prediction horizons (45, 60, 120 minutes) performed somewhat better with small variance (1). Accordingly, increasing the prediction horizon would require decreasing the variance to get a better accuracy result.

5.2.5.2 Evaluation Criteria

To measure the predictive accuracy of the two models, two different measurements were used: The mean absolute forecast error (MAE) and the symmetric mean absolute percentage error (SMAPE). The MAE (well-known as a prediction error) was calculated by taking the average of the absolute difference between the anticipated and actual number of the bike counts for all 70 stations in the entire year (7). The SMAPE is an accuracy measure and is calculated as shown in (8).

$$\text{MAE} = \frac{\sum_{i=1}^{n} |Y_t - A_t|}{n} \tag{7}$$

$$\text{SMAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \frac{|Y_t - A_t|}{(|A_t| + |Y_t|)/2} \tag{8}$$

where n is the number of observations, and Y_t and A_t are the predicted and actual number of bike

counts respectively.

5.2.5.3 Results

We used the aforementioned optimal values for both MBGDLR and LWR algorithms to predict the bike counts at 70 stations in the San Francisco Bay Area at different prediction horizons. The results, given in Table 5-2, show that LWR performs slightly better than MBGDLR for all prediction horizons. The smallest prediction error was 0.309 bikes/station (4% prediction error) under a 15-minute prediction horizon while the prediction error was 0.318 bikes/station using MBGDLR. As shown in Figure 5-8, the prediction error increased as the prediction horizon increased, with the 120-minute prediction horizon having the largest prediction error at 1.1 bikes/station and 1.2 bikes/station for LWR and MBGDLR respectively.

Table 5-2 Performance comparison of MBGDLR and LWR at different prediction horizons

PREDICTION HORIZONS (MINUTES)	MBGDLR		LWR	
	MAE	SMAPE	MAE	SMAPE
15	0.318	0.04	0.309	0.04
30	0.514	0.06	0.488	0.06
45	0.676	0.08	0.633	0.75
60	0.813	0.09	0.756	0.086
120	1.2	0.13	1.101	0.11
Average	0.7	0.08	0.66	0.074

Although LWR performed slightly better than MBGDLR, the former takes much longer than the latter to return a prediction. The computational time for LWR was 45 times longer than it was for MBGDLR. When increasing the batch size and the window size for both MBGDLR and LWR respectively, the computational time was greatly increased for LWR but was not when using MBGDLR (PC configuration: Intel® Core™ i7-6700 CPU @ 3.40GHz, Ram 16 GB, 64-bit operating system, x64-based processor).

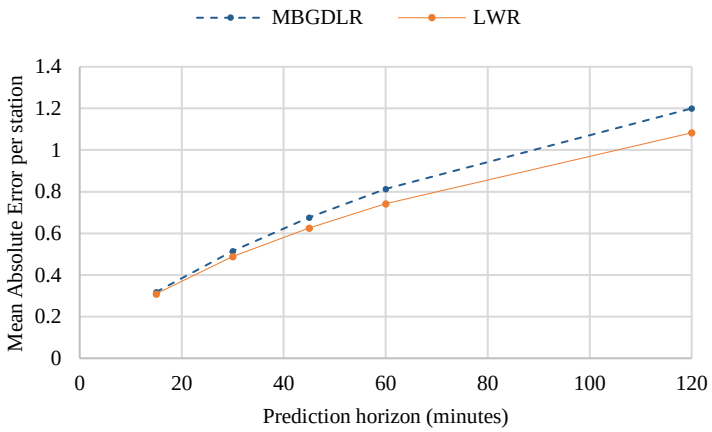


Figure 5-8 Prediction error for MBGDLR and LWR at different prediction horizons

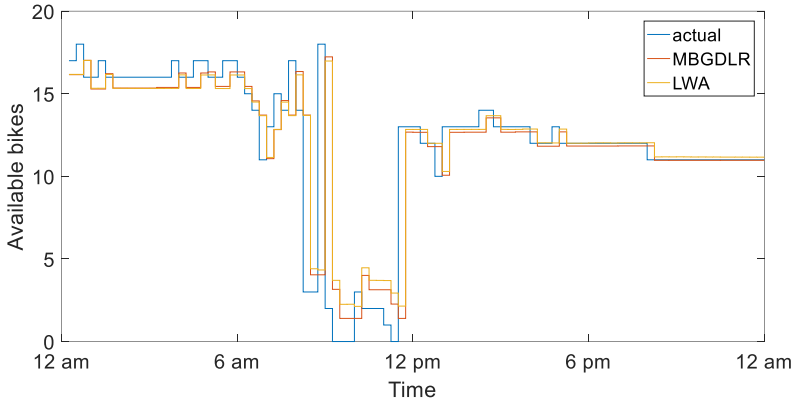


Figure 5-9 One-day pattern of expected and actual bike availability at 15-minute prediction window for MBGDLR and LWR algorithms, station 59

To investigate why LWR performed slightly better than MBGDLR, though the expected pattern does not differ much (Figure 5-9), we looked at station-level prediction error for all prediction horizons at all stations and compared both algorithms. Results are shown in Figure 5-10 (note: only the 15-minute prediction horizon is presented here as an example). As Figure 5-10 shows, predictions are in line for almost all stations except stations 15, 17, and 18, with better results shown for LWR. Analyzing the patterns of these three particular stations led us to conclude that they are slightly different compared to other stations. They look more stable at some point during the day and thus produce a small variance, holding an overfitting issue.

Based on Figure 5-10, the largest prediction error happens for both algorithms at stations 41, 58, and 59. To understand this, we investigated the stations' patterns and found them to be very dynamic, indicating that they ran out of bikes or racks almost every day. To verify this, we performed a spatial analysis and found that stations 58 and 59 are next to each other and are also located quite close to a train station in San Francisco, making them more likely to be affected by the trains' timetables. And while station 41 is far from stations 58 and 59, and is not close to any train station, an examination of the BSS's adjacency matrix revealed that station 41 and 59 are highly correlated and connected. This means that station 59 receives the highest number of bikes from station 41, especially at 5:00 p.m., compared to other stations, as shown in Figure 5-11. Approximately 15 minutes after bikes are taken from station 41, station 59 starts receiving almost the same number of bikes (15 minutes is the approximate bicycling time between stations).

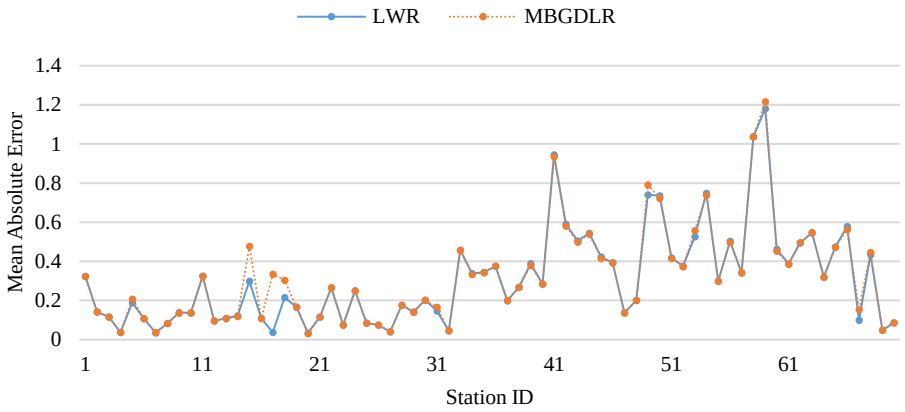


Figure 5-10 MAE per station for MBGDLR and LWR algorithms across stations at a 15-min prediction window

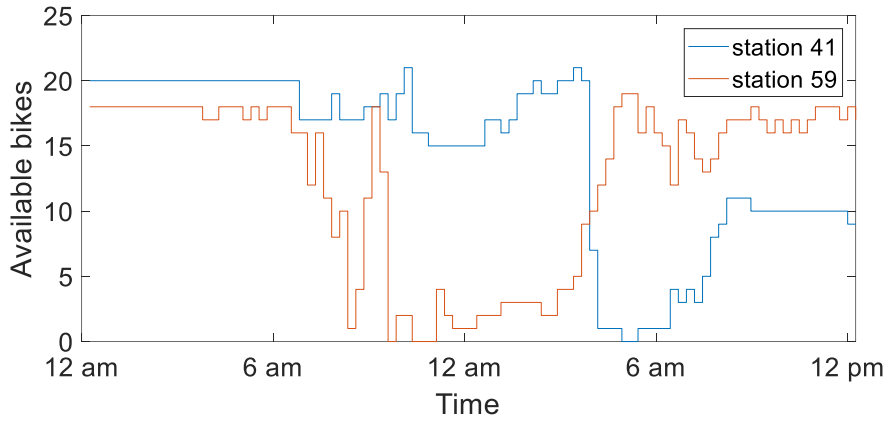


Figure 5-11 Pattern of bike availability for stations 41 and 59

5.2.6 Comparisons with Other Algorithms

We compared the results of MBGDLR and LWR algorithms with one online and two offline algorithms: the first order of the dynamic linear model (DLM; (*102*)), RF, and LSBoost. The two off-line models (RF and LSBoost) used 20 predictors (e.g., time, weather, and neighboring information) and were implemented with an optimal number of trees for producing the best accuracy:180 and 140 trees for RF and LSBoost respectively (*44*). Note that the off-line models had to be built using predictors. The online model (DLM) used only the previous station status, and was built using the optimal values of the variance of the noise for observation and evolution equations.

As shown in Figure 5-12, all algorithms returned a comparable prediction accuracy under 15-minute and 30-minute prediction windows, with the exception of LSBoost. For the rest of the prediction windows, RF outperformed all other algorithms. However, when comparing the computational time for the five algorithms, RF had the largest running time, followed by LWR. MBGDLR had the smallest computational time, followed by DLM. Although RF gives the smallest prediction accuracy, it takes longer to predict (77 times longer than MBGDLR and 12 times longer than DLM).

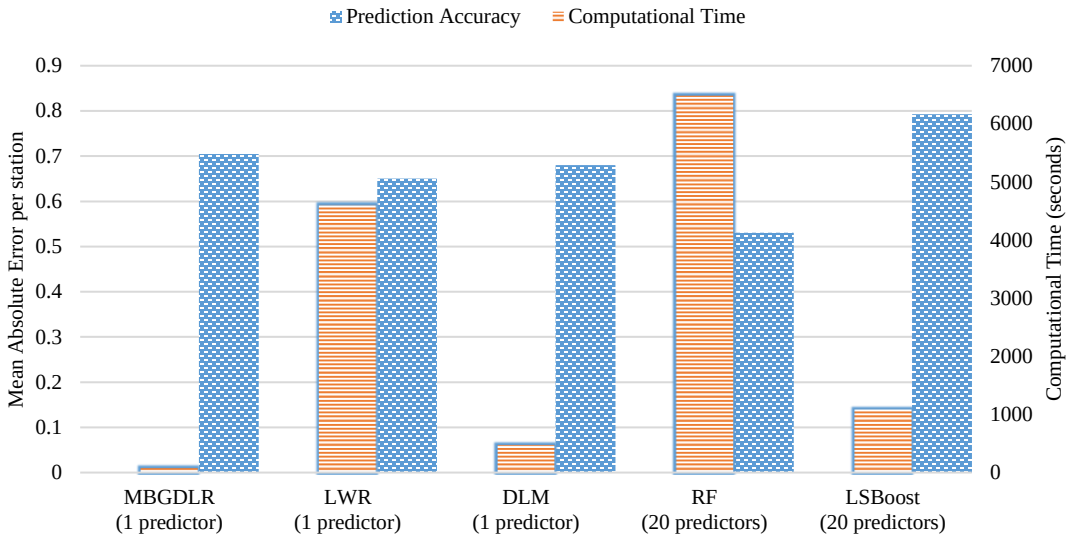


Figure 5-12 Comparison of the average computational time and MAE of all prediction winodws for MBGDLR , LWR, DLM, RF, and LSBoost algorithms for all 70 stations

Based on the previous comparison considering both prediction accuracy and computational time, we can conclude that MBGDLR is better than the rest of the algorithms due to its ability to predict with a relatively small prediction error in a very short time. That makes MBGDLR a promising algorithm for implementation in BSS apps that inform bikers about station statuses in advance.

One way to explain the differences in computational time between these algorithms is to look at the mechanism of each. MBGDLR outperforms all other algorithms in terms of computational time due to the simplicity of its linear regression form. Also, the MBGDLR model can be used as either univariate or multivariate given that the parameters are the same.

Although DLM has the same concept, it appears to be slower due to its need to estimate the variance of noise for the whole data set. That makes it theoretically difficult to estimate and might lead to instability, especially with a large data set (i.e., a lot of matrices would lead to a singular matrix that could not be inversed).

