Final Report

Traffic State Prediction: A Traveler Equity and Multi-modal Perspective

Hesham A. Rakha
Virginia Tech Transportation Institute
3500 Transportation Research Plaza
Blacksburg, VA 24061
Tel: 540-231-1505; Fax: 540-231-1555; Email: hrakha@vt.edu

Date
May 2019

Prepared for the Urban Mobility & Equity Center, Morgan State University, CBEIS 327, 1700 E. Coldspring Lane, Baltimore, MD 21251

ACKNOWLEDGMENT

This effort was partially funded by the University Mobility and Equity Center (UMEC).

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation's University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

1. Report No. UMEC-005

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
Traffic State Prediction: A Traveler Equity and Multi-modal Perspective

5. Report Date
May 2019

6. Performing Organization Code

7. Author(s) *Include ORCID #*
Hesham A. Rakha (https://orcid.org/0000-0002-5845-2929), Mohammed Almannaa, Huthaifa Ashqar, Mohammed Elhenawy, and Ahmed Ghanem

8. Performing Organization Report No.

9. Performing Organization Name and Address
Virginia Tech Transportation Institute
3500 Transportation Research Plaza
Blacksburg, VA 24061

10. Work Unit No.

11. Contract or Grant No.
69A43551747123

12. Sponsoring Agency Name and Address
US Department of Transportation
Office of the Secretary-Research
UTC Program, RDT-30
1200 New Jersey Ave., SE
Washington, DC 20590

13. Type of Report and Period Covered
Final

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Traffic congestion has become a major problems in many urban areas, and has related environmental, economic, and equity impacts. One potential method of reducing urban traffic congestion is developing tools that plan multi-modal trips to encourage more people to ride public transportation and to provide better driving alternatives for less affluent citizens. Traffic state prediction is the key component to planning multi-modal trips in a complex transportation network. This research attempts to address transportation system state prediction problems considering private vehicles, public transit, and bike share services within the context of a multimodal transportation system. For public transit service, the proposed effort focuses on developing real-time passenger demand prediction models using multiple data sources to enhance prediction accuracy. For bike share services, the proposed effort focuses on developing prediction models for the number and travel times of bikes. Finally, for private vehicles, this research develops a comprehensive traffic prediction tool by including different categories of prediction models. The proposed prediction algorithms and tools are evaluated by comparing their performance using the field data collected in multimodal transportation system to the performance of existing prediction methods using the same data.

17. Key Words: Multimodal Transportation System, Transit Passenger Demand Prediction, Bike Share System, Travel Time Prediction.

18. Distribution Statement

19. Security Classif. (of this report) :
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
141

22. Price

Table of Contents

List of Figures

List of Tables

Introduction

Traffic congestion has become a major problems in many urban areas, and has related environmental, economic, and equity impacts. When traffic is congested, various transportation modes cannot run efficiently, causing an increase in air pollution, carbon dioxide (CO_2) emissions, and fuel use. In 2007, Americans lost $87.2 billion in wasted fuel and lost productivity. This waste reached $115 billion in 2009. Congestion also increases travel time. For example, in 1993, driving in congested conditions caused a delay of about 1.2 minutes per kilometer of travel along arterials. The congestion problem has worsened, as reported by the Texas Transportation Institute, which found that Americans' wasted hours in traffic congestion increased fivefold between 1982 and 2005.

One potential method of reducing urban traffic congestion is developing tools that plan multi-modal trips to encourage more people to ride public transportation and to provide better driving alternatives for less affluent citizens. The ability to accurately predict passenger demand can help public transit agencies minimize operational costs and improve bus service quality by properly allocating limited resources. The planning of trips also requires accurate prediction of travel times on roads in order to provide travelers with alternative modes of travel.

Smart cities have many components, including smart transportation. Smart transportation integrates different transportation networks and allows them to work together so travelers, and commuters in particular, can enjoy seamless multi-modal trips based on their preferences. This will encourage more commuters to use public transportation systems and many traffic-related problems, such as congestion, will be relaxed. In designing smart transportation systems, it is important to consider the last mile problem, which must be solved in order for different transportation networks to work together efficiently. This problem is defined as "the short distance between home and public transit or transit stations and the workplace, which may be too far to walk."

One solution to this problem is a bike sharing system (BSS), which takes advantage of the BSS's operating data to efficiently operate the network. Smart bike sharing systems use recent technologies to monitor the status of each station in the network, collect bike usage data and other relevant data, and use state-of-the-art algorithms to build predictive models, predict future bike availability, and find good solutions for the issue of imbalance in the distribution of bikes in order to guarantee users' satisfaction and meet their demand. A well-operated BSS can help solve the last mile problem, thereby encouraging more people to use public transportation and relieving traffic congestion.

Due to relatively low capital and operational costs, as well as ease of installation, many U.S. cities are investing in BSSs. A technical report distributed by the Bureau of Transportation in April 2016 indicated that there are 2,655 BSS stations in 65 U.S. cities, and that 86.3% of these stations are connected to another means of scheduled public transportation (Contardo, Morency, & Rousseau, 2012). These numbers show that the physical infrastructure for BSSs already exists and that they are good candidates for connecting different transportation networks. In 2013, San Francisco launched the Bay Area Bike Share System (now Ford GoBike), a membership-based system providing 24-hours-per-day, 7-days-per-week self-service access to short-term rental bicycles. Members can check out a bicycle from a network of automated stations, ride to the station nearest their destination, and leave the bicycle safely locked for someone else to use (Bay Area Bike Share). The Bay Area Bike Share is designed for short, quick trips, and as a result, additional fees apply for trips longer than 30 minutes. In this system, 70 bike stations connect users to public transit, businesses, and other destinations in four areas: downtown San Francisco, Palo Alto, Mountain View, and downtown San Jose (Bay Area Bike Share). Bay Area Bike Share is available to everyone 18 years and older with a credit or debit card. The system is designed to be used by commuters and tourists alike, whether they are trying to get across town at rush hour, traveling to and from Bay Area Rapid Transit and Caltrain stations, or pursuing daily activities (Bay Area Bike Share).

However, BSSs suffer from a central recurring problem: rebalancing. Rebalancing is a daily problem for BSS operators, who have to find an efficient way to redistribute (i.e., rebalance) bikes from full stations to empty stations to meet expected demand patterns. This redistribution problem is a generalization of the well-known traveling salesman problem, which involves finding the shortest route passing through each of a collection of locations and then returning to a starting point. This problem was first proposed in J. Schuijbroek, R. C. Hampshire, & W.-J. Van Hoeve (2017) as a one-commodity pick-up and delivery traveling salesman problem. The problem is NP-hard, so heuristic optimization techniques are applied to determine a near optimum tour (i.e., route). Rebalancing can be classified as either static, dynamic, or incentivized. In both static and dynamic rebalancing, BSS operators usually use a fleet of trucks to perform the task. Static rebalancing is generally referred to in the literature as the static bicycle repositioning problem (SBRP). The common assumption of SBRP algorithms is that the number of bikes at each station either remains the same or changes slightly, and does not affect the rebalancing outcome. Thus, demand prediction is needed to check the validity of this assumption. The dynamic bicycle repositioning problem (DBRP) assumes that moving bikes will have a significant impact on BSS user demand, which will affect the rebalancing outcome. As such, demand predictions have to be input into the algorithm for solving the DBRP so that they are incorporated into the solution. Incentivized rebalancing is based on providing BSS users with incentives to contribute to the system rebalancing. The BSS sends control signals to users suggesting slight changes to their planned journeys, providing them with alternate routes, or offering the option to return bikes for system credit. These suggestions will depend on the demand prediction at stations near the destination station of the planned trip. Consequently, real-time system state prediction and bike travel time prediction are important in BSS design and management.

This research adopts the state of art statistical learning, machine learning, to predict the available bikes at each station in the network. The machine learning models employed will consider several factors, including weather, season, and stations' spatial locations. These models are essential in multi-modal trip planning, where it is not acceptable to guide the user to pick up a bike from an empty station or to return a bike to a full station at the end of a trip. Moreover, the proposed research adopts machine learning to model bikes' travel time. Predicting travel time for bikes is important in examining scheduled transportation mode and for predicting the state of the bike drop-off station.