5.2.7 Conclusions

BSSs have increased and expanded in many cities in over 50 countries, reducing the negative impact of the increased number of motor vehicles on the roadways. However, imbalances reduce BSS's efficiency, resulting in some stations running out of either bikes or racks. To remedy this, a quick online predictive model needs to be developed and either fed into BSS apps, so that bikers can be informed in advance and change their destination, or used for rebalancing models to redistribute bikes before imbalance occurs. This chapter adopted two online algorithms to predict bike counts at stations in a BSS: MBGDLR and LWR. These two algorithms were adopted to create univariate models and were then tested for 70 stations in the San Francisco Bay Area. Different prediction horizon windows were used: 15, 30, 45, 60, and 120 minutes. Short prediction windows (15, 30, and 45 minutes) can be used to inform bikers of a station's status in advance (and thus mitigate the impact of logistical challenges), while the longer windows (60 and 120 minutes) enable operating agencies to redistribute bikes.

The results show that LWR performed slightly better than MBGDLR for all prediction windows.

The smallest prediction error was 0.309 bikes/station for LWR compared to 0.318 bikes/station for MBGDLR under a 15-minute prediction window. The prediction error increased as the prediction window increased, and the 120-minute prediction window had the largest prediction error with 1.1 bikes/station and 1.2 bikes/station for LWR and MBGDLR respectively. MBGDLR was shown to be 55 times faster than LWR.

A comparison was made with DLM, RF, and LSBoost, and the results revealed that RF outperformed all other algorithms but was very slow, and thus unsuitable for use as an online model. When taking into account both prediction accuracy and computational time, MBGDLR was shown to be the best model to use for prediction. Further, it does not use any other external variables, such as weather or time information, making it simple for practical use.

CHAPTER 6 CAN PORTABLE STATIONS RESOLVE BIKE SHARE SYSTEM STATION IMBALANCES?

6.1 Introduction

Due to the large increase in vehicles on the road over the years, cities face challenges in providing high-quality transportation services. Traffic jams are a clear sign that cities are overwhelmed, and that current transportations networks and systems cannot accommodate the current demand without a change in policy, infrastructure, transportation modes, and commuters' choice of transportation mode. In response to this issue, cities in a number of countries have started putting a threshold on the number of vehicles on the road by deploying a partial or complete ban on cars in the city center. For example, in Oslo, leaders have decided to completely ban privately-owned cars from its center by the end of 2019, making it the first European city to totally ban cars in the city center. Instead, public transit and cycling will be supported and encouraged in the banned-car zone, and hundreds of parking spaces in the city will be replaced by bike lanes. As another example, in Dublin, Ireland, a proposal has been made to totally ban privately-owned cars from selected areas of the city center and push for public transit and bicycle use (*2*).

As an effort by governments to support bicycling and offer alternative transportation modes, bike-sharing systems (BSSs) have been introduced over 50 countries (*1*). BSSs aim to encourage people to travel via bike by distributing bicycles from stations located across an area of service. Residents and visitors can borrow a bike from any station and then return it to any station near their destination. Bicycles are considered an affordable, easy-to-use, and, healthy transportation mode, and BSSs show significant transportation, environment, and health benefits. In transportation, BSSs replace partially privately-owned car trips with bicycling, thereby mitigating traffic jams in the city. A survey conducted by McNeil at al. found that 80% or more of BSS users said they use BSSs for shopping/errands, social/recreational, trips to and from public transit, and commute trips (*3*), confirming that BSSs are becoming a reliable and convenient transportation mode for both recreational and non-recreational trips. In the environmental and health fields, the reduction in privately-owned car trips means less carbon energy consumption and carbon emissions. Qiu and He found that using BSSs in Beijing could save workers 8 minutes per day and that this saving could result in reducing fuel consumption by 225.05 thousand tons (*4*). This would contribute in increasing the GDP of Beijing by Ren Min Bi (RMB) 1.2 billion (RMB is the official currency of china) and reducing the health costs by RMB 2420.57 million yuan.

As the use of BSSs has grown, imbalance has become an issue and an obstacle for further growth. Imbalance occurs when bikers cannot drop off or pick-up a bike because the bike station is either full or empty. This problem has been investigated extensively by many researchers and policy makers, and several solutions have been proposed (*5-9*).The main approaches are static and dynamic, both of which deal with the movement of bikes between stations either during or at the end of the day to overcome imbalance. They both assume the location and number of bike stations are fixed and only the bikes can be moved. This is a realistic assumption given that current BSSs have only fixed stations. However, cities are dynamic and their geographical and economic growth affects the distribution of trips in cities and thus constantly changes BSS users' behavior. In addition, work-related bike trips cause certain stations to face a high-demand level during weekdays, while these same stations are at a low-demand level on weekends, and thus become useless (*18*). Moreover, fixed stations fail to accommodate big events such football games, holidays, or sudden weather changes.

One solution for adapting to these challenges is installing and reinstalling stations; however, this is costly and impractical. Taking a different approach, a new generation of BSSs was introduced in China in 2015—the dock-less (or station-free) BSS takes an approach in which the BSS does not have stations. Rather, bikes are distributed along city sidewalks. Residents and visitors can rent a bike from anywhere and leave it within a defined zone. Although this innovative approach partially overcomes the issue of imbalance and gives bikers more flexibility, it does create other problems. First, this system has created chaotic parking problem in high-density cities where users leave their bikes in inappropriate locations, especially during rush hours and in the city center and at tourist sites (*19*). Second, in low-density cities, bikes are often left in remote locations and thus become sparse in the city, making it more difficult for users to find a bike. Eventually, the efficiency and reliability of the BSS will be affected negatively and as a result, customer satisfaction and the BSS's revenue decreases.

In this chapter, we propose a new generation of BSS in which we assume some of the bike stations can be portable. This approach takes advantage of both types of BSS: dock and dock-less. This idea is supported by the fact that many bike stations, for example in the San Francisco Bay Area, are installed on streets (Figure 6-1), and thus can be easily linked to portable stations (*1*)[1]. The proposed portable stations can function as either individual stations (standalone) or as an extension of the existing bike stations. This concept is proposed to overcome the constraints of most current rebalancing algorithms in the following ways: (1) the locations of the docking stations are no longer fixed (2) the capacity (Q) of each station will become Q+X, where X represents the size of the portable station (3) the (un)loading time of bikes during repositioning operations would be zero, thus minimizing labor costs (4) there will be no time required for the portable stations to find parking, as they can be linked to the existing stations.

The goal of this research effort was to develop a simulation-based portable stations model as a proof-of-concept. A BSS of 35 stations in the San Francisco Bay area was utilized and tested using the proposed approach, and the results show that adding only one portable station to the BSS can reduce missed bike pick-ups and thus enhance customer satisfaction by approximately 10% on average compared to the traditional static approach. Moreover, adding one more portable station could reduce missed bike pick-ups by almost 25% and reduce repositioning operations by as much as three times.

Figure 6-1 Off-street bike station located at 594 Howard St., San Francisco (Source: Google Earth)

6.2 Related Work

Previous research efforts have been largely spent on two main rebalancing approaches: the static bicycle repositioning problem (SBRP) and the dynamic bicycle repositioning problem (DBRP). The SBRP neglects the bikes' movements while rebalancing the stations, so static repositioning is done overnight when there is minimal bike usage. Unlike the SBRP, the DBRP takes into consideration the bikes' movement while rebalancing, and can thus be done anytime during the day.

For the SBRP approach, research efforts vary based on the objective function, size of the service vehicles, the allowance of multiple visits, and the adopted technique (*29, 30, 52*). Espegren et al. proposed a model to minimize the deviation from the optimal status of the stations (*29*). The proposed model allowed for more than one visit for stations by a fleet of vehicles. Their objective function allows for a non-perfect solution. Caggiani and Ottomanelli developed a modular decision support system with an objective function of minimizing both deviation from the stations' optimal status and the cost of moving bikes between stations (*30*). Their proposed system also included finding the optimal time horizon and route for the service vehicle.

Chemla et al. adopted the branch-and-cut algorithm to rebalance bike distribution using only one service vehicle (*52*). The objective function was minimizing the distance traveled by the service vehicle. Elhenawy and Rakha proposed a rebalancing algorithm, called the deferred acceptance algorithm, based on the game theory algorithm. Their proposed algorithm had two phases: tour construction and tour improvement. The objective function was to minimize the total tour cost (*15*). Kadri and Kacem formulated the balancing problem mathematically with two lower and four upper bonds (*53*). The two lower bonds were developed based on Eastman's bound while the four upper bonds were based on a genetic algorithm. These bonds were incorporated in a branch-and-bound algorithm. The authors used a fleet of vehicles and aimed to minimize the duration of imbalanced stations.

These aforementioned studies using the SBRP approach assume the user's demand to be negligible

while performing the repositioning operations. Consequently, rebalancing efforts are conducted at the end of day, making rebalancing a day-to-day operation, which means that this approach fails to prevent imbalance during the day. As a response, researchers investigated a faster approach, the DBRP, to reposition bikes dynamically by allowing repositioning decisions to be adapted over the planning horizon. The DBRP approach was shown to give a better result than the SBRP approach due to its ability to rebalance continually during the day (*10, 26, 28, 31, 54-56*).

Recent DBRP research work differs mainly with regard to the objective function of rebalancing, routing and rebalancing technique, size of the fleet of the service vehicle, and scalability. Brinkmann et al. developed a dynamic model to overcome imbalance by incorporating two strategies: short and long (*31*). The short-term strategy aimed to find the bike stations that are at risk of being imbalanced, while the long-term strategy suggested a number of stations to be considered for repositioning operations based on the short-term strategy. The objective of their developed model was to minimize the number of times that stations are imbalanced with only one service vehicle. Contardo et. al used Danzig-Wolf and Benders decomposition to rebalance bike distribution using a scalable methodology with lower and upper bounds (*28*). The goal of their proposed model was to minimize stations' deviation from their optimal status using a fleet of service vehicles with large instances of stations. Ghosh et al. adopted a mixed integer linear programming approach with the goal of maximizing service and minimizing the cost of repositioning operations (*26*). They used clustering techniques for simplification purposes. Multiple service vehicles and large instances were used in the proposed model. Chiariotti et al. proposed a dynamic model using birth-death processes (*55*). They predicted stations' statuses and then determined the optimal time for repositioning operations. The graph theory was used to choose the optimal path to order service vehicle destinations.

Although the DBRP has advanced repositioning operations substantially, all existing approaches assume the stations are fixed, ignoring the dynamic spatial-temporal demand. For example, recent studies have shown the pattern of use differs significantly on weekdays and weekends, making some stations useless at certain times and days (*14, 18*). In addition, (*18*) showed that some stations experience imbalance only during specific weekdays but have low-demand on the other days of the week. As a real-life example, the GoBike BSS in the San Francisco Bay Area opened in 2013 and the locations of stations have changed significantly since then (*4*)[4][3]. That is due to the fact the city changes dynamically and thus the trips' distribution evolves, following new business and entertainment locations.

To the best of our knowledge, the stations' relocations were rarely based on academic research (*103*). In (*103*) , the authors proposed a mathematical model to formulate bike movements between stations as a scheduling problem using a mixed integer programming approach. However, the proposed approach is not applicable to BSSs because the developed model cannot accommodate their complex dynamics, which include bottlenecking in operations and uncertainty. To fill this gap, an agent-based simulation approach is proposed to address these issues. Computational experiments demonstrated obtainable high-quality solutions that are applicable in industry-scale applications. Based on the obtained results, several insights and recommendations were made regarding the use of portable stations.

6.3 Data Set

This study used Ford GoBike's BSS trip dataset, containing data collected from August 2013 to August 2015 in the San Francisco Bay Area. During that period, the BSS had 70 stations covering five cities (Figure 6-2): San Francisco (35 stations), Palo Alto (5 stations), Mountain View (7 stations), Redwood City (7 stations), and San Jose (16 stations). The dataset contains 669,960 trips, each of which includes bike ID, trip duration, trip start day and time, trip end day and time, trip start ID station, and trip end ID station. Another file called "station" was also used; this file contains geographic and operating information of each station, including latitude, longitude, capacity, city, and installation date.

During the analysis phase, we found that the demand during off-peak hours was stable (Figure 6-3), so we only considered the hours of 6:00 a.m. to 8:00 p.m. when simulating the network. That is, we assumed the level of demand at stations at the end of the day (i.e., 8:00 p.m.) was the same as at the beginning of the day (i.e., 6:00 a.m.). Also, for simplicity purposes, we used only trips that occurred on Mondays (104 days). We also only used stations located in San Francisco, as our analysis showed it had the highest imbalance compared to the other four cities.

During data preparation, a source-destination matrix was built to count all trips for each hour between each pair of stations, and then a probability transition matrix was created. This was done for all Mondays during the period from 6:00 a.m. to 8:00 p.m. Similarly, the associated travel times with source-destination matrix were extracted from the trips dataset. Given the presence of outliers in the trip dataset (either very short or long trips), we only extracted 95% of the trips, meaning we excluded the shortest and longest 2.5% of trips.