In this report, we addressed transportation system state prediction problems considering private vehicles, transit, and BSSs within the context of a multimodal transportation system. The proposed effort focused on developing prediction models for the number of bikes and bike travel times. In addition, we developed a comprehensive traffic prediction tool by including different categories of prediction models.

This work yields eight contributions to the literature. First, we proposed a new hierarchical classifier that increases the accuracy of traditional transportation mode classification algorithms. We also investigated the possibility of improving classification accuracy by extracting new frequency domain features. The proposed framework has two layers. The first layer contains a multiclass classifier that discriminates between five transportation modes and identifies the two most probable modes. The second layer consists of binary classifiers that differentiate between the two chosen modes that were identified in the first layer. In addition, the proposed framework combines the new extracted features with traditionally used time domain features to create a pool of features.

Second, we proposed a new supervised clustering algorithm to provide a global view of network-wide bike availability across stations. To do so, we developed a novel supervised clustering algorithm built using two well-known algorithms: the Gale-Shapley student optimal college admission (CA) algorithm (Gale & Shapley, 1962) and the K-median algorithm.

Third, we introduced an effective approach to quantifying the effect of various features on the prediction of bike counts at each station in the San Francisco Bay Area Bike Share. The Random Forest (RF) technique was used to rank the predictors, then guided forward step-wise regression and Bayesian information criterion (BIC) were used to develop and compare BSS prediction models, respectively.

Fourth, we adopted state-of-the-art machine learning and statistical techniques to build predictive models of bike availability at each station in the BSS. The built models were compared in terms of mean absolute error (MAE), prediction accuracy, computational time and we identified which algorithm was suitable for which condition.

Fifth, we investigated the traditionally-known vs. a novel quality-of-service (QoS) measurement, and found that neither exposed the spatial dependencies between stations nor did either discriminate between stations in a BSS. Therefore, we proposed a novel QoS measurement, Optimal Occupancy, that captures the impact of a BSS' heterogeneity and reflects the spatial dependencies between stations.

Sixth, we built a Markov chain model for each bike station. The models were 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, hence minimizing the rebalancing cost.

Seventh, we proposed 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 BSS: dock and dock-less. The proposed portable stations can function as either individual stations (standalone) or as an extension of existing bike stations. This concept was 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 will 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.

Eighth, we developed different bike travel time prediction models using machine learning techniques. The main contribution of this work is finding the best predictors to explain bike travel time variability. The techniques used in this work do not require any assumptions about the data.

In terms of the report layout, following the introduction, we will discuss each of the eight contributions separately. Each contribution will be discussed in depth, including related work, methodology, and results. Finally, the summary findings and conclusions of the work will be presented.

Chapter 1. Smartphone Transportation Mode Recognition Using a Hierarchical Machine Learning Classifier and Pooled Features from Time and Frequency Domains

1.1 Introduction

The application of smartphones to data collection has recently attracted researchers' attention. Apps have been developed and effectively used to collect data from smartphones in many sectors. In transportation, researchers can track smartphones and obtain information such as speed, acceleration, and the rotation vector from the GPS, accelerometer, and gyroscope sensors embodied in smartphones (Susi, Renaudin, & Lachapelle, 2013). These data can then be used to recognize the user's transportation mode, which has several applications, as shown in Table 1-1.

Table 1-1. Transportation mode detection applications (Elhenawy, Jahangiri, & Rakha, 2016).

Application	Description
Transportation Planning	Instead of using traditional approaches such as questionnaires, travel diaries, and telephone interviews (Leon Stenneth, Ouri Wolfson, Philip S. Yu, & Bo Xu, 2011b; X. Yu et al., 2012), the transportation mode information can be automatically obtained through mobile phone sensors.
Safety	Knowing the transportation mode used would help in developing safety applications. For example, violation prediction models have been studied for passenger cars and bicycles (Jahangiri, Rakha, & Dingus, 2015).
Environment	Physical activities, health, and calories burned, and carbon footprint associated with each mode can be obtained when the transportation mode is known (S. Reddy et al., 2010).
Information Provision	Traveler information can be provided based on the transportation mode (Manzoni, Maniloff, Kloeckl, & Ratti, 2010; Stenneth et al., 2011b).

In this study, we investigated the possibility of improving the overall accuracy of transportation mode detection by proposing a new hierarchical framework classifier and by looking for a new features set. This chapter makes two major contributions to existing work in this realm. First, it proposes a two-layer hierarchical framework in which the first layer contains one multi-classifier using a dataset of the five transportation modes. The second layer consists of 10 binary classifiers, each of which is specialized in only one pair of modes and uses a features subset that discriminates between this pair. Second, new frequency domain features were extracted and pooled with the time domain features that have been traditionally used.

Following the introduction, this chapter is organized into six sections. First, the approaches, features, and machine learning techniques of previous studies are reviewed. Next, the dataset and the extracted features are described. Third, background is presented on the machine learning techniques applied in this study. Next, the proposed framework is presented. In the fifth section, details are provided on the data analysis used to detect different transportation modes. Finally, the chapter concludes with a summary of new insights and recommendations for future transportation mode recognition research.

1.2 Related Work

Researchers have developed several approaches to discriminate between transportation modes effectively (Kwapisz, Weiss, & Moore, 2011; Leon Stenneth, Ouri Wolfson, Philip S Yu, & Bo Xu, 2011a; Susi et al., 2013). Machine learning techniques have been used extensively to build detection models and have shown high accuracy in determining transportation modes. Supervised learning methods such as the following have been employed:

- **K-Nearest Neighbor (KNN)** (Jahangiri & Rakha, 2015)
- **Support Vector Machines (SVMs)** (Bolbol, Cheng, Tsapakis, & Haworth, 2012; Nham, Siangliulue, & Yeung, 2008; Nick, Coersmeier, Geldmacher, & Goetze, 2010; S. Reddy et al., 2010; Zhang, Qiang, & Yang, 2013; Zheng, Liu, Wang, & Xie, 2008)
- **Decision Trees** (Manzoni et al., 2010; S. Reddy et al., 2010; Stenneth et al., 2011b; Widhalm, Nitsche, & Brandie, 2012; X. Yu et al., 2012; Zheng et al., 2008)
- **RFs** (Jahangiri & Rakha, 2015)

Several factors affect the accuracy of detecting transportation modes, such as the monitoring period (positive association), number of modes (negative association), data sources, motorized classes, and sensor positioning (see more details in (Elhenawy et al., 2016; Jahangiri & Rakha, 2015)). However, one of the more important factors that affects the accuracy of mode detection is the machine learning framework classifier.

An additional important consideration is the domain of the extracted features. Extracted features from the time domain have been used widely in many studies (Biljecki, Ledoux, & Van Oosterom, 2013; Jahangiri & Rakha, 2015; Nham et al., 2008; Nick et al., 2010; Sasank Reddy et al., 2010; Stenneth et al., 2011b) and have achieved a significant, high accuracy.

The factors applied to mode detection affect the accuracy of models. Table 1-2 summarizes the obtained accuracies and factors for some of the aforementioned studies. Note that no direct comparison can be made between the studies listed in Table 1-2 because the factors considered and the datasets used varied from study to study.

Table 1-2. **Summary of some past studies (Elhenawy et al., 2016).**

Accuracy (%)	Features Domain	Machine Learning Framework	Monitoring Period	No. of Modes	Data Sources	More than One Motorized Mode?	Sensor Positioning	Dataset	Study
97.31	Time	Traditional	4 s	3	Accelerometer	Yes	No requirements	Not mentioned	(Nick et al., 2010)
93.88	Frequency	Traditional	5 s, 50% overlap	6	Accelerometer	Yes/No	Participants were asked to keep their device in the pocket of their non-dominant hip	Collected from 4 participants	(Nham et al., 2008)
93.60	Time and frequency	Traditional	1 s	5	Accelerometer GPS	No	No requirements	Collected from 16 participants	(S. Reddy et al., 2010)
93.50	Time	Traditional	30 s	6	GPS, GIS[a] maps	Yes	No requirements	Collected from 6 participants	(Stenneth et al., 2011b)
95.10	Time	Traditional	1 s	5	Accelerometer, gyroscope, rotation vector	Yes	No requirements	Collected from 10 participants	(Jahangiri & Rakha, 2015)
91.60	Time	Traditional	Entire trip	11	GPS, GIS maps	Yes	No requirements	Two different datasets, one of which included 1,000 participants	(Biljecki et al., 2013)
96.32	Time	Hierarchical	1 s	5	Accelerometer, gyroscope, rotation vector	Yes	No requirements	Collected from 10 participants	(Elhenawy et al., 2016)

[a] GIS: Geographic Information System

1.3 Dataset

1.3.1 Data Collection

The dataset used is available at the Virginia Tech Transportation Institute (VTTI) and was collected by Jahangiri and Rakha (2015) using a smartphone application (Jahangiri & Rakha, 2015). The application was provided to 10 travelers who work at VTTI to collect data for five different modes: driving a passenger car, bicycling, taking a bus, running, and walking. The data were collected from GPS, accelerometer, gyroscope, and rotation vector sensors and stored on the devices at the application's highest possible frequency. Data collection was conducted on different workdays (Monday through Friday) and during working hours (8 a.m. to 6 p.m.). Several factors were considered for collecting realistic data that reflects natural behaviors. No specific requirement was applied in terms of sensor positioning. The data were collected on different road types with different speed limits in Blacksburg, Virginia, and some epochs may reflect traffic jam conditions occurring in real-world conditions. Thirty minutes over the course of the study for each mode per person were considered sufficient for collection of enough data.

For comparison purposes with previous studies using the same types of data (Elhenawy et al., 2016; Jahangiri & Rakha, 2015), the extracted features were considered to have a meaningful relationship with different transportation modes. Furthermore, features that might be extracted from the absolute values of the rotation vector sensor were excluded. Additionally, in order to allow this framework to be implemented in cases where no GPS data were available, features that might be extracted from GPS data were also excluded.

1.3.2 Time Domain Features

From the time window t, time domain features were created by applying the measures in Table 1-3. These measures were applied twice: first, using the measurements of the data array for the i^{th} feature from window t; and second, using the measurements of the derivative of the same data array for the i^{th} feature from window t. This resulted in 165 time domain features.

Table 1-3. Measurements of time domain features.

No.	Measure	No.	Measure
1	$mean()$	6	$range()$
2	$max()$	7	$Interquartile\ range,\ iqr()$
3	$min()$	8	$signChange()$
4	$variance()$	9	$energy()$
5	$standard\ deviation()$	10	$spectralEntropy()$

1.3.3 Frequency Domain Features

Jahangiri and Rakha (2015) collected readings from the mobile sensors at a frequency of almost 25 Hz. Because the output samples of the sensors were not synchronized, they implemented a linear interpolation to build continuous signals from the discrete samples. Consequently, they sampled the constructed sensor signals at 100 Hz and divided the output of each sensor in each direction ($x, y,$ and z) into non-overlapping windows of 1-s width. Finally, the features used for mode recognition were extracted from each window. These features were mainly traditional statistics, such as mean, minimum, and maximum. The use of these features achieved a good accuracy in mode recognition.

However, some information loss can be expected due to the usage of the summary statistics. Summary statistics consists of some descriptive statistics analysis for variability, center tendency, and distribution, such as mean, range, and variance. Summary statistics occasionally fail to detect the correlations, extract optimal information, and define probabilities (Fedor-Freybergh & Mikulecký, 2005; Tan, 2006). Since each window is considered as a signal in the time domain, we can improve the mode recognition accuracy by transforming this signal into the frequency domain using the short-time Fourier transform. After transforming the time domain signal to the frequency domain and neglecting the phase information, we visually inspected the resultant spectrum and found that most of the information was provided by the first 20 components. Therefore, in this study, we used the magnitude of the first 20 components as the new frequency independent features. Transforming time domain into frequency domain not only adds new transferred features from an original space (i.e., time) to a new space (i.e., frequency), but also imposes more control on the loss of information. This process added another 180 features extracted from the frequency domain to the dataset (i.e., 345 features pooled in total).

1.4 Methods

This section describes the feature selection algorithm and the machine learning classifiers used in the proposed hierarchical framework.

1.4.1 K-Nearest Neighbor (KNN)

KNN is a common algorithm in supervised learning that classifies the data points based on the K nearest points. K is a user parameter that can be determined using different techniques. The test observation (i.e., y_j^{test}) is classified by taking the majority vote of the classes of the K nearest points (i.e., y_j^{train}), as shown in (1-1)
(Friedman, Baskett, & Shustek, 1974).

$$\boldsymbol{y_j^{test}} = \frac{\boldsymbol{1}}{\boldsymbol{K}} \sum_{\boldsymbol{X_j^{train}} \in \boldsymbol{N_K}} \boldsymbol{y_j^{train}} \tag{1-1}$$

where, y_j^{test} is the class of the testing data; y_j^{train} is the class of the training data; X_j^{train} is the testing data; and K is the number of classes.

1.4.2 Classification and Regression Tree (CART)

The CART algorithm was introduced in the early 1980s by Olshen, and Stone (Olshen & Stone, 1984). This algorithm is a type of decision tree where each branch represents a binary variable. At each split, the CART algorithm trains the tree using a greedy algorithm. Different splits are tested, and the split with the lowest cost is chosen. After many splits, each branch will end up in a single output variable that is used to make a single prediction. The CART algorithm will stop splitting when reaching a certain criterion. The two most common stopping criteria are setting a minimum count of the training instances assigned to each leaf and choosing a pruning level that produces the highest accuracy.

1.4.3 Support Vector Machines (SVMs)

The SVM algorithm is a supervised learning technique that is used to classify the data by maximizing the gap between classes. The SVM algorithm attempts to find the hyperplane (i.e., splitter) that gives the largest minimum distance to the training data as given in Equation (1-2). The SVM tries to find the weight (w) that produces the largest margin around the hyperplane (see (1-2)), while satisfying the two constraints (see (1-3) and (1-4); (Hsu & Lin, 2002).

$$\min_{\boldsymbol{w},\boldsymbol{b},\boldsymbol{\xi}} \left(\frac{\mathbf{1}}{\mathbf{2}} \boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{w} + \boldsymbol{C} \sum_{\boldsymbol{n}=\mathbf{1}}^{\boldsymbol{N}} \boldsymbol{\xi}_{\boldsymbol{n}}\right) \tag{1-2}$$

subject to:

$$\boldsymbol{y}_{\boldsymbol{n}}\left(\boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{\phi}(\boldsymbol{x}_{\boldsymbol{n}}) + \boldsymbol{b}\right) \geq \mathbf{1} - \boldsymbol{\xi}_{\boldsymbol{n}}, \boldsymbol{n} = \mathbf{1}, \ldots, \boldsymbol{N} \tag{1-3}$$

$$\boldsymbol{\xi}_{\boldsymbol{n}} \geq \mathbf{0}, \boldsymbol{n} = \mathbf{1}, \ldots, \boldsymbol{N} \tag{1-4}$$

where,

w	Parameters to define the decision boundary between classes
C	Penalty parameter
ξ_n	Error parameter to denote margin violation
b	Intercept associated with the hyperplanes
$\phi(x_n)$	Function to transform data from X space into some Z space
y_n	Target value for n^{th} observation

1.4.4 Random Forest (RF)

Breiman proposed RF as a new classification and regression technique in supervised learning (Breiman, 2001). The RF method randomly constructs a collection of decision trees in which each tree chooses a subset of features to grow, and the results are then obtained based on the majority votes from all trees. The number of decision trees and the selected features for each tree are user-defined parameters. The reason for choosing only a subset of features for each tree is to prevent the trees from being correlated. RF was applied in this study to select the best subset of features to be used in classification since the RF technique offers several advantages. For example, it runs efficiently on large datasets and many input features without the need to create extra dummy variables, and it ranks each feature's individual contribution in the model (Breiman, 2001; Loh, 2011).

1.5 Proposed Framework

As many features could be used to discriminate between modes, we applied feature selection to choose the subset of features with the highest importance in discriminating between modes. The subset of selected features, which was used in the classifiers, depends on the classified modes. This means that the subset of features selected to discriminate between all modes will be different from the subset of features selected to discriminate between only two modes. In this study, RF was used to select the best 100-feature subset for each classifying step. Selected features were scaled so that the feature values were normalized to be within the range of $[-1, 1]$.

Figure 1-1 shows the importance of features in different ranks for all the modes combined and for different pairs of modes. The least important feature is ranked 0, and the highest is ranked 2. In Figure 1-1, it can be seen that the important feature of one pair may be different from other pairs and that its rank within pairs may also vary.