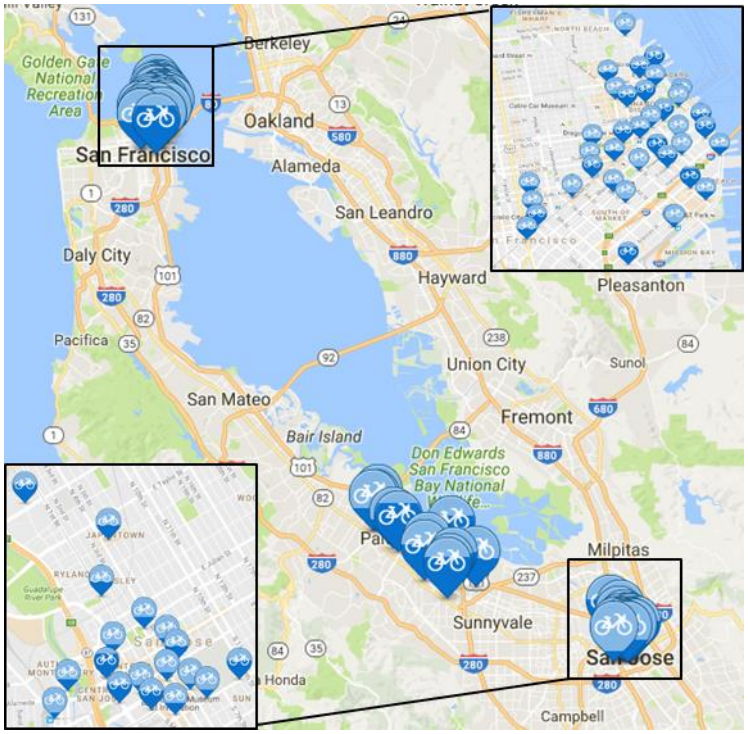


Figure 6-2 Locations of bike stations in San Francisco Bay Area (*88*)

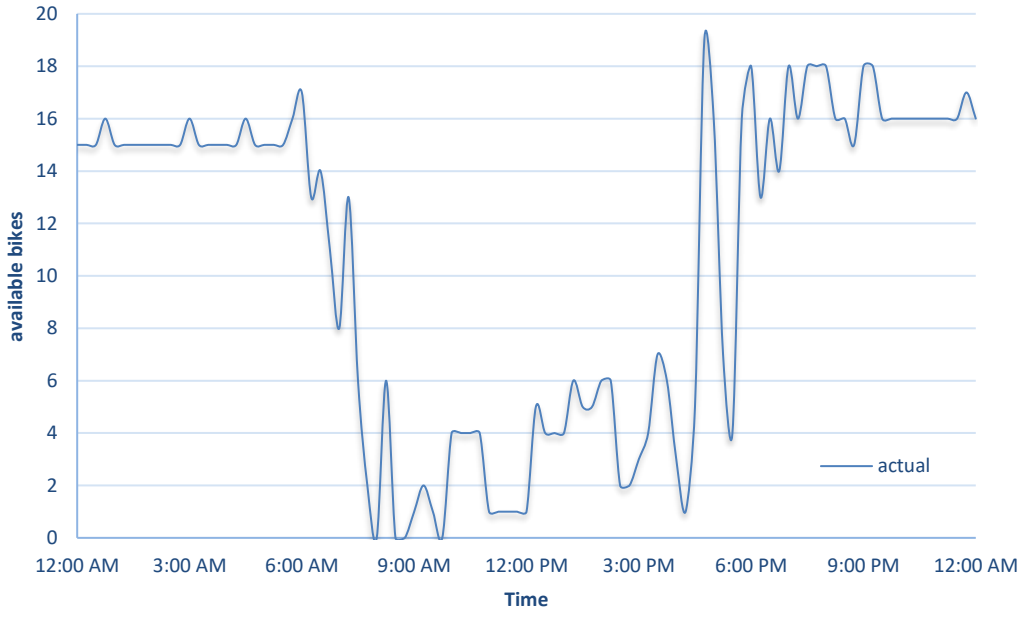


Figure 6-3 Bike counts for randomly selected station during one day

6.4 Agent-Based Simulation Model

We developed a BSS simulator using MATLAB software. The simulator was used to simulate the BSS in downtown San Francisco (35 stations; Figure 6-4). We investigated the proposed portable station by simulating two scenarios. In the first scenario, we added only one portable station to downtown San Francisco. In the second scenario, we increased the number of the portable stations by one. Due to the lack of socioeconomic information for the study area, we assumed the portable station could only be linked to the existing stations, meaning it could not be standalone. That is, we do not know the arrival rate for other points of interest; we know only the bike stations' locations. We assumed the capacity of the portable station to be Q bikes, and that it moved at a speed of 15 mph given that it is moving in a congested business area.

Our simulation model assumed the following:

1. The arrival of bikers at stations to pick-up bikes follows a Poisson process in which the hourly arrival rate was estimated based on 2-year historical data.
2. Bikers' travel time follows a Gamma distribution, where the Gamma distribution's parameters were estimated using the 2-year historical data between each pair of stations.
3. The bikers chose their destination based on a multinomial distribution whose parameters were estimated using the 2-years of data. In other words, we estimated a transition matrix with independent rows. Each row i in this matrix contains multinomial probabilities which control the transition from station i to station j where $i, j \in \{1,2, \dots ,34,35\}$.
4. In the case of a full station, bikers neither wait nor leave the bike unlocked. Instead, they start searching the BSS app for the nearest station with an empty rack to drop off the bike.
5. In the case of an empty station, the biker will find another mode of transportation.
6. Each station in the BSS starts the day at a half-capacity level. That is, we assume the chances are similar for both imbalanced states: empty and full. Therefore, the goal is for stations to remain at the same level so that there is no need for rebalancing in order for the system to operate the next day.
7. The portable station can start from any station at the beginning of the day, then keeps moving between stations until the end of the day.
8. Before a portable station leaves the station it is linked to, it tries to make the number of available bikes as close as possible to half capacity by picking up or dropping off bikes.
9. The portable station decides on the next station and moves to that station at the beginning of each hour with a speed of 15 mph.
10. Simulation was conducted every deci-second.

The portable station chooses the next station in a greedy way based on Equation (1) and Equation (2) as follows:

$$R_{t+1} = S_t - \lambda_{t+1} + {P_{t+1}^T}^T \lambda_{t+1} \quad (1)$$

$$\min_i \arg\left(\left|\frac{Q_j}{2}\right| - H_j + r_i\right) \quad (2)$$

Where

- S_t is a column vector where each element i is current available bikes at station i
- λ_{t+1} is a column vector where each element i is the arrival rate of bikers per hour at station i at time $t+1$
- P_{t+1} is the transition matrix at time $t+1$
- Q_j is the capacity of portable station j
- H_j is the number of bikes loaded on the portable station at time t
- r_i is the i^{th} element of the R_{t+1}

Equation (1) predicts the number of bikes at each station at time $t+1$ when the initial number of bikes at time t is S_t. As shown in Equation (2), the portable station prefers the station that will keeps it loaded at half capacity after it visits the station.

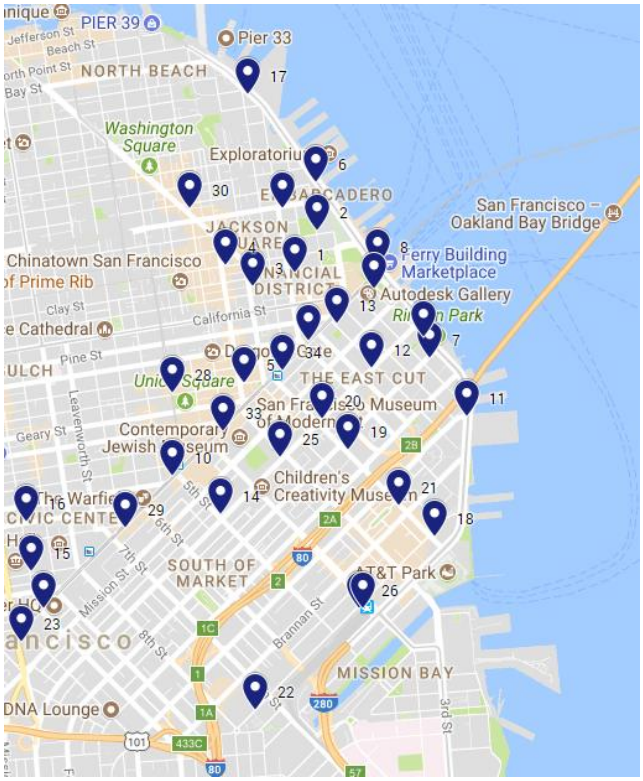


Figure 6-4 Locations of 35 bike stations in downtown San Francisco (Source: Google Maps)

6.5 Model Testing

6.5.1 Evaluation Criteria

We ran the simulation 35 times with and without the portable station. Two measures were used to quantify the benefits of the portable station. The first measure was the sum of missed bike pick-ups, which is the count of bikers arriving at BSS stations who are unable to find available bikes. Missed pick-ups due to BSS imbalances threaten the reliability and sustainability of BSSs and could result in reduced customer satisfaction. The BSS's operating agency loses its revenue and the bikers who are unable to use the BSS go back to using their own car, contributing to city

congestion.

The second measure was the sum of the absolute difference between the initial number of bikes available (start of the day) at each station and the number of bikes available at the end of the operation period/day (Equation 3)

$$\sum_{i=1}^{35} \left| B_i^{start\ of\ the\ day} - B_i^{end\ of\ the\ day} \right| \quad (3)$$

where B_i^t is the number of available bikes at station i at time t. This measure is important as it is related to the number of bikes that need to be relocated during the rebalancing process. We should highlight that the main goal of the portable stations is to increase user satisfaction with a byproduct of reducing the rebalancing effort at the end of the day.

6.5.2 Results

We ran a simulation of the proposed portable station 35 times, with each repetition representing a 24-hour day simulation with a different portable station starting point. The varied starting point was intended to add a randomness to the results and avoid any effects of a particular initial starting point. The aggregate results show that adding one portable station to the network can reduce missed pick-ups and thus enhance customer satisfaction by approximately 10% on average compared to the traditional SBRP approach (Figure 6-5 and Figure 6-6).

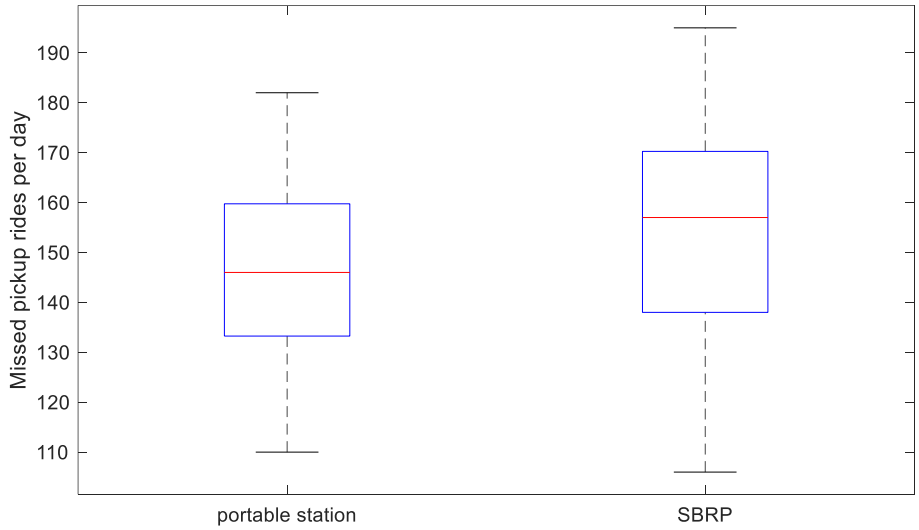


Figure 6-5 Box plot of the average missed bike pick-ups per day for the two approaches: portable stations and SBRP

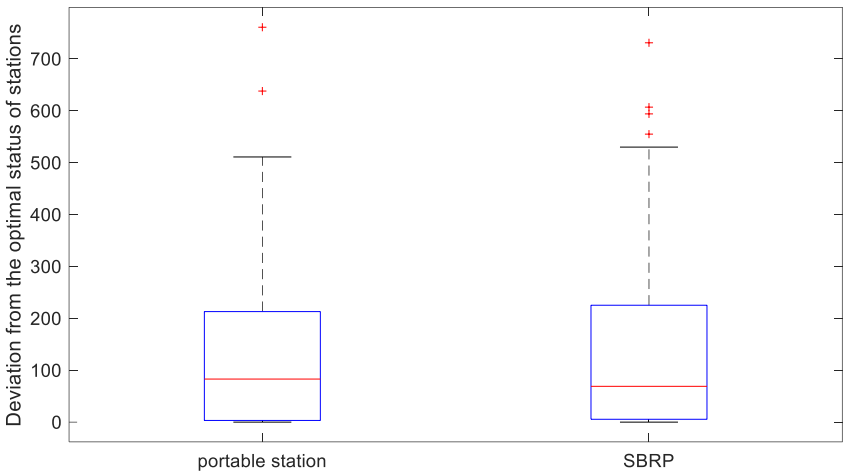


Figure 6-6 Box plot of the average deviation from the optimal status for the two approaches: portable stations and SBRP

To test the significance of the results, a permutation test (resampling test) was utilized. This test draws randomly from a set of the datapoints with a goal of estimating the precision of a sample. The test revealed that the results of these two approaches were significantly different, with a p-value of 0.0004 (at 0.05 level).