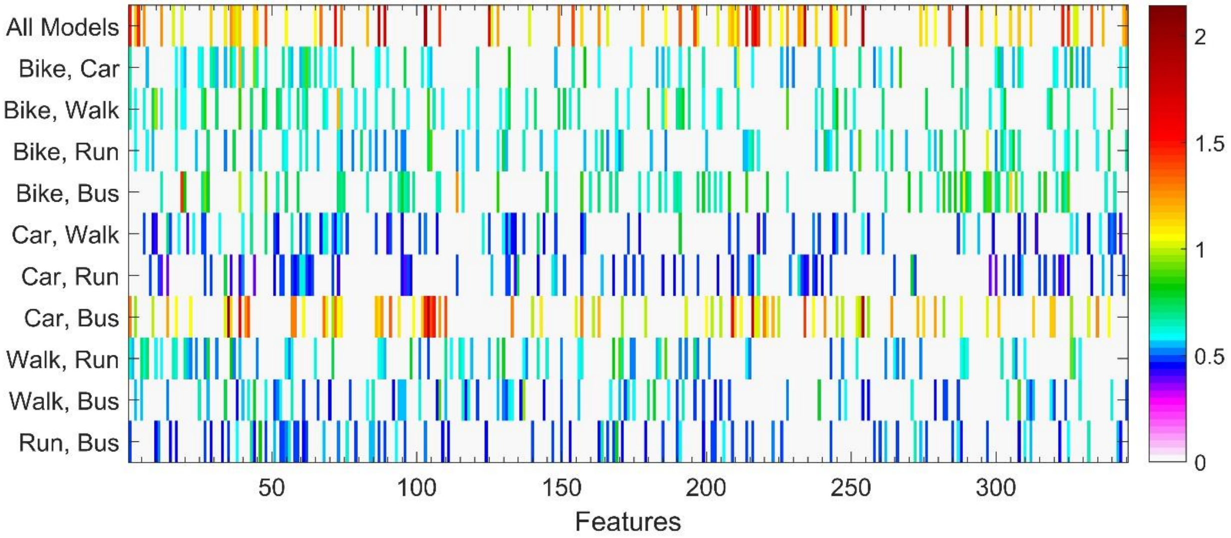


Figure 1-1. Importance of features for different pairs of modes.

This study proposes a new approach to detect transportation mode. Two layers are applied as a hierarchical framework. The first layer consists of only one multiclass classifier to discriminate between the five modes, and the second layer consists of a pool of 10 binary classifiers, which are used to classify only two modes. The first layer is trained using the 100 features to return the two most likely modes. The second layer is trained using a different 100 features, specialized to differentiate between only two modes, to return one mode out of the most two likely modes resulting from the first layer. Bayes' rule (i.e., $posterior\, probability \propto likelihood \times prior\, probability$) is used in this framework to combine the output of the two layers. The mode that has the largest posterior probability is chosen, given that the first layer probability is the prior probability and the second layer probability is the likelihood.

1.6 Data Analysis and Results

This section discusses the results of the machine learning techniques used in this study that were developed in MATLAB.

1.6.1 K-Nearest Neighbors Algorithm (KNN)

In this study, KNN was used to identify the mode from the five transportation modes in the first layer and the two modes in the second layer. The optimal *K* was chosen after testing different numbers of *K* versus the overall classification accuracy. To select the best model at each value of *K*, a 10-fold cross-validation was performed, and the average highest accuracy among the 10 folds was chosen. As shown in Figure 1-2, using the pooled features in the hierarchical framework achieved a higher classification accuracy than only using the time domain features in the same framework. However, using the time domain features in the proposed hierarchical framework outperformed traditional KNN classification for pooled features. The optimal K was found to be 7, with the highest accuracy of 95.49%.

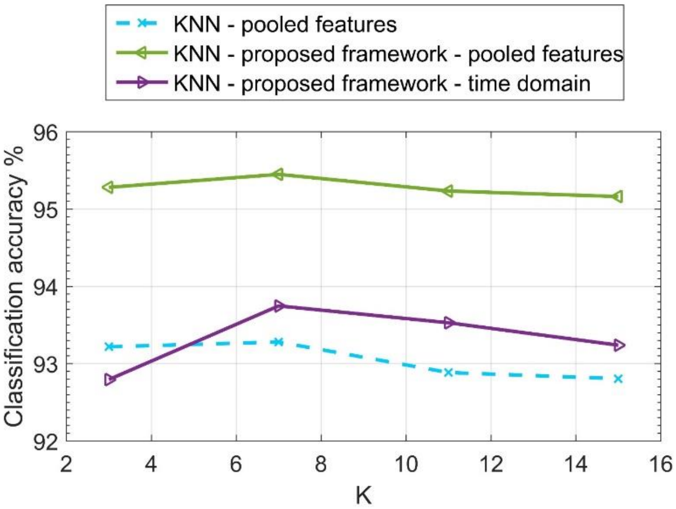


Figure 1-2. Classification accuracy for KNN in different cases at different neighbors.

1.6.2 Classification and Regression Tree (CART)

Ten folds for the cross-validation process were applied for each pruning level, ranging from 2 to 20, and the average was taken as a comparison value with other pruning levels. Figure 1-3 provides a comparison between time domain features, frequency domain features, and pooled features under different pruning levels. The figure shows that using pooled features (compared to the same applied approach) produces the highest accuracy of 93.52% at six pruning levels when applying the proposed framework among all other cases. Figure 1-3 also shows that the classification accuracy of using only frequency domain features (compared to the same applied approach) is lower than using time domain features.

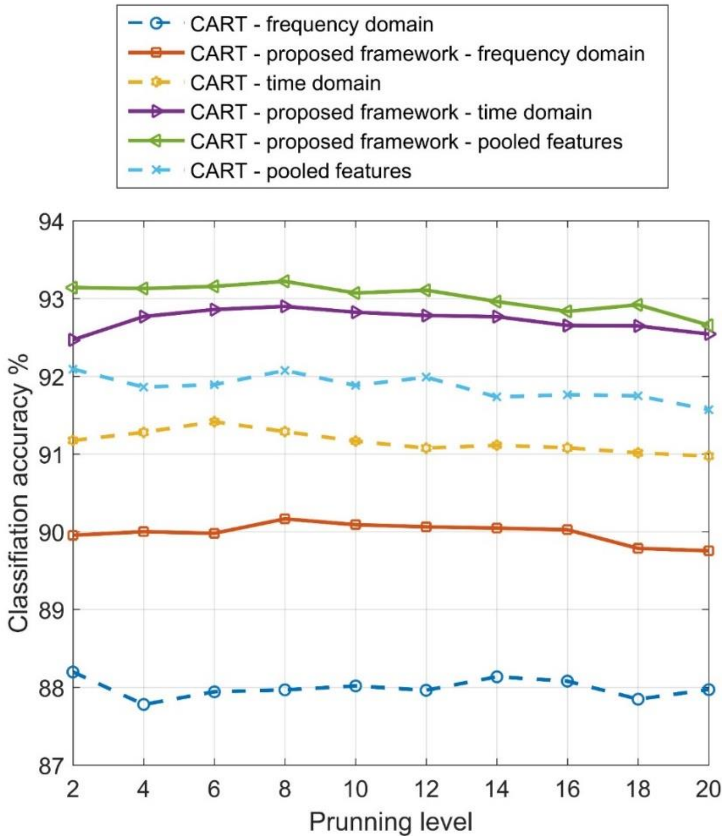


Figure 1-3. Classification accuracy for CART in different cases at different pruning levels.

1.6.3 Support Vector Machine (SVM)

SVM was applied in the proposed framework using time domain, frequency domain, and pooled features. A 10-fold cross-validation was applied to develop a single model. The results show that using pooled features improved the average overall classification accuracy from 96.10% to 97.00%. The overall accuracy for using only the frequency domain features was the lowest at 93.92%. Table 1-4 presents the overall classification accuracy for the 10-fold testing applying the proposed SVM framework.

Table 1-4. Overall classification accuracy for the SVM using time domain, frequency domain, and pooled features.

Fold	Time domain features (%)	Frequency domain features (%)	Pooled features (%)
1	96.04	93.78	97.12
2	96.32	93.65	97.31
3	95.88	92.90	96.88
4	96.10	93.04	97.32
5	95.98	94.01	96.76
6	96.02	94.79	96.81
7	96.38	93.62	96.98
8	95.71	94.57	96.93
9	96.32	94.52	97.01

Fold	Time domain features (%)	Frequency domain features (%)	Pooled features (%)
10	96.25	94.28	96.91
Average	96.10	93.92	97.00

The confusion matrix applying SVM in the proposed framework using pooled features is given in Table 1-5. The precision for run mode was the highest. The precision for bus mode was the lowest. However, the recall was the lowest for run mode and highest for bike mode.

Table 1-5. Confusion matrix for SVM using pooled features.

		Actual					
		Bike	**Car**	**Walk**	**Run**	**Bus**	**Precision**
Predicted	**Bike**	97.13	0.52	1.17	0.40	0.58	97.33
	Car	0.66	93.57	0.16	0.13	3.06	95.88
	Walk	0.92	0.08	93.59	0.92	0.29	97.68
	Run	0.37	0.05	0.93	92.82	0.20	98.36
	Bus	0.92	2.42	0.40	0.32	93.11	95.81
	Recall	97.13	93.57	93.59	92.82	93.11	

1.6.4 Random Forest (RF)

The RF was run with different numbers of trees to investigate the impact of the number of trees on the classification accuracy. A number of trees ranging from 200 to 400 was chosen, as the highest benefit was expected to be gained in this range according to previous studies (see more details in (Elhenawy et al., 2016; Jahangiri & Rakha, 2015). Applying RF in the proposed framework using pooled features resulted in the highest classification accuracy of 96.24% at 200 trees, as illustrated in Figure 1-4. Figure 1-4 also illustrates that applying RF using a traditional approach to classify the modes but also using pooled features produced higher accuracy than the RF in the proposed framework using only the time domain features in classification.

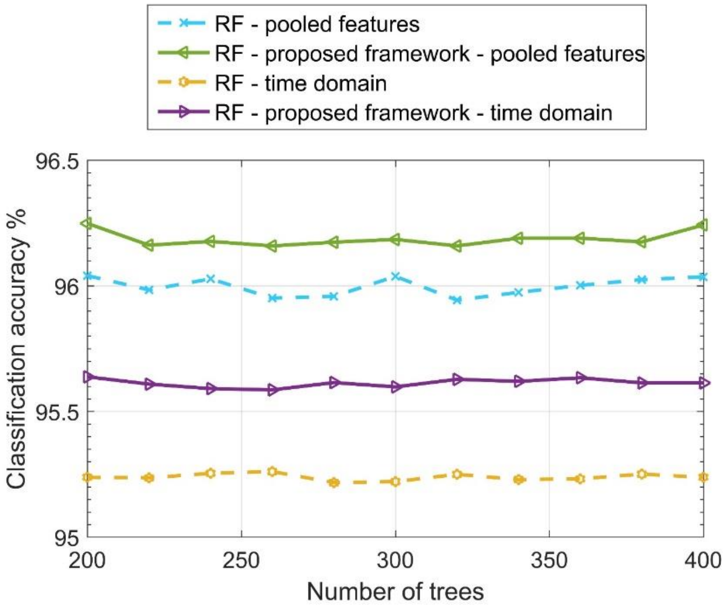


Figure 1-4. Classification accuracy for RF in different cases at different number of trees.

A comparison between time domain, frequency domain, and pooled features was carried out using the RF method in the proposed framework, as shown in Table 1-6. The results demonstrate that using the pooled features improved the overall classification accuracy from 95.61% to 96.24%.

Table 1-6. Overall classification accuracy for RF using time domain, frequency domain, and pooled features.

Fold	Time domain features (%)	Frequency domain features (%)	Pooled features (%)
1	95.95	94.15	96.35
2	95.73	94.34	96.59
3	95.61	93.91	96.07
4	95.37	94.02	96.22
5	95.51	93.85	96.24
6	95.56	94.09	96.30
7	95.67	93.82	96.23
8	95.78	93.64	96.13
9	95.49	94.18	96.39
10	95.47	93.78	95.88
Average	95.61	93.98	96.24

Table 1-7 shows the confusion matrix for the RF proposed framework using pooled features. The run mode had the highest precision and the bus mode had the lowest precision.

Table 1-7. Confusion matrix for RF using pooled features.

		Actual					
		Bike	**Car**	**Walk**	**Run**	**Bus**	**Precision**
Predicted	**Bike**	94.63	0.40	2.59	0.05	0.94	95.96
	Car	0.97	92.54	0.13	0.00	2.78	95.96
	Walk	1.87	0.10	91.74	0.25	0.70	96.92
	Run	0.75	0.05	1.47	90.39	0.57	96.96
	Bus	1.78	2.43	0.13	0.00	91.67	95.48
	Recall	94.63	92.54	91.74	90.39	91.67	

1.6.5 Heterogeneous Framework RF-SVM

A heterogeneous framework was performed in which the RF classifier was used in the first layer to classify all modes and a binary SVM classifier was applied in the second layer. The overall classification accuracy was improved from 96.32% to 97.02% by using pooled features compared to only using time domain features, as presented in Table 1-8.

Table 1-8. Overall classification accuracy for RF-SVM using time domain, frequency domain, and pooled features.

Fold	**Time domain features (%)**	**Frequency domain features (%)**	**Pooled features (%)**
1	96.51	94.26	96.96
2	96.38	94.74	96.91
3	96.52	94.78	96.86
4	96.26	94.83	96.83
5	96.44	93.71	96.97
6	96.10	95.17	96.66
7	96.12	95.30	97.36
8	96.16	94.86	97.11
9	96.33	94.49	97.39
10	96.36	94.86	97.16
Average	96.32	94.70	97.02

Table 1-9 and Table 1-10 provide the confusion matrix for applying RF-SVM in the proposed framework using time domain features and the pooled features, respectively.

Table 1-9. Confusion matrix for RF-SVM using time domain features.

		Actual					
		Bike	**Car**	**Walk**	**Run**	**Bus**	**Precision**
Predicted	**Bike**	97.83	0.75	1.32	0.72	2.02	95.39
	Car	0.44	94.74	0.15	0.05	3.84	95.51
	Walk	1.03	0.10	97.61	0.98	0.15	97.80
	Run	0.00	0.00	0.20	97.63	0.05	99.74
	Bus	0.69	4.41	0.73	0.62	93.93	93.50
	Recall	97.83	94.74	97.61	97.63	93.93	

Table 1-10. Confusion matrix for RF-SVM using pooled features.

		Actual					
		Bike	**Car**	**Walk**	**Run**	**Bus**	**Precision**
Predicted	**Bike**	96.12	0.34	1.17	0.06	0.71	97.79
	Car	0.69	96.81	0.16	0.01	2.85	96.27
	Walk	1.22	0.10	97.27	0.36	0.44	97.82
	Run	0.65	0.05	1.18	99.54	0.48	97.55
	Bus	1.32	2.70	0.22	0.04	95.52	95.67
	Recall	96.12	96.81	97.27	99.54	95.52	

1.7 Conclusions and Recommendations for Future Work

This study proposes a two-layer hierarchical framework classifier to distinguish between five transportation modes using new extracted frequency domain features pooled with traditionally used time domain features. We investigated the possibility of improving the classification accuracy using pooled features in the proposed framework by applying several techniques: KNN, CART, SVM, RF, and RF-SVM. The results show that using pooled features in the proposed framework increased the classification accuracy for all the applied classifiers. For the same data, the highest reported accuracy was 95.10% using the traditional approach for detection, whereas the proposed approach in this study achieved an accuracy of 97.02%. This implies that (a) pooling new features to be selected as classifying features increases the classification accuracy regardless of the applied approach and algorithm, and (b) applying the proposed hierarchal framework further increases the classification accuracy. The proposed hierarchical framework outperformed the traditional approach of applying only a single layer of classifiers.

Although using pooled features increased accuracy, using the new extracted features alone (i.e., frequency domain) resulted in a lower accuracy than only using time domain features. Transferring time domain into a new space (i.e., frequency domain) and using the magnitude of the first 20 components enhanced the control on the information loss. This means that combining different features together in a big pool and then choosing the best subset returns better results than using one domain of features alone. Finally, the heterogeneous classifier, using RF in the first layer and SVM in the second layer, was found to produce the best overall performance.