In exploring the portable station's path, we observed the following:

1. Although the path changed significantly with each starting station, this does not seem to play a role in the final imbalance results given the high degree of connectivity between stations.
2. The portable station was more likely to leave its location each hour, suggesting that the length of stay might be reduced to be a variable instead of constant.
3. The majority of the imbalanced conditions for all stations occurred during the second half of the day (1:00–8:00 p.m.) (Figure 6-7), suggesting that portable stations should be deployed at certain times of day instead of throughout the entire day
4. Four stations carried almost 50% of the missed pick-ups (Figure 6-8, circled in red). This can be explained by the fact that these four stations are close to either a public transportation service or a hub for other bike stations, indicating that the downtown area should be divided into four areas, with each area having its own designated portable station.

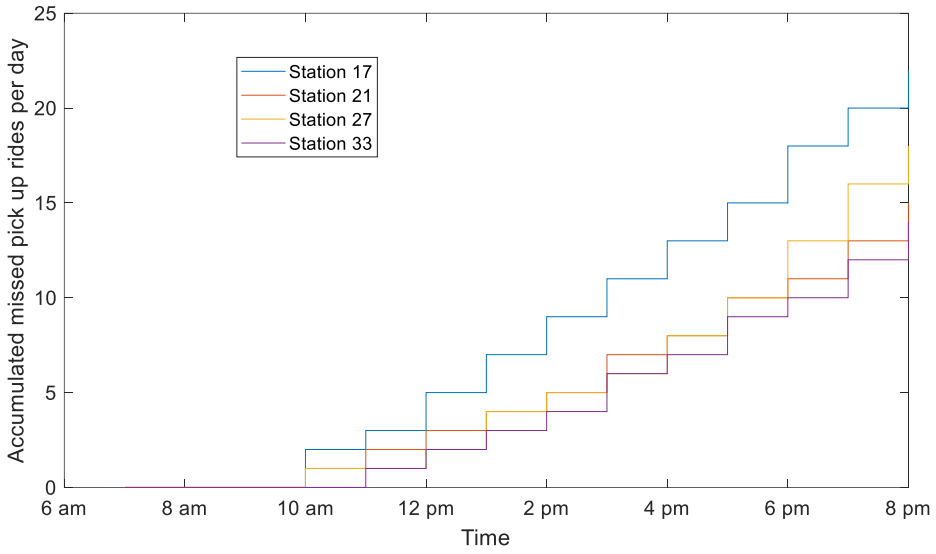


Figure 6-7 Average accumulated missed bike pick-ups per day when using portable station

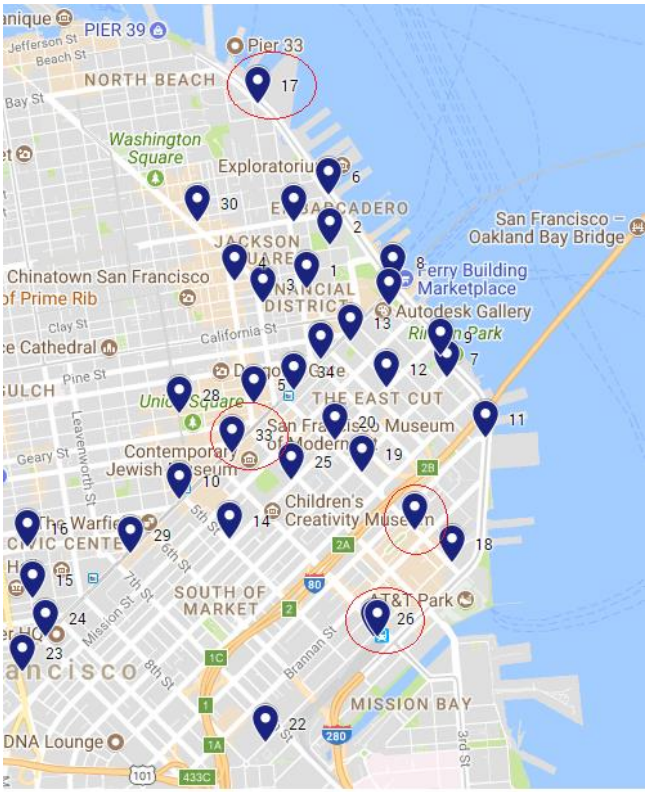


Figure 6-8 Four stations circled in red had 50% of missed bike pick-ups

To consider other simulation scenarios in terms of the size and the number of portable stations, we conducted a sensitivity analysis with respect to both customer satisfaction (represented by missed bike pick-ups) and imbalanced operation (represented by deviation from the optimal status). First, we investigated the effect of the size of the portable station on the reduction of missed bike pick-up (Figure 6-9). The reduction of the missed pick-up increased two times when the number of

bikes at the portable station increased from 20 to 30. Second, we analyzed the impact on both customer satisfaction and repositioning operation of increasing the number of portable stations from one to two (Figure 6-10). Adding one more portable station increased the percentage of the reduction in missed pick-ups to almost 25%. The sensitivity analysis also showed that adding a portable station could increase the reduction in the deviation from the stations' optimal status as much as three times.

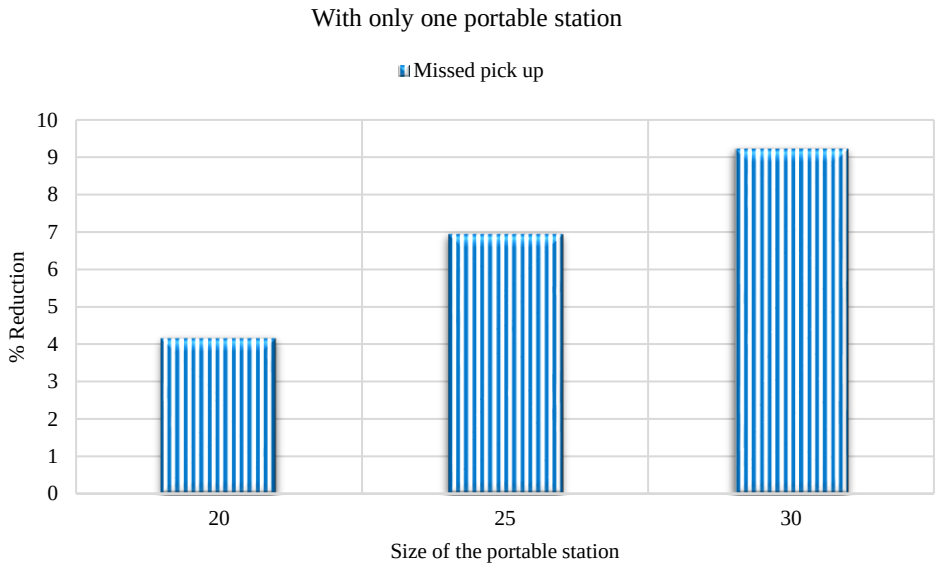


Figure 6-9 The effect of the size of the portable station on the missed bike pick-ups

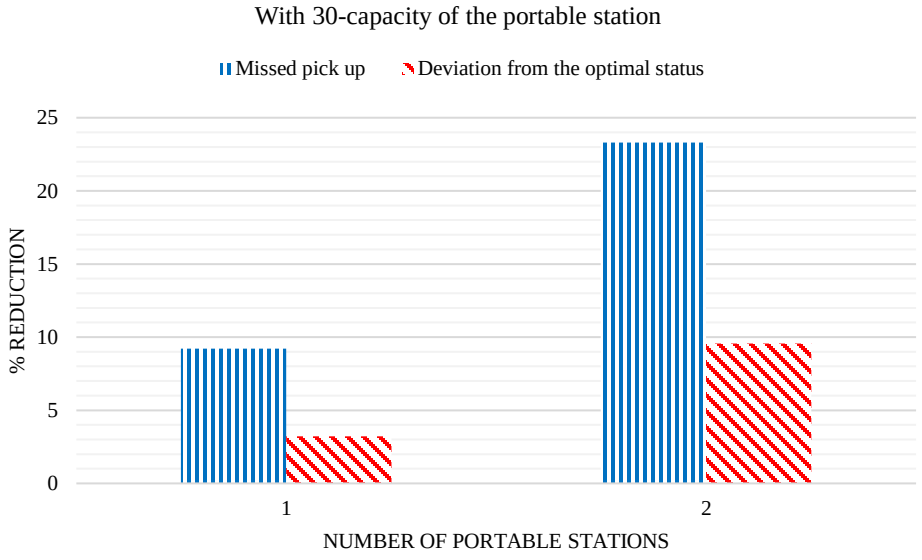


Figure 6-10 The effect of the number of portable stations on missed bike pick-ups and deviation from stations' optimal status

Along with improving both user satisfaction and imbalanced operation with the addition of only one or two portable stations, this approach addresses shortcomings of the other two main

rebalancing approaches (static and dynamic). First, the portable station does not have to spend time looking for a parking sport. Second, no loading or unloading bikes of is needed given that bikers would be able to drop off/pick-up the bike directly from the portable station without any assistance from operating agency personnel. Third, while the other two approaches require a depot for the service truck, this approach does not, as the portable station could be part of the BSS. Fourth, it makes the best use of the BSS's stations by moving the low-demand stations to high-demand areas, capturing the dynamic growth of the city without changing the infrastructure.

We believe the results of this approach could be improved significantly if assuming the portable stations can be standalone, making the length of stay a variable, and by developing an optimal technique to give high priority to stations in need of urgent-help when moving the portable stations.

6.6 Conclusions

Bike sharing systems (BSSs) have expanded rapidly worldwide due to their significant benefits to environmental, transportation, and health sectors. Yet logistical issues threaten BSSs' ability to continue growing and maintain their customers. If users cannot rent or drop off a bike because the station is either empty or full they will be less likely to participate in a BSS. Previous studies have investigated two main approaches to rebalancing—static and dynamic—both of which assume all stations are fixed. In this chapter, we investigated the advantage of having portable bike stations, using an agent-based simulation approach as a proof-of-concept. We used data from a period covering 2 years of BSS operation in the San Francisco Bay Area. Results revealed that adding one portable station could decrease the missed pick-ups by approximately 10%, leading to enhanced customer satisfaction and operation repositioning. Sensitivity analysis showed that adding one more portable station could increase the percentage in the reduction of missed pick-ups to almost 25%. Finally, the obtained results showed that adding one portable station could increase the reduction in the deviation from the optimal status of stations as much as three times.

In the future, we will enhance the proposed rebalancing approach by

1. Investigating the possibility and advantage for making portable stations' length of stay variable instead of constant.
2. Developing an optimal way to move the portable stations instead of using the greedy approach, enhancing both repositioning imbalance and customer satisfaction.
3. Considering spatial and temporal clustering techniques when assigning and timing portable stations.
4. Using the Markov Chain process to identify optimal bike counts at the start of the day instead of assuming that stations are at half capacity.
5. Adapting a predictive model to anticipate instantaneous demand.
6. Analyzing downtown San Francisco's socioeconomic information to estimate the number of trips at each point to determine if the portable station can operate as a standalone station.

CHAPTER 7 A NEW MATHEMATICAL APPROACH TO SOLVE BIKE SHARE SYSTEM STATION IMBALANCES BASED ON PORTABLE STATIONS

7.1 Introduction

Large urban areas are often associated with traffic congestion, high carbon mono/dioxide emissions (CO/CO2), fuel waste, and associated decreases in productivity. The estimated loss attributed to missed productivity and wasted fuel increased from $87.2 to $115 between 2007 and 2009 (*33*). Driving in congested areas also results in long trip times. For instance, in 1993, drivers experienced trips that were 1.2 min/km longer in congested conditions (*34*).

As a result, commuters are encouraged to leave their cars at home and use public transportation modes instead. However, public transportation modes fails to deliver commuters to their exact destination. Users have to walk some distance, which is commonly called the "last mile"(*104*). Bike sharing systems (BSSs) have started to fill this gap, offering a flexible and convenient transportation mode for commuters, around the clock. This is in addition to individual financial savings, health benefits, and reduction in congestion and emissions. Resent reports have shown BSSs multiplying over 50 countries (*26*). For instance, in the U.S., over 108 cities have implemented BSSs; there are 3,378 BSSs across the US according to a report published by the Bureau of Transportation (*105*).

This notable expansion of BSSs also brings daily logistical challenges due to the imbalanced spatial-temporal demand, causing some stations to run empty while others become full. Rebalancing the bike inventory in a BSS is crucial to ensure customer satisfaction and the whole system's effectiveness (*32*). Most of the operating costs are also associated with rebalancing. Two main approaches have been thoroughly investigated in the literature: the static bicycle repositioning problem (SBRP) and the dynamic bicycle repositioning problem (DBRP).

Both approaches involve deploying a fleet of trucks to move bikes between stations either (1) overnight for SBRP or (2) during the day for DBRP. Both approaches assume stations are fixed and thus don't take into account that the demand changes from weekday to weekend as well as from peak to non-peak hours, making some stations useless during specific days of the week and times of day. Furthermore, cities change continually with regard to demographics or structures and thus the distribution of trips also changes continually, leading to re-installation of stations to accommodate the dynamic change, which is both impractical and costly.