As a future recommendation, it is important to use a further deep analysis, such as Canonical Correlation Analysis, to correlate between the features in order to obtain better coordinated results. Furthermore, future work should investigate the sensitivity of the results to the monitoring period and the potential use of GPS data.

Chapter 2. Novel Supervised Clustering Algorithm for Transportation System Applications

2.1 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 transportation network’s efficiency 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 assist operating agencies in answering 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 (Arbelaez, Maire, Fowlkes, & Malik, 2011), marketing to find similar customer behaviors (Roberts, 1995), the insurance industry to identify fraud (Ngai, Hu, Wong, Chen, & Sun, 2011; Thiprungsri & Vasarhelyi, 2011), and in transportation to identify similar patterns in various modes of transport (Calafate, Soler, Cano, & Manzoni, 2015; Elhenawy, Chen, & Rakha, 2014; Weijermars & van Berkum, 2005). 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 BSS data (Froehlich, Neumann, & Oliver, 2009b; Vogel, Greiser, & Mattfeld, 2011a). 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 (Côme Etienne & Oukhellou Latifa, 2014). 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 perform 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 (Bar-Hillel, Hertz, Shental, & Weinshall, 2003; Eick, Zeidat, & Zhao, 2004; Sinkkonen, Kaski, & Nikkilä, 2002). 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 (Eick et al., 2004; Spinelli, 2017).

In this chapter, we propose a new supervised clustering algorithm based on the CA game theory algorithm (Gale & Shapley, 1962) to simultaneously maximize the reciprocal of the within-cluster sum of distances (similarity) and the cluster purity. 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, it was tested using the aforementioned the San Francisco Bay Area BSS dataset, 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.

2.2 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 (Calafate et al., 2015; Elhenawy et al., 2014; Elhenawy & Rakha, 2015; Weijermars & van Berkum, 2005). Various classical approaches, such as k-means, Ward's hierarchical clustering algorithms, and density-based clustering, have been adopted to accomplish this. 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 (e.g., 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 (Demiryurek, Pan, Banaei-Kashani, & Shahabi, 2009; Elhenawy & Rakha, 2017a).

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 for similar time events. The proposed algorithm is scalable (polynomial order), fast, and ready for practitioners' use. It makes no assumptions about the dataset and requires only one parameter—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.

2.3 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 (D. Xu & Tian, 2015). 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 (Eick et al., 2004). Another algorithm uses the labels to learn the best similarity measure that produces the desirable clustering solution (Finley & Joachims, 2005). 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 (Basu, Banerjee, & Mooney, 2002; Basu, Bilenko, & Mooney, 2003) (this is not covered in this chapter, as it is very different from the work conducted here).

Two examples of unsupervised clustering algorithms are the well-known k-means and hierarchical clustering algorithms (Hartigan & Wong, 1979; Johnson, 1967). The k-means simply partitions the data points into clusters, minimizing the distortion of each cluster (Hartigan & Wong, 1979). 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 (Johnson, 1967). At every level of the tree-based structure, similar clusters are merged into one cluster. The key to this clustering algorithm are 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 (Schölkopf, Smola, & Müller, 1998), kernel self-organizing maps (MacDonald & Fyfe, 2000), and kernel fuzzy c-means (Z.-d. Wu, Xie, & Yu, 2003). These algorithms attempt to cluster the data points by transforming them into a higher dimensional feature space and then carrying 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 (Eick et al., 2004; Spinelli, 2017). 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 (Eick et al., 2004). 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 (Spinelli, 2017). 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 (Awasthi & Zadeh, 2010). Yet this approach assumes that the teacher knows the ground truth of the data, which is not the case in many datasets (i.e., assumes two datasets: training and test).

Recently, supervised clustering algorithms have been enhanced greatly by using a multi-objective approach (Chen et al., 2015; Forestier, Gançarski, & Wemmert, 2010; Handl & Knowles, 2007; Law, Topchy, & Jain, 2004; Marcu, 2005). 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 these algorithms 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 (Law et al., 2004). In the first step, the authors 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, not in the objective function. Handi and Knowles proposed a multi-objective evolutionary algorithm, maximizing the compactness and connectivity of the clusters simultaneously (Handl & Knowles, 2007). 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-dominated solutions that were created by the proposed algorithm.

None of the previous approaches used both purity and 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, and these 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 (Marcu, 2005). 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 scalability is an issue. Forestier et al. proposed a collaborative clustering algorithm that incorporates three components: cluster quality, class label, and link-based constraints (Forestier et al., 2010). This approach randomly selects a subset of the dataset as background knowledge, 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 simultaneously increase both cluster purity and member similarity. 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.

2.4 The College Admission (CA) 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 (Gale & Shapley, 1962). The stable marriage problem involves one-to-one matching. The 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.

2.5 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 based on the CA algorithm (Gale & Shapley, 1962). 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 (2-1). The similarity measure is computed using (2-2). The algorithm terminates when the stopping criteria of (2-3) are met.

$$purity(c_i)^t = \max_m \left(\frac{n_i^m}{n_i}\right) \tag{2-1}$$

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

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

$$(1-\alpha) \left|\frac{\sum_{i=1}^{K} similarity(c_i)^t - \sum_{i=1}^{K} similarity(c_i)^{t-1}}{\sum_{i=1}^{K} similarity(c_i)^{t-1}}\right| < \varepsilon \tag{2-3}$$

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).

One advantage of the proposed algorithm is that it is not necessary 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. $P = \begin{bmatrix} \frac{n_1^1}{n_1} & \cdots & \frac{n_1^M}{n_1} \\ \vdots & \ddots & \vdots \\ \frac{n_K^1}{n_K} & \cdots & \frac{n_K^M}{n_K} \end{bmatrix}$
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 2-1. 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.

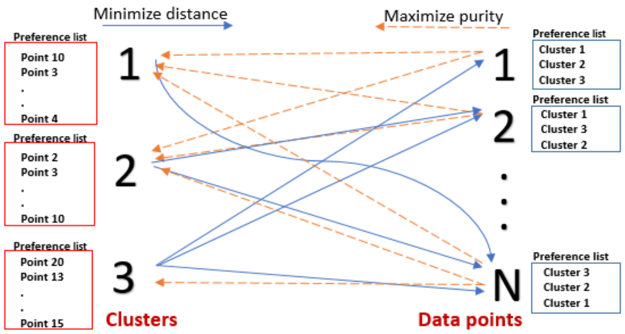


Figure 2-1. CA based clustering.

In the first step, the proposed algorithm first randomly chooses three points as centroids for the three clusters. Then, it partitions 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 2-1.

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 is recalculated. The algorithm repeats the entire process of building new preference lists, matching, and calculating new centroids and the P matrix. The algorithm stops when there is no significant improvement in the purity and similarity.

2.6 Datasets

We used docking station data collected from August 2013 to August 2015 in the San Francisco Bay area. The docking station data included 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).

2.7 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.

2.7.1 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 the aforementioned model order selection technique, CC, to determine the number of clusters (Monti, Tamayo, Mesirov, & Golub, 2003). 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 CC algorithm is run using different initial centroids (i.e., the centroids the algorithm begins with) (Șenbabaoğlu, Michailidis, & Li, 2014). 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 $c\_index \in [0,1]$ (2-5). The CDF for a particular CM is defined over the range [0, 1] as follows:

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

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 with 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 2-2 shows 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 $K = 2$, with results presented in the following section. Similarly, the optimal number of clusters with regard to the day-of-week label is $K = 3$.

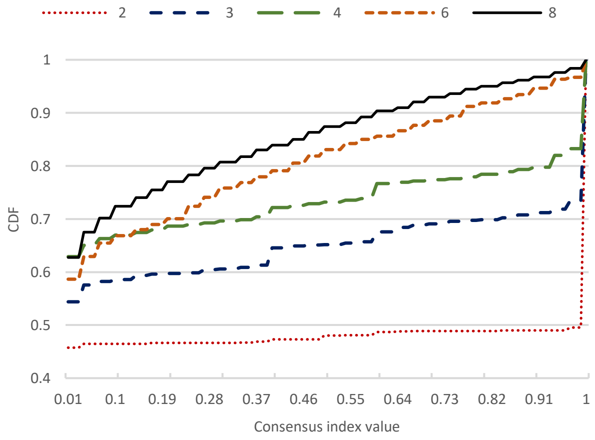


Figure 2-2. CDF against consensus index value for each cluster – time of day using BSS station status data.