An alternative approach was proposed by Saltzman and Bradford; they suggested finding the most imbalanced stations and then changing their infrastructure by either increasing the empty or full racks during a one-time adjustment (*38*). However, this approach would be costly and an impermanent solution, as demand changes over time.

In this paper, we propose a new generation of BSS in which we assume some stations are portable, meaning they can move during the day. They can be either stand-alone or an extension of existing stations with the goal of accommodating the dynamic changes in the distribution of trips during the day. To investigate the advantage of having portable bike stations, we developed a simulation environment and assigned the portable stations using both greedy and stable marriage approaches. We also modeled the portable stations mathematically as a linear programming problem.

The proposed BSS was applied to ~2 years (2013–2015) of a BSS data set from the San Francisco Bay Area. Our results show that adding two portable stations to a 35-station network led to > 20% reduction in missed pickups and >10% in deviation from stations' optimal status in the whole network.

7.2 Related Work

There are three major approaches to bike redistribution: static (SBRP), dynamic (DBRP), or incentivized. For both SBRP and DBRP approaches, operating agencies move bikes between stations using service vehicles with a goal of keeping the system balanced. SBRP neglects bikes' movement during the redistribution process and is usually done at night. DBRP takes bikes' movement into consideration during the redistribution process and thus can be done at any time during the day. Incentivized rebalancing encourages users to contribute to system rebalancing by sending them suggestions to change their destination slightly in favor of keeping the system balanced.

For the SBRP approach, researchers have typically adapted models and proposed solutions to solve the imbalance as a night problem (*29, 30, 52*). Espegren et al. proposed an optimization model aiming to minimize the optimality deviation for stations (*29*). The authors used a heterogeneous fleet of service vehicles to move bikes between stations that allowed for multiple visits and arrived at a non-perfect rebalancing solution. Caggiani and Ottomanelli proposed a modular decision support system to rebalance the bike distribution by determining the best bike repositioning and the best relocation time horizon with the optimal carrier vehicle route (*30*). The objective function was minimizing both deviation and the cost of repositioning using carrier vehicles. Chemla et al. used a single truck to rebalance the BSS using a branch-and-cut algorithm (*52*). The proposed algorithm aimed to minimize the distance traveled by the truck.

The DBRP approach was shown to be more realistic and more powerful given that it considers the movement of bikes during the rebalancing operation and also allows for rebalancing the system during the day (*10, 26, 28, 54*). Brinkmann et al. attempted to solve the imbalance in a BSS by formulating the problem as a stochastic inventor routing problem (*31*). The objective function was to minimize the number of times the bike station either runs out of bikes or reaches its return limit. The authors proposed a short-term strategy that determines which stations are more likely to be out of service. This strategy and the long-term relocating strategies were used to compromise between the number of served stations and relocation operations with the goal of minimizing the number of times a station runs out of bike or reaches its limit. Contardo et. al proposed a dynamic approach to address the imbalance in a BSS using Danzig-Wolf and Benders decomposition (*28*). A scalable methodology to provide lower and upper bounds was developed. The objective function was set to minimize the total unmet demand (i.e., deviation). Regue and Recker proposed a framework to solve the repositioning problem using proactive dynamic vehicle routing along with machine leaning techniques. Four models were proposed: a demand for casting model, a station inventory model, a redistribution needs model, and a vehicle-routing model (*10*). First, they determined the future station inventory levels using different datasets. Then, they formulated the problem as a stochastic linear integer problem to estimate the number of needed bikes at each station at a given time. After that, pickup and drop off activities were determined and used as an input to the vehicle routing problem. It should be noted here that the output of the pickup and drop off are deterministic. Ghosh et al. used a mixed integer linear programming approach to rebalance a BBS (*26*). The objective function of the problem was maximizing the served demand and minimizing the cost caused by using balancing carrier vehicles. Then, they solved the problem using two steps: finding the repositioning solution for bikes and finding the routing solution for carrier vehicles. To simplify the problem and speed up the solution process, they clustered the bike

stations with respect to their location. BSS data from Washington, DC and Boston, MA were used, and a significant improvement in the operational efficiency was shown.

A recently introduced type of BSS allows for operation of the system without stations, or dock-less operation. Bikes can be rented and returned from/to anywhere (usually within a predefined zone). This is in an attempt to overcome the rebalancing problem and also to give more freedom of mobility for users. Although this has partially solved the rebalancing problem in that there is no longer a drop-off problem, it significantly affects the pick-up side of the problem. Over time, bikes end up being distributed sparsely throughout the city, making it difficult for users to find a bike. This reduces system efficiency and thus customer satisfaction.

In these research efforts, we propose a smart BSS with portable bike stations. The capacity of stations is adjustable over time, accommodating for dynamic change during the day. The proposed BSS has the potential to reduce rebalancing costs and increase system efficiency.

7.3 Methodology

To quantify the advantage of the portable stations, we developed a BSS simulator using MATLAB. Using 2-year historical data, we calculated the arrival rate of users for each station, assuming it follows a Poisson process. Bikers' travel times were assumed to be a Gamma distribution. Bikers determined their destination based on multinomial probabilities that were calculated from the 2-year historical data. We assumed users turned away and used another transportation mode if the station was empty and would try to find another nearby station to drop off their bike if the first station was full.

Portable stations were set to start the day at half capacity. That was to give an equal probability of the station being either empty or full. The portable stations started randomly from any fixed station at the beginning of the day and then made a decision every hour either to move or stay based on two approaches: greedy and stable marriage (explained following). The goal of the portable stations was to try to keep every station at a half capacity level. We assumed each portable station's capacity and speed to be 20 bikes and 15 mph respectively. Simulation was conducted every deci-second. The two approaches to moving the portable stations are explained in the following subsections.

7.3.1 Greedy Approach

The portable station chooses the next station in a greedy way based on (1) and (2) as follows:

$$R_{t+1} = S_t - \lambda_{t+1} + {P_{t+1}^T}^T \lambda_{t+1} \quad (1)$$

$$\min_i \arg\left(\left|\frac{Q_j}{2}\right| - H_j + r_i\right) \quad (2)$$

Where

S_t is a column vector where each element i is current available bikes at station i

λ_{t+1} is a column vector where each element i is the arrival rate of bikers per hour at station i at time t+1

P_{t+1} is the transition matrix at time t+1

Q_j is the capacity of portable station j

H_j is the number of bikes loaded on the portable station at time t

r_i is the i^{th} element of the R_{t+1}

Equation (1) predicts the number of bikes at each station at time $t+1$ when the initial number of bikes at time t is S_t. As shown in (2), the portable station prefers the station that will keeps it loaded at half capacity after it visits the station.

7.3.2 Stable Marriage Approach

In 1962, Gale and Shapley proposed the deferred acceptance algorithm as a solution to the stable marriage problem, in which an equal number of men and women are matched such that no one has an incentive to leave his/her matched partner [15]. The stable marriage problem involves one-to-one matching. Each man has a ranked list of women they prefer, and each woman has a ranked list of men they prefer. The best-qualified candidates get engaged first, followed by the lesser-qualified candidates.

The stable marriage algorithm finds a stable matching solution through a series of iterations. At each iteration, the men propose to the best-qualified women, and the women have to reply back by either accepting the offer or not. At the end of the iteration, some men get engaged and others do not. Men then update their list accordingly in the next iteration and offer a proposal to women who did not get engaged in the previous iterations, regardless of whether they have been accepted or not. The men's lists do not change, but women can change their decision at each iteration if they are offered a better proposal. The algorithm keeps iterating until reaching a stable matching solution.

This algorithm was adapted so that it matches between the portable and fixed stations at each iteration. The portable stations search for the fixed stations that are close to it, while the fixed stations search for the portable stations that fulfill its needs (the availability of racks or bikes).

7.3.3 Optimization Mathematical Model

The mathematical optimization approach can provide optimal solutions for many real cases. However, this approach is not the ideal approach for solving large scale problems. In this paper, the mixed integer programming model is developed to minimize the total associated costs for the drop-off and pick-up of bikes at a bike station. As the portable station bike sharing problem is defined as an NP-hard problem, future work will involve integrating the proposed mathematical model with simulation approaches to improve the findings presented here. The future proposed simulation-optimization approach depends on using the findings from the current simulation approach developed in this paper as inputs (e.g., arrival and departure times of bikes) for the mathematical optimization model.

Indices

B: Number of bikes in the system

b: Index of bike, $b = 1,..B$

K: Number of bike stations

k: Index of bike station; $k = 1, \dots, K$

γ_k : Mean number of bikes drop-off per unit time period at station $k;\ k = 1, \dots, K$

μ_k :Mean number of bikes pick up per unit period at station $k;\ k = 1, \dots, K$

M_k: Number of service facilities or rakes for each station ; $k = 1, \dots, K$

p_{k0} : Probability of 0 bikes in the system at station k; $k = 1, \dots, K$

p_{kn} : Probability of n bikes in the system at station k; $k = 1, \dots, K$; $1 \leq n \leq B$

w_k : Average time a unit spends in the queue at station k; $k = 1, \dots, K$

W_k : Average time units spend in the system at station k; $k = 1, \dots, K$

l_k : Average number of bikes waiting in the queue at station k; $k = 1, \dots, K$

L_k : Average number of bikes spends in the system at station k; $k = 1, \dots, K$

t_k : Penalty cost for each bike waiting in queue at station k; $k = 1, \dots, K$

i_k: Penalty cost for each unit time spend waiting in queue at station k; $k = 1, \dots, K$

c_{bk}: The costs of each km distance between portable station b and fixed station $k; k = 1, \dots, K \; and \; b = 1, \dots, B$

Parameters

d_{bk}: Distance between portable station b and fixed station k; $b = 1 \dots B$ and $k = 1 \dots K$.

D_k : Total demand of station k, number of bikes that can be used daily; $k = 1 \dots K$.

Q_k: The extended capacity for fixed station k by adding a portable station ; $k = 1 \dots K$.

q_k: The initial capacity of fixed station k, the initial number of racks; $k = 1 \dots K$.

b_k: The cost of adding a portable station at fixed station k; $k = 1 \dots K$.

V: Total investment for adding all portable stations to the fixed stations.

f: Total associated costs for the given system

Decision Variables:

$$z_{bk} = \left\{ \begin{array}{l} 1: \text{Bike } b \text{ has been assigned to station } k; b = 1 \dots B \text{ and } k = 1 \dots K. \\ 0: otherwise \end{array} \right\}$$

$$y_{kk} = \left\{ \begin{array}{l} 1: \text{station } k \; is \; open \; and \; ready \; to \; be \; used; k = 1 \dots K. \\ 0: otherwise \end{array} \right\}$$

Objective function:

The objective function in (3) is to minimize the total associated costs for the drop-off and pick-up of bikes at a bike station. The developed equation demonstrates three terms of costs: the costs for

distances between portable and fixed station, the penalty costs of waiting time and waiting bikes in the queue, and the costs of adding new portable stations.

$$f = Min(\sum_{b=1}^{B}\sum_{k=1}^{K} c_{bk}d_{bk}\, z_{bk} + \sum_{k=1}^{K}(l_k t_k + i_k w_k)y_{kk} + \sum_{k=1}^{K} b_k Q_k y_{kk}) \quad (3)$$

Constraints (4) impose that each bike b is assigned to exactly one station k;

$$\sum_{k=1}^{K} z_{bk} = 1 \qquad \forall\, b \in B \quad (4)$$

Constraints (5) ensure that no bike b is assigned to a station k unless a station is open at that location.

$$z_{bk} \le Y_{kk} \qquad \forall\, b \in B, k \in K, \quad (5)$$

Constraints (6) ensure that total number of assigned bikes to station k is less than or equal the total capacity of this station. The station capacity demonstrates the initial capacity q_k and the extended capacity Q_k. In this set of constraints, the station capacity means the number of racks in this station.

$$\sum_{b=1}^{B} z_{bk} \le (q_k + Q_k)y_{kk} \quad \forall\ k \in K \quad (6)$$

Constraints (7) ensure that the total costs of adding S portable stations are less than the total investment V, where the cost of each portable station at fixed station k is b_k*;i.e.* parking costs are considered.

$$\sum_{k=1}^{K} b_k\, Q_k \le V \quad (7)$$

Constraints (8) ensure the assigned number of bikes to open station k is less than the expected demand for this station. Following, the expected demand will be found based on the historical data.

$$\sum_{b=1}^{B} z_{bk} \le D_k\ y_{kk} \qquad \forall\ k \in K \quad (8)$$

Constraints (9) ensure that the assignment decision will result in either no assignment, $Z_{bk} = 0$, or assignment, $z_{bk} = 1$.

$$z_{bk} \in \{0, 1\} \qquad \forall b \in B, k \in K \quad (9)$$

Constraints (10) ensure that the station open decision will result in either not open, $y_{kk} = 0$, or assignment, $y_{kk} = 1$.

$$y_{kk} \in \{0, 1\} \qquad \forall\ k \in K \quad (10)$$

In (11) the extended capacity is less than or equal zero.

$$Q_k \ge 0 \qquad \forall\, k \in S \quad (11)$$

In (12), the probability of 0 bikes in the system for each station k is calculated as follow:

$$p_{k0} = \left(\frac{1}{\sum_{n_k=0}^{M_k-1} \frac{1}{n_k!}\left(\frac{\gamma_k}{\mu_k}^{n_k}\right) + \frac{1}{M_k!}\left(\frac{\gamma_k}{\mu_k}^{M_k}\right)\left(\frac{M_k\mu_k}{M_k\mu_k - \gamma_k}\right)} \right) y_{kk} \qquad 1 \le n_k \le B; \quad (12)$$

Equation (13) calculates the average number of bikes waiting in a queue and being served (parked) for each station k as follows:

$$L_k = \left(\frac{\gamma_k \mu_k \left(\frac{\gamma_k}{\mu_k}\right)^{M_k}}{(M_k-1)!(M_k\mu_k-\gamma_k)^2} p_{k0} + \frac{\gamma_k}{\mu_k} \right) y_{kk} \qquad \forall\, k \tag{13}$$

Equation (14) calculates average times of bikes waiting in the queue and being served (parked) for each station k as follows:

$$W_k = \left(\frac{\mu_k \left(\frac{\gamma_k}{\mu_k}\right)^{M_k}}{(M_k-1)!(M_k\mu_k-\gamma_k)^2} p_{k0} + \frac{1}{\mu_k} \right) y_{kk} \qquad \forall\, k \tag{14}$$

Equation (15) calculates the average number of bikes waiting in queue for each station k as follows:

$$l_k = \left(L_k - \frac{\gamma_k}{\mu_k}\right) y_{kk} \qquad \forall\, k \tag{15}$$

Equation (14) calculates the average time for bikes waiting in queue for each station k as follows:

$$w_k = \left(W_k - \frac{1}{\mu_k}\right) y_{kk} \qquad \forall\, k \tag{16}$$

7.4 Data Set

This study used Ford GoBike's BSS trip dataset, containing data collected from August 2013 to August 2015 in the San Francisco Bay Area. During that period, the BSS had 35 stations covering San Francisco (Figure 7-1). The dataset contains around half a million trips, each of which includes bike ID, trip duration, trip start day and time, trip end day and time, trip start ID station, and trip end ID station. Another file called "station" was also used; this file contains geographic and operating information for each station, including latitude, longitude, capacity, city, and installation date.

During the analysis phase, we found that the demand during off-peak hours was stable (Figure 7-2), so we only considered the hours of 6:00 a.m. to 8:00 p.m. when simulating the network. That is, we assumed the level of demand at stations at the end of the day (i.e., 8:00 p.m.) was the same as at the beginning of the day (i.e., 6:00 a.m.).

During data preparation, a source-destination matrix was built to count all trips for each hour between each pair of stations, and then a probability transition matrix was created. This was done for all days of week during the period from 6:00 a.m. to 8:00 p.m. Similarly, the associated travel times with source-destination matrix were extracted from the trips dataset. Given the presence of outliers in the trip dataset (either very short or long trips), we only extracted 95% of the trips, meaning we excluded the shortest and longest 2.5% of trips.

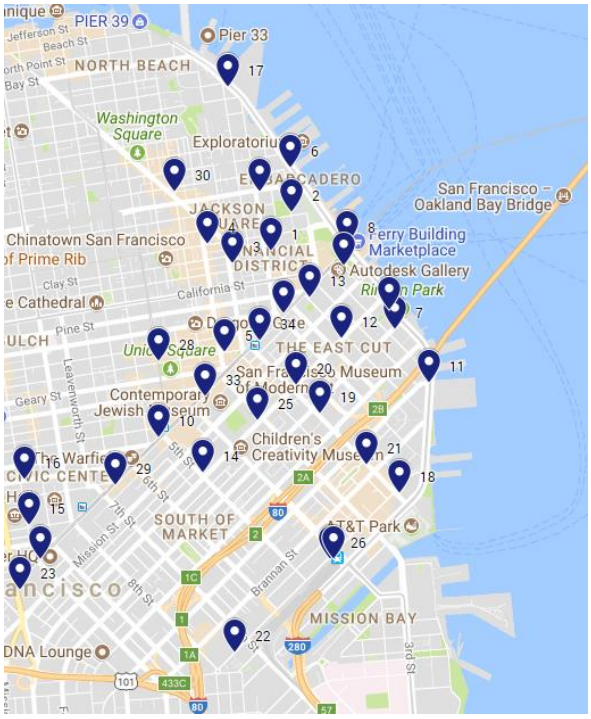


Figure 7-1 Locations of 35 bike stations in downtown San Francisco (Source: Google Maps)

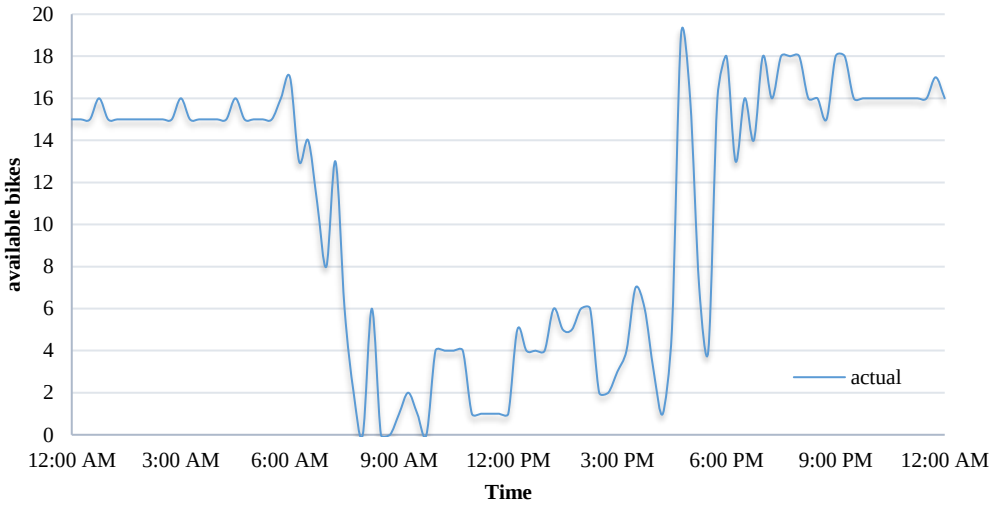


Figure 7-2 Bike counts for a station during the entire day

7.4.1 Model Testing

The simulation was applied to downtown San Francisco for 35 stations and was run 50 times using two approaches: greedy and stable marriage. Two scenarios were considered: without and with portable stations. To quantify the impact of the portable stations, we used two measurements: missed pickup rides and the stations' status at the end of the day. The missed pickups measurement counts the times that users arrived and couldn't find a bike due to the imbalance problem. The second measurement evaluates the status of the stations at the end of the day to determine how

imbalanced the system is, thus quantifying the repositioning efforts needed at the end of the day for the whole network (Equation.3).

$$\sum_{i=1}^{35} \left| B_i^{start\ of\ the\ day} - B_i^{end\ of\ the\ day} \right| \quad (17)$$

Where B_i^t is the number of available bikes at station i at time t.

7.4.2 Results and Discussion

During the simulation, portable stations started the day at half capacity and then made a decision every hour to either stay at the current location or move to another station. Two approaches were used for the movement of the portable station: greedy and stable marriage. The tuning parameters were the size and number of the portable stations.

Using only one portable station of 20 bikes with the greedy approach led to a reduction of 7% in missed pickups and 2% in deviation from stations' optimal status in the whole network (Figure 7-3). Increasing the number of portable stations from 1 to 2 significantly improved the network and resulted in reduction of over 20% in missed pickups and over 10% in deviation.

As for the stable marriage approach, given that it is a matching problem, at least two portable stations are required. The results show that using two portable stations with 20 bikes each could reduce missed pickups and deviation from stations' optimal status by around 12% and 4% respectively. Increasing the number of portable stations from 2 to 3 would further reduce missed pickups and deviation, albeit slightly, from ~10% to ~15% and ~4% to ~15%, respectively, as Figure 7-3 shows. This is because the BSS network is dense and small. With regard to the reduction in missed pickups at the day-of-week level, Saturdays and Sundays had the highest reduction of >70% in missed pickups. When comparing the computational time for the approaches, we found the stable approach to be slightly faster (15%) than the greedy approach.

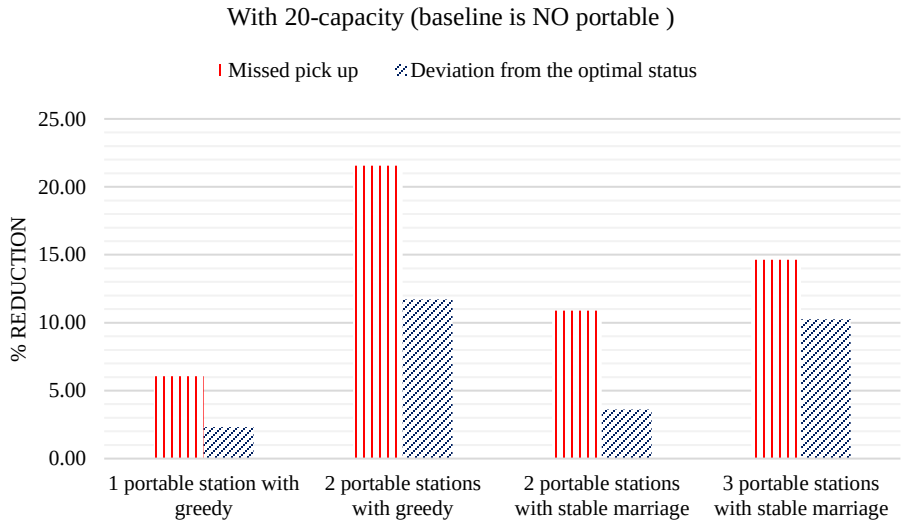


Figure 7-3 The results of two greedy approaches

We investigated the size of the portable stations and found that it has no significant impact on either the missed pickups or the deviation from the stations' optimal status. This is due to the fact the average size of the bike stations in the network was 20 bikes.

We took a closer look at the most imbalanced stations—numbers 1, 2, and 26—and investigated

the reduction in missed pickup rides using three different scenarios as shown in Figure 7-4. Generally, the second and third scenario did better than the first, as the results show.

We studied station 26 (located at San Francisco Caltrain–Townside at 4th), which suffers from high demand and is therefore more likely to be imbalanced, especially on weekdays, as it is located next to a metro station. We found that missed pickups could be reduced by up to 27% using 3 portable stations with 20 bikes on all days of the week. Investigating the days of the week revealed that missed pickups on Mondays could be reduced the most, by around 40%, followed by Wednesday.

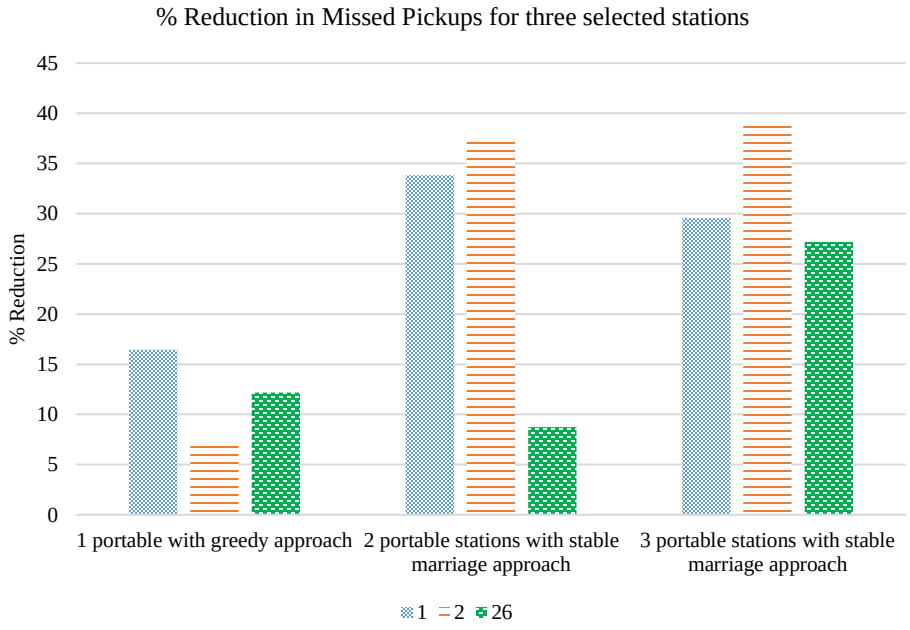


Figure 7-4 The percentage of reduction in missed pickups for three selected stations in the network using three different approaches

7.5 Conclusions

This paper proposes a new generation of BSS where some bike stations are movable with the goal of minimizing imbalances with less cost and effort. We proposed using two approaches to move the portable stations: greedy and stable marriage. Additionally, a mathematical formulation was developed.

The adopted approach was tested on a BSS in the downtown San Francisco area using the two aforementioned approaches. The results show that adding two portable stations to a 35-bike BSS network using a greedy approach led to a reduction of over 20% in missed pickup rides and over 10% in the deviation from optimal status. Increasing the size of the portable stations from 20 to 30 did not improve the results significantly. Generally, weekends seemed to have the highest reduction rate in terms of missed pickups and deviation from the optimal status.