2.8 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 2-3, 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

(Kaltenbrunner, Meza, Grivolla, Codina, & Banchs, 2010) 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.

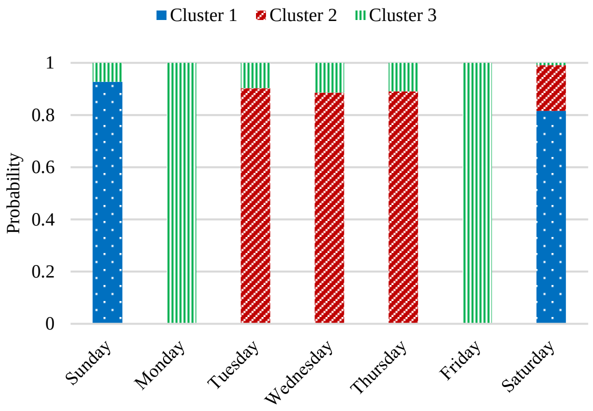


Figure 2-3. The probability of the day of week being in one of the three clusters (*K* = 3).

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 2-4.

Three observations can be made from Figure 2-4. 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.

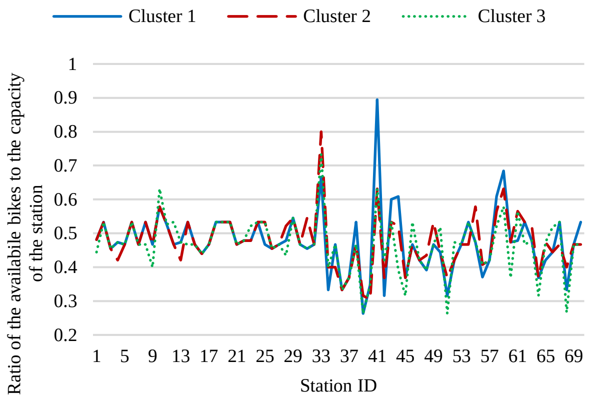


Figure 2-4. The ratio of the available bikes to station capacity for the three clusters at station in the network.

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 2-5 and Figure 2-6, 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 2-6 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 BSS operators to manage the bike stations and propose temporal and spatial plans. The clustering results give operators a general view of the status of stations and clarify where the imbalances would occur with respect to time of day and day of week, leading to better monitoring of the system as a whole.

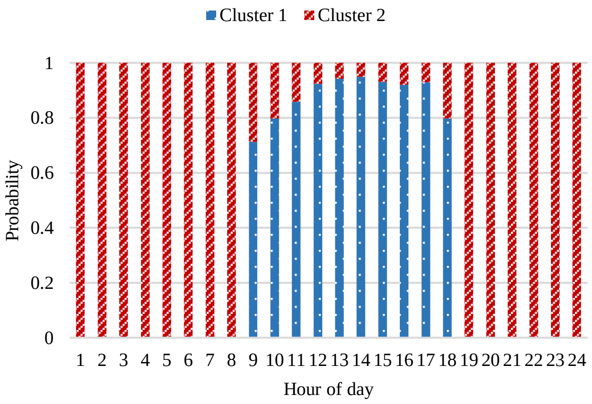


Figure 2-5. Probability of hour being in one of the two clusters (*K* = 2).

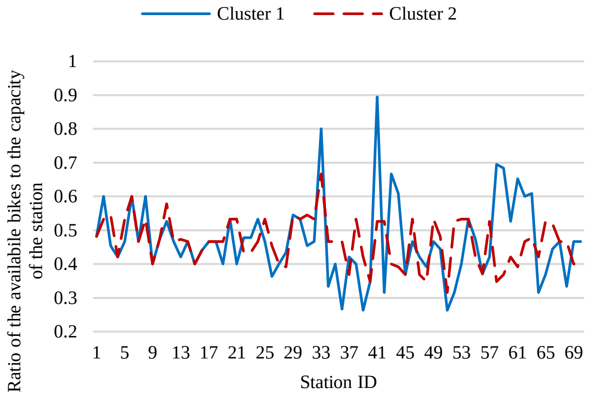


Figure 2-6. Available bikes of the two clusters for each station in the network.

2.9 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 and similarity of the BSS dataset. 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 3. Quantifying the Effect of Various Features on the Modeling of Bike Counts in a Bike-Sharing System

3.1 Introduction

A growing population, with more people living in cities, has led to increased pollution, noise, congestion, and greenhouse gas emissions. One possible approach to mitigating these problems is encouraging the use of BSSs. BSSs are an important part of urban mobility in many cities and are sustainable, environmentally-friendly systems. As urban density and its related problems increase, it is likely that more BSSs will exist in the future. The relatively low capital and operational cost, ease of installation, existence of pedal assistance for people who are physically unable to pedal for long distances or on difficult terrain, and better tracking of bikes are some of the properties that strengthen this prediction (DeMaio, 2009).

One of the first BSSs in the U.S. came into existence in 1994 with a small bike sharing program in Portland, which had only 60 bicycles available for public use. At present, although the BSS experience is still relatively limited, many cities, such as San Francisco and New York, have launched programs to serve users using different payment structures and conditions. One of the largest information technology (IT)-based systems, based in Montreal, Canada, is BIXI (BIcycle-TaXI), which employs the concept of using a bicycle like a taxi. In fact, this system, with its use of advanced technologies for implementation and management, illustrates a shift into the fourth generation of BSSs (Susan, Stacey, & Hua, 2010).

In 2013, San Francisco launched the Bay Area Bike Share BSS, a membership-based system providing 24 hours a day, 7 days a week self-service access to short-term rental bicycles. A detailed description of the BSS is provided in the introduction to this report.

This chapter proposes an approach to constructing a bike count model for the San Francisco Bay Area BSS. The count of bikes in each station, each of which has a finite number of docks, fluctuates. Thus, a repositioning (or redistribution) operation must be performed periodically to meet this fluctuation. Coordinating such a large operation is complicated, time consuming, polluting and expensive (DeMaio, 2009). Predicting the available number of bikes in each station over time is one of the key tasks to making this operation more efficient. Moreover, this chapter attempts to quantify the effect of several variables on the bike count model for each station in the Bay Area BSS network, including the significance of the 70 stations, the month-of-the-year, the day-of-the-week, time-of-day, and various weather conditions.

In terms of the chapter layout, following the introduction, this chapter is organized into five sections. First, related work, focused on the proposed model in previous studies, is discussed. Next, a background of count model regression, RF, and BIC are presented. Third, the different datasets used in this study are described. In the fourth section, the details of the data analysis used to construct a predictive bike count model are provided. Finally, the chapter concludes with a summary of new insights and recommendations for future bike count model research.

3.2 Related Work

The modeling of bike sharing data using various features, including time, weather, built-environment, transportation infrastructure, etc., is an area of significant research interest. In general, the main goals of data modeling are to boost the redistribution operation (Contardo et al., 2012; Raviv, Tzur, & Forma, 2013; Schuijbroek, Hampshire, & van Hoeve, 2013), to gain new insights into and correlations between bike demand and other factors (David William Daddio, 2012; Rixey, 2013; Rudloff & Lackner, 2013a; X. Wang, Lindsey, Schoner, & Harrison, 2015), and to support policy makers and mangers in making optimized decisions (David William Daddio, 2012; Vogel, Greiser, & Mattfeld, 2011b). Generally, the main approach to modeling and predicting bike sharing data is regression count modeling. A recent study modeled the demand for bikes and return docks using data from the BSS Citybike Wien in Vienna, Austria. The influence of weather (temperature and precipitation) and full/empty neighboring stations on demand was studied using different count models (Poisson, negative binomial [NB] and hurdle). The authors found that although the hurdle model worked best in modeling the demand of bike sharing stations, these models were complex and might not be ideal for optimization procedures. They also found that NB models outperformed Poisson models because of the dispersion issue in the data, which will be discussed later in this chapter (Rudloff & Lackner, 2013a). However, an early study used count series to predict the stations' usage based on Poisson mixtures, providing insight into the relationship between station neighborhood type and mobility patterns (Come Etienne & Oukhellou Latifa, 2014a).