Additionally, results show the greedy approach outperforms the stable marriage approach in terms of reduction, though it is slightly slower computationally.

CHAPTER 8 CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER RESEARCH

This chapter summarizes the objectives and findings of the dissertation, followed by recommendations for future research.

8.1 Dissertation Conclusions

Many cities have realized the negative effects of the increasing number of vehicles on the roads, such as greater congestion, emissions, and pollution rates. In response, various cities have discussed methods to reduce these rates. For example, in Oslo, leaders have decided to completely ban privately-owned cars from its center by the end of 2019, making it the first European city to totally ban cars in the city center. Instead, public transit and cycling will be supported and encouraged in the banned-car zone, and hundreds of parking spaces in the city will be replaced by bike lanes. Also, in South Korea, a massive, first-of-its-kind 100 million square foot city is being designed to reduce or even eliminate the need for cars and instead allow only bikes and public transportation modes. At a cost of $35 billion, completion of this district is expected by 2020.

BSSs have also been shown to be an energy-efficient and reliable transportation mode, and have been introduced in 1,139 cities and over 50 countries. In the San Francisco Bay Area, Saltzman and Bradford found that 92% of all weekday trips using BSSs were made by daily commuters going to and from work, showing significant faith in the BSSs' reliability (*38*). According to the National Association of City Transportation Officials, in the U.S, in 2016 alone there were over 28 million bike trips, an increase of 25% compared to 2015. This increased usage of bikes led many cities to either expand their existing system or launch a new one. For example, Ford, the operator of the GoBike BSS in the San Francisco Bay Area, started the system in 2013 with 700 bikes and 70 stations, and now plans to expand their system to 7,000 bikes and over 300 stations by the end of 2018.

Due to the unbalanced spatial-temporal demand of bike trips, many bike stations become empty or full during the day. This significantly affects the reliability and usefulness of the BSS, which may prompt riders to return to using their personal cars or to adopt another transportation mode, consequently increasing congestion and thus auto emissions and pollution. This in turn, would lead to a decrease in the number of BSS users, reducing the system's revenue. Operating agencies have recognized the imbalance issue and have started to establish more bike stations close to one another, aiming to keep them within no more than a 5-minute walk. However, this solution is difficult to implement, both financially and practically.

This dissertation proposes a new generation of BSSs in which some bike stations can be portable. This approach takes advantage of both types of BSS: dock-based and dock-less. Towards this goal, a BSS optimization framework was developed at both the tactical and operational level. Specifically, the framework consists of two levels: predicting bike counts at stations using fast, online, and incremental learning approaches and then balancing the system using portable stations. The goal is a framework for solving the dynamic bike sharing repositioning problem in order to minimize unmet demand, leading to increased user satisfaction and reduced repositioning/rebalancing operations.

This dissertation contributes to the field in five ways. First, a multi-objective supervised clustering algorithm was developed to identify the similarity of bike-usage with respect to time events. Second, a dynamic, easy-to-interpret, rapid approach to predict bike counts at stations in a BSS was developed. Third, a univariate inventory model using a Markov chain process that provides an optimal range of bike levels at stations was created. Fourth was an investigation of the advantages of portable bike stations, using an agent-based simulation approach as a proof-of-concept. Fifth, mathematical and heuristic approaches were proposed.

The research findings are presented as follows:

- Chapter 3 develops a new supervised clustering algorithm to potentially assist agencies and researchers in anticipating bike availability at stations with respect to a time event. The proposed algorithm was tested on a BSS in the San Francisco Bay Area. The proposed algorithm clusters bike availability data at 15-minute intervals across the network and finds the similarity between them according to day of the week and hour of the day. Subsequently, it provides an expected pattern of bike usage for each cluster. The algorithm provides insight into the usage patterns of the San Francisco Bay BSS that operators can use to anticipate imbalances in the system and plan accordingly. Moreover, the clustering results show that the days of the week can be grouped into three clusters, one for weekends and the other two for weekdays. The time of day is clustered into two groups, peak and off-peak hours. Given that each cluster has an associated pattern of bike availability, a prediction can be made to identify the imbalance in the system for each day of the week and each hour of the day. An exploratory spatiotemporal analysis was conducted, leading to different suggestions on how to rebalance the system with minimum cost and effort, thus making the network a more-effective component of the smart transportation system in a smart city.
- Chapter 4 proposes a Markov chain model for each station and day of the week. The daily imbalances were examined and the optimal inventory level that minimizes the probability of a station reaching an empty or full state was identified. The analysis shows that the optimal initial conditions vary from one day of the week to another for the same station, and thus the optimal initial conditions for each day of the week are presented. The results show that San Francisco has the highest percentage of category "Imbalance probability >25% for >45% of the initial conditions," followed by San Jose. This demonstrates that San Francisco BSSs experience high bike demands, and thus are more likely to have an imbalance problem during the day. The proposed approach would be less effective for the San Francisco BSS and more effective for the other cities given that the daily evolution of states for San Francisco varies considerably.
- Chapter 5 proposes bike count prediction models using state-of-the-art machine learning and statistical algorithms. Dynamic linear and incremental learning models were adapted with a goal of finding a good model in terms of both prediction accuracy and computational time without any other external variables such as weather or spatiotemporal information. The results of the online and incremental learning algorithms were compared with the machine learning algorithms: Random forest (RF) and LSboosting (LSBoost). The results show that all algorithms returned a comparable prediction accuracy under 15-minute and 30-minute prediction windows, with the exception of LSBoost. For the rest of the prediction windows, RF outperformed all other algorithms. However, when comparing the computational time for the five algorithms, RF had the largest running time, followed by

Locally Weighted Regression (LWR). The Mini-batch Gradient Descent for Linear Regression (MBGDLR) algorithm had the smallest computational time, followed by Dynamic Linear Models (DLM). Although RF gives the smallest prediction accuracy, it takes longer to predict (77 times longer than MBGDLR and 12 times longer than DLM). Based on the previous comparison considering both prediction accuracy and computational time, MBGDLR is better than the rest of the algorithms due to its ability to predict with a relatively small prediction error in a very short time.

- Chapter 6 investigates the advantage of having portable bike stations, using an agent-based simulation approach as a proof-of-concept. The results based on only Mondays revealed that adding one portable station could decrease the missed pick-ups by approximately 10%, leading to enhanced customer satisfaction and operation repositioning. Sensitivity analysis showed that adding one more portable station could increase the percentage in the reduction of missed pick-ups to almost 25%. Finally, the obtained results showed that adding one portable station could increase the reduction in the deviation from the optimal status of stations as much as three times.
- Chapter 7 formulates the movement of portable stations using two approaches: greedy and stable marriage. Additionally, a mathematical formulation was developed. The results show that adding two portable station to a 35-bike-station network with the greedy approach led to a reduction over 20% in the missed pickup rides and over 10% in the deviation from optimal status. Increasing the size of the portable station from 20 to 30 would not improve the results significantly. Generally, weekends seem to have the highest reduction in terms of missed pickups and deviation from the optimal status. Additionally, results show the greedy approach outperforms the stable marriage approach in terms of the reduction, though it is slightly slower computationally.

8.2 Recommendations for Further Research

To extend and enhance the research presented in this dissertation, the following actions are recommended:

- Incorporating more predictors in the dynamic and incremental learning models, such as weather information, seasonality, and availability and location of the other public transportation modes (bus or metro) and their schedules. In addition, it might be worthwhile to investigate the benefit of clustering the months or days, and then adapting the dynamic and incremental learning models for each cluster
- Investigating the possibility and advantage of making portable stations' length of stay variable instead of constant.
- Considering spatial and temporal clustering techniques when assigning and timing portable stations.
- Analyzing downtown San Francisco's socioeconomic information to estimate the number of trips at each point to determine if a portable station can operate as a standalone station.
- Using other BSS datasets that are more complex and dense.

BIBLIOGRAPHY

1. DeMaio, P., *Bike-sharing: History, impacts, models of provision, and future.* Journal of public transportation, 2009. **12**(4): p. 3.
2. *Proposals to ban cars and taxis from Dublin city centre*. 2018; Available from: https://www.joe.ie/news/proposals-to-ban-cars-and-taxis-from-dublin-city-centre-499064.
3. McNeil, N., et al., *Breaking Barriers to Bike Share: Insights from Bike Share Users.* 2017.
4. Qiu, L.-Y. and L.-Y. He, *Bike Sharing and the Economy, the Environment, and Health-Related Externalities.* Sustainability, 2018. **10**(4): p. 1145.
5. Schuijbroek, J., R. Hampshire, and W.-J. van Hoeve, *Inventory rebalancing and vehicle routing in bike sharing systems.* 2013.
6. Alvarez-Valdes, R., et al., *Optimizing the level of service quality of a bike-sharing system.* Omega, 2016. **62**: p. 163-175.
7. Pfrommer, J., et al., *Dynamic vehicle redistribution and online price incentives in shared mobility systems.* IEEE Transactions on Intelligent Transportation Systems, 2014. **15**(4): p. 1567-1578.
8. Contardo, C., C. Morency, and L.-M. Rousseau, *Balancing a dynamic public bike-sharing system*. Vol. 4. 2012: Cirrelt Montreal.
9. Angeloudis, P., J. Hu, and M.G. Bell, *A strategic repositioning algorithm for bicycle-sharing schemes.* Transportmetrica A: Transport Science, 2014. **10**(8): p. 759-774.
10. Regue, R. and W. Recker, *Proactive vehicle routing with inferred demand to solve the bikesharing rebalancing problem.* Transportation Research Part E: Logistics and Transportation Review, 2014. **72**: p. 192-209.
11. Rixey, R., *Station-level forecasting of bikesharing ridership: station network effects in three US systems.* Transportation Research Record: Journal of the Transportation Research Board, 2013(2387): p. 46-55.
12. Wang, X., et al., *Modeling bike share station activity: Effects of nearby businesses and jobs on trips to and from stations.* Journal of Urban Planning and Development, 2015. **142**(1): p. 04015001.
13. Froehlich, J., J. Neumann, and N. Oliver. *Sensing and Predicting the Pulse of the City through Shared Bicycling*. in *IJCAI*. 2009.
14. Kaltenbrunner, A., et al., *Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system.* Pervasive and Mobile Computing, 2010. **6**(4): p. 455-466.
15. Ashqar, H.I., et al. *Modeling bike availability in a bike-sharing system using machine learning*. in *Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2017 5th IEEE International Conference on*. 2017. IEEE.
16. Yang, H., et al., *Use of Deep Learning to Predict Daily Usage of Bike Sharing Systems*. 2018.
17. Yoon, J.W., F. Pinelli, and F. Calabrese. *Cityride: a predictive bike sharing journey advisor*. 2012. IEEE.
18. Almannaa, M.H., et al. *Network-wide bike availability clustering using the college admission algorithm: A case study of San Francisco Bay area*. in *Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2017 5th IEEE International Conference on*. 2017. IEEE.
19. Cui, W., *The Effects of Urban Density on the Efficiency of Dockless Bike Sharing System-A Case Study of Beijing, China*. 2018, Arizona State University.
20. Etienne, C. and O. Latifa, *Model-Based Count Series Clustering for Bike Sharing System Usage Mining: A Case Study with the Vélib'System of Paris.* ACM Transactions on Intelligent Systems and Technology (TIST), 2014. **5**(3): p. 39.
21. Daddio, D.W. and N. and Mcdonald, *Maximizing bicycle sharing: an empirical analysis of capital bikeshare usage.* 2012.
22. Etienne, C. and O. Latifa, *Model-Based Count Series Clustering for Bike Sharing System Usage Mining: A Case Study with the Velib' System of Paris.* ACM Trans. Intell. Syst. Technol., 2014. **5**(3): p. 1-21.

23. Singla, A., et al. *Incentivizing Users for Balancing Bike Sharing Systems*. in *AAAI*. 2015.
24. Ashqar H., et al., *Modeling bike availability in a bike-sharing system using machine learning*, in *5th IEEE International Conference on MODELS AND TECHNOLOGIES FOR INTELLIGENT TRANSPORTATION SYSTEMS*. 2017: Napoli, Italy.
25. Jerez, J.M., et al., *Missing data imputation using statistical and machine learning methods in a real breast cancer problem.* Artificial intelligence in medicine, 2010. **50**(2): p. 105-115.
26. Ghosh, S., et al., *Dynamic Repositioning to Reduce Lost Demand in Bike Sharing Systems*. Journal of Artificial Intelligence Research, 2017. **58**: p. 387-430.
27. Shaheen, S., S. Guzman, and H. Zhang, *Bikesharing in Europe, the Americas, and Asia: past, present, and future.* Transportation Research Record: Journal of the Transportation Research Board, 2010(2143): p. 159-167.
28. Kloimüllner, C., et al. *Balancing bicycle sharing systems: an approach for the dynamic case*. in *European Conference on Evolutionary Computation in Combinatorial Optimization*. 2014. Springer.
29. Espegren, H.M., et al. *The Static Bicycle Repositioning Problem-Literature Survey and New Formulation*. in *International Conference on Computational Logistics*. 2016. Springer.
30. Caggiani, L. and M. Ottomanelli, *A modular soft computing based method for vehicles repositioning in bike-sharing systems.* Procedia-Social and Behavioral Sciences, 2012. **54**: p. 675-684.
31. Brinkmann, J., M.W. Ulmer, and D.C. Mattfeld, *Inventory Routing for Bike Sharing Systems.* Transportation Research Procedia, 2016. **19**: p. 316-327.
32. Fricker, C. and N. Gast, *Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity.* EURO Journal on Transportation and Logistics, 2016. **5**(3): p. 261-291.
33. Schrank, D.L. and T.J. Lomax, *2009 urban mobility report*. Texas Transportation Institute: Texas A & M University.
34. Arnott, R. and K. Small, *The Economics of Traffic Congestion.* American Scientist, 1994. **82**(5): p. 446-455.
35. Walker, A. *How bike share is changing American cities*. 2017 [cited 2018; Available from: https://www.curbed.com/2017/3/21/15006248/bike-share-ridership-transit-safety.
36. Buck, D., et al., *Are bikeshare users different from regular cyclists? A first look at short-term users, annual members, and area cyclists in the Washington, DC, region.* Transportation Research Record: Journal of the Transportation Research Board, 2013(2387): p. 112-119.
37. Fishman, E., *Bikeshare: A review of recent literature.* Transport Reviews, 2016. **36**(1): p. 92-113.
38. Saltzman, R.M. and R.M. Bradford, *Simulating a More Efficient Bike Sharing System.* Journal of Supply Chain and Operations Management, 2016. **14**(2): p. 36.
39. Shaheen, S.A., *Public Bikesharing in North America: Early Operator and User Understanding, MTI Report 11-19.* 2012.
40. Schuijbroek, J., R.C. Hampshire, and W.J. van Hoeve, *Inventory rebalancing and vehicle routing in bike sharing systems*. European Journal of Operational Research, 2017. **257**(3): p. 992-1004.
41. Parikh, P. and S.V. Ukkusuri. *Estimation of optimal inventory levels at stations of a bicycle sharing system*. in *Transportation Research Board Annual Meeting*. 2015.
42. Raviv, T. and O. Kolka, *Optimal inventory management of a bike-sharing station.* IIE Transactions, 2013. **45**(10): p. 1077-1093.
43. Rudloff, C. and B. Lackner, *Modeling demand for bikesharing systems: neighboring stations as source for demand and reason for structural breaks.* Transportation Research Record: Journal of the Transportation Research Board, 2014(2430): p. 1-11.
44. Ashqar, H.I., et al., *Quantifying the Effect of Various Features on the Modeling of Bike Counts in a Bike-Sharing System*, in *97th Transportation Research Board Annual Meeting*. 2018: Washington DC.
45. Bar-Hillel, A., et al. *Learning distance functions using equivalence relations*. in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*. 2003.

46. Sinkkonen, J., S. Kaski, and J. Nikkilä. *Discriminative clustering: Optimal contingency tables by learning metrics*. in *European Conference on Machine Learning*. 2002. Springer.
47. Demiriz, A., K.P. Bennett, and M.J. Embrechts, *Semi-supervised clustering using genetic algorithms.* Artificial neural networks in engineering (ANNIE-99), 1999: p. 809-814.
48. Vogel, P., T. Greiser, and D.C. Mattfeld, *Understanding bike-sharing systems using data mining: Exploring activity patterns.* Procedia-Social and Behavioral Sciences, 2011. **20**: p. 514-523.
49. Saridis, G. and G. Stein, *Stochastic approximation algorithms for linear discrete-time system identification.* IEEE Transactions on Automatic Control, 1968. **13**(5): p. 515-523.
50. Froehlich, J., J. Neumann, and N. Oliver. *Sensing and Predicting the Pulse of the City through Shared Bicycling*. 2009.
51. Gallop, C., C. Tse, and J. Zhao, *A seasonal autoregressive model of Vancouver bicycle traffic using weather variables.* i-Manager's Journal on Civil Engineering, 2011. **1**(4): p. 9.
52. Chemla, D., F. Meunier, and R.W. Calvo, *Bike sharing systems: Solving the static rebalancing problem.* Discrete Optimization, 2013. **10**(2): p. 120-146.
53. Kadri, A.A., I. Kacem, and K. Labadi, *Lower and upper bounds for scheduling multiple balancing vehicles in bicycle-sharing systems.* Soft Computing, 2018: p. 1-22.
54. Brinkmann, J., M.W. Ulmer, and D.C. Mattfeld, *Short-term strategies for stochastic inventory routing in bike sharing systems.* Transportation Research Procedia, 2015. **10**: p. 364-373.
55. Chiariotti, F., et al., *A dynamic approach to rebalancing bike-sharing systems.* Sensors, 2018. **18**(2): p. 512.
56. Vogel, P. and D.C. Mattfeld. *Modeling of repositioning activities in bike-sharing systems*. in *World conference on transport research (WCTR)*. 2010.
57. Arbelaez, P., et al., *Contour detection and hierarchical image segmentation.* IEEE transactions on pattern analysis and machine intelligence, 2011. **33**(5): p. 898-916.
58. Roberts, J.A., *Profiling levels of socially responsible consumer behavior: a cluster analytic approach and its implications for marketing.* Journal of marketing Theory and practice, 1995. **3**(4): p. 97-117.
59. Ngai, E., et al., *The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature.* Decision Support Systems, 2011. **50**(3): p. 559-569.
60. Thiprungsri, S. and M.A. Vasarhelyi, *Cluster analysis for anomaly detection in accounting data: An audit approach.* 2011.
61. Weijermars, W. and E. van Berkum. *Analyzing highway flow patterns using cluster analysis*. in *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.* 2005. IEEE.
62. Elhenawy, M., H. Chen, and H.A. Rakha, *Dynamic travel time prediction using data clustering and genetic programming.* Transportation Research Part C: Emerging Technologies, 2014. **42**: p. 82-98.
63. Calafate, C.T., et al., *Traffic Management as a Service: The Traffic Flow Pattern Classification Problem.* Mathematical Problems in Engineering, 2015. **2015**: p. 14.
64. Eick, C.F., N. Zeidat, and Z. Zhao. *Supervised clustering-algorithms and benefits*. in *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*. 2004. IEEE.
65. Spinelli, V., *Supervised box clustering.* Advances in Data Analysis and Classification, 2017. **11**(1): p. 179-204.
66. Gale, D. and L.S. Shapley, *College admissions and the stability of marriage.* The American Mathematical Monthly, 1962. **69**(1): p. 9-15.
67. Elhenawy, M. and H.A. Rakha, *Automatic Congestion Identification with Two-Component Mixture Models.* Transportation Research Record: Journal of the Transportation Research Board, 2015. **2489**: p. 11-19.
68. Elhenawy, M. and H. Rakha, *Applying Cluster Analysis Techniques to Traffic Operations*. 2017, Virginia Department of Transportation. p. 59.

69. Demiryurek, U., et al. *Towards modeling the traffic data on road networks*. in *Proceedings of the Second International Workshop on Computational Transportation Science*. 2009. ACM.
70. Xu, D. and Y. Tian, *A Comprehensive Survey of Clustering Algorithms.* Annals of Data Science, 2015. **2**(2): p. 165-193.
71. Finley, T. and T. Joachims. *Supervised clustering with support vector machines*. in *Proceedings of the 22nd international conference on Machine learning*. 2005. ACM.
72. Basu, S., A. Banerjee, and R. Mooney. *Semi-supervised clustering by seeding*. in *In Proceedings of 19th International Conference on Machine Learning (ICML-2002*. 2002. Citeseer.
73. Basu, S., M. Bilenko, and R.J. Mooney. *Comparing and unifying search-based and similarity-based approaches to semi-supervised clustering*. in *Proceedings of the ICML-2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining*. 2003.
74. Hartigan, J.A. and M.A. Wong, *Algorithm AS 136: A k-means clustering algorithm.* Journal of the Royal Statistical Society. Series C (Applied Statistics), 1979. **28**(1): p. 100-108.
75. Johnson, S.C., *Hierarchical clustering schemes*. Psychometrika, 1967. **32**(3): p. 241-254.
76. Schölkopf, B., A. Smola, and K.-R. Müller, *Nonlinear component analysis as a kernel eigenvalue problem.* Neural computation, 1998. **10**(5): p. 1299-1319.
77. MacDonald, D. and C. Fyfe. *The kernel self-organising map*. in *Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000. Proceedings. Fourth International Conference on*. 2000. IEEE.
78. Wu, Z.-d., W.-x. Xie, and J.-p. Yu. *Fuzzy c-means clustering algorithm based on kernel method*. in *Computational Intelligence and Multimedia Applications, 2003. ICCIMA 2003. Proceedings. Fifth International Conference on*. 2003. IEEE.
79. Awasthi, P. and R.B. Zadeh. *Supervised clustering*. in *Advances in neural information processing systems*. 2010.
80. Marcu, D., *A Bayesian model for supervised clustering with the Dirichlet process prior.* Journal of Machine Learning Research, 2005. **6**(Sep): p. 1551-1577.
81. Forestier, G., P. Gançarski, and C. Wemmert, *Collaborative clustering with background knowledge.* Data & Knowledge Engineering, 2010. **69**(2): p. 211-228.
82. Chen, N., et al., *An evolutionary algorithm with double-level archives for multiobjective optimization.* IEEE transactions on cybernetics, 2015. **45**(9): p. 1851-1863.
83. Law, M.H., A.P. Topchy, and A.K. Jain. *Multiobjective data clustering*. in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. 2004. IEEE.
84. Handl, J. and J. Knowles, *An evolutionary approach to multiobjective clustering.* IEEE transactions on Evolutionary Computation, 2007. **11**(1): p. 56-76.
85. Monti, S., et al., *Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data.* Machine learning, 2003. **52**(1-2): p. 91-118.
86. Şenbabaoğlu, Y., G. Michailidis, and J.Z. Li, *Critical limitations of consensus clustering in class discovery.* Scientific reports, 2014. **4**: p. 6207.
87. Lu, C.-C., *Robust multi-period fleet allocation models for bike-sharing systems.* Networks and Spatial Economics, 2016. **16**(1): p. 61-82.
88. Bay Area Bikeshare. *Introducing Bay Area Bike Share, your new regional transit system*. 2016; Available from: http://www.bayareabikeshare.com/faq#BikeShare101.
89. Almannaa, M., M. Elhenawy, and H. Rakha, *A Novel Supervised Clustering Algorithm for Transportation System Applications.* IEEE Transactions on Intelligent Transportation Systems, 2019.
90. Ashqar, H.I., et al., *Quantifying the Effect of Various Features on the Modeling of Bike Counts in a Bike-Sharing System*. 2018.
91. Wang, B. and I. Kim, *Short-term prediction for bike-sharing service using machine learning.* Transportation research procedia, 2018. **34**: p. 171-178.

92. Xu, C., et al., *Forecasting the Travel Demand of the Station-Free Sharing Bike Using a Deep Learning Approach*. 2018.
93. Harrison, J. and M. West, *Bayesian forecasting & dynamic models*. Vol. 1030. 1999: Springer New York City.
94. Feng, C., J. Hillston, and D. Reijsbergen, *Moment-based availability prediction for bike-sharing systems.* Performance Evaluation, 2017. **117**: p. 58-74.
95. Gast, N., et al. *Probabilistic forecasts of bike-sharing systems for journey planning*. in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 2015. ACM.
96. Petris, G., S. Petrone, and P. Campagnoli, *Dynamic Linear models with R*. 2009, University of Arkansas, Fayetteville AR: Springer.
97. bayareabikeshare. *Introducing bay area bike share, your new regional transit system*. 2016; Available from: http://www.bayareabikeshare.com/faq#BikeShare101.
98. Rudloff, C. and B. Lackner. *Modeling demand for bicycle sharing systems–neighboring stations as a source for demand and a reason for structural breaks*. 2013.
99. *Introducing Bay Area Bike Share, your new regional transit system*. 2016; Available from: http://www.bayareabikeshare.com/faq#BikeShare101.
100. Dressel, J. and H. Farid, *The accuracy, fairness, and limits of predicting recidivism.* Science Advances, 2018. **4**(1).
101. West, M., P.J. Harrison, and H.S. Migon, *Dynamic generalized linear models and Bayesian forecasting.* Journal of the American Statistical Association, 1985. **80**(389): p. 73-83.
102. Almannaa, M.H., M. Elhenawy, and H.A. Rakha, *Predicting Bike Availability in Bikesharing Systems Using Dynamic Linear Models*. 2018.
103. Walteros, J. and R. Swamy, *Locating Portable Stations to Support the Operation of Bike Sharing Systems.* 2017.
104. Midgley, P., *The role of smart bike-sharing systems in urban mobility.* Journeys, 2009. **2**(1): p. 23-31.
105. Firestine, T., *Bike-Share Stations in the U.S.* 2016, Departement of Transportation Bureau of Transportation Statistics: Washington, DC, .