In a study by Wang et al., log-linear and NB regression models were used to estimate total station activity counts. The factors they used mostly had economic; built-environment; transportation infrastructural; and social aspects, such as neighborhood sociodemographic (i.e., age and race), proximity to the central business district, proximity to water, accessibility to trails, distance to other bike share stations, and measures of economic activity. All the variables were found to be significant. Log-likelihood was used as a measure of the goodness of fit of the Poisson and NB models (X. Wang et al., 2015). Linear least regression with data from the on-the-ground Capital Bikeshare system was implemented in another study to explain station demand based on the demographic, socioeconomic, and built-environment characteristics (Daddio, 2012).

Several studies used methods other than count models to model bike sharing data. A multivariate linear regression analysis was used in another study to study station-level BSS ridership. That study investigated the correlation between BSS ridership and the following factors: population density; retail job density; bike, walk, and transit commuters; median income; education; presence of bikeways; nonwhite population (negative association); days of precipitation (negative association); and proximity to a network of other BSS stations. The authors found that the demographic, built environment, and access to a comprehensive network of stations were critical factors in supporting ridership (Rixey, 2013).

A study by Gallop et al. used continuous and year-round hourly bicycle counts and weather data to model bicycle traffic in Vancouver, Canada. The study used seasonal autoregressive integrated moving average analysis to account for the complex serial correlation patterns in the error terms and tested the model against actual bicycle traffic counts. The study demonstrated that the weather had a significant and important impact on bike usage. The authors found that the weather data (namely temperature, rain, humidity, and clearness) were generally significant; temperature and rain, specifically, had an important effect (Gallop, Tse, & Zhao, 2011).

It is also worth noting that some studies used methods other than regression to either model BSS data or to develop new insights and understandings of BSSs (see (Contardo et al., 2012; Vogel et al., 2011b). For example, a mathematical formulation for the dynamic public bike-sharing balancing problem was introduced using two different models: arc-flow formulation and Dantzig-Wolfe decomposition formulation. The demand was computed by considering the station either a pickup or delivery point, with a real-time and length period between two stations (Contardo et al., 2012).

3.3 Methods

3.3.1 Count Models

In the model used for this study, the outcomes y_i (bike count in our prediction model) are discrete non-negative integers representing the number of available bikes at a specified time at each station in the network. Count models based on generalized linear models (GLMs) were applied. Specifically, two models were used to predict the bike count $(1 - demand)$ in the network: the Poisson regression model (PRM), and the NB regression model (NBRM). Following are brief descriptions of these two models; more details can be found in the literature (Cameron & Trivedi, 2013; Long & Freese, 2006).

3.3.2 Poisson Regression Model (PRM)

In the PRM, each observation i is allowed to have a different value of mean μ, where μ_i is estimated from recorded characteristics. The PRM assumes that y has a Poisson distribution, and its logarithm (i.e., link function) can be modeled by a linear combination of parameters. However, the Poisson distribution assumes that the mean and variance are equal $Var(y) = \mu$. If this condition is not met, there is an over-dispersion in the data, implying that more complex models need to be applied. The probability density for the PRM is

$$f(y, \mu) = \frac{\exp(-\mu)\mu^y}{y!} \qquad (3\text{-}1)$$

The GLM of the mean μ on a vector predictors x_i is formulated as

$$\log(\mu_i) = \beta_i x_i^T \qquad (3\text{-}2)$$

where β_i are the estimated regression coefficients and $\log(\mu_i)$ is the natural logarithm.

3.3.3 Negative Binomial Regression Model (NBRM)

The NBRM is considered a generalization of PRM. It is based on a Poisson-gamma mixture distribution that assumes that the count y_i is dependent on two parameters: the mean μ_i and some dispersion parameter θ. It basically loosens the assumption in PRM that the variance is equal to the mean and adjusts the variance independently. In fact, the Poisson distribution is a special case of the NB distribution. The probability density for the NBRM is

$$f(y,\mu) = \frac{\Gamma(y+\theta)}{\Gamma(\theta)y!}\frac{\mu^y\theta^\theta}{(\mu+\theta)^{y+\theta}} \tag{3-3}$$

The GLM of the mean μ on a vector predictors x_i is formulated as

$$\log(\mu_i) = \beta_i x_i^T \tag{3-4}$$

where β_i are the estimated regression coefficients and $\log(\mu_i)$ is the natural logarithm.

3.3.4 PRM vs NBRM

The Poisson distribution assumes that the mean and variance are the same. However, occasionally, the data shows that variance might be higher or lower than the mean. This situation is called over-dispersion/under-dispersion and NBRM is able to accommodate it. The NB distribution has an additional parameter to the Poisson distribution, which adjusts the variance independently from the mean. In fact, the Poisson distribution is a special case of the NB distribution. Thus, the PRM and the NBRM have the same mean structure, but the NBRM has one parameter more than the PRM to regulate the variance independently from the mean. As Cameron and Trivedi explain,

> if the assumptions of the NBRM are correct, the expected rate for a given level of the independent variables will be the same in both models. However, the standard errors in the PRM will be biased downward, resulting in spuriously large z-values and spuriously small p-values (Cameron & Trivedi, 1986, 2013).

3.3.5 Random Forest (RF)

One of the characteristics of this type of dataset is that it is often very large. It is therefore crucial to implement machine learning to identify potential explanatory variables (Vogel et al., 2011b). Moreover, when a model contains a large number of predictors it becomes more complex and overfitting can occur. To avoid this, the RF, as introduced by Breiman in 2001 (Breiman, 2001), was applied. RF creates an ensemble of decision trees and randomly selects a subset of features to grow each tree. While the tree is being grown, the data are divided by employing a criterion in several steps or nodes. The correlation between any two trees and the strength of each individual tree in the forest affect, also known as the forest error rate in classifying each tree. Practically, the mean squared error of the responses is used for regression.

The fact that in RF each tree is constructed using a different bootstrap sample from the original data ensures that the RF extracts an unbiased estimate of the generalization error. This is called the OOB (out-of-bag) error estimate, which can be used for model selection and validation without the need of a separate test. The OOB was used to validate the significance of the subsequent inference of each parameter in this study. The RF technique offers several advantages. For example, it runs efficiently with a large amount of data and many input variables without the need to create extra dummy variables; it can handle highly nonlinear variables and categorical interactions; and it ranks each variable’s individual contributions in the model. However, RF also has a few limitations. For instance, the observations must be independent, which is assumed in our case. Moreover, model interpretation after averaging many tree models is generally more difficult than interpreting a single-tree model. However, this is not relevant to our model, as it was used only for ranking the predictors. For more details see (Breiman, 2001; Loh, 2011).

In this study, RF was used as a technique to rank the effect of the different parameters in the model. This rank was exploited as a systematic guide in the forward step-wise technique. Performing a direct stepwise regression for a BSS is difficult, as there are many predictors involved in the process, which is time consuming, expensive, and requires expensive statistical software (for example, see (David William Daddio, 2012)). Therefore, we employed the BIC (discussed in the next section) to choose the most accurate model while maintaining model simplicity. We started by modeling the most important parameter resulting from RF as the only explanatory variable (i.e., the regressor). Then, forward step-wise regression was applied and the log-likelihood was found and applied to determine the accumulated BIC.

3.3.6 Bayesian Information Criterion (BIC)

BIC was the criterion selected to compare between models following a forward step-wise regression guided by the results of RF. In general, the model with the lowest BIC is preferred. However, since there were 111 predictors, the result was a set of 111 models. Adding predictors may increase the log-likelihood, leading to overfitting, and log-likelihood does not take into account the number of predictors. BIC makes up for the number of predictors in the model by introducing a penalty term. Given that $\hat{L}$ is the maximum likelihood, n is number of observations, and k is the number of predictors, BIC is defined as (Wit, Heuvel, & Romeijn, 2012).

$$BIC = -2.\ln \hat{L} + k.\ln(n) \qquad (3\text{-}5)$$

As shown in the equation, $k.\ln(n)$ is the term to make up for the number of predictors in each model.

3.4 Dataset

This study used anonymized bike trip data collected from August 2013 to August 2015 in the San Francisco Bay Area, as shown in Figure 3-1 (Hamner, 2016). This study used two datasets. The first dataset included station ID, number of bikes available, number of docks available, and time of recording. The time data included year, month, day-of-the-month, time-of-day, and minutes at which an incident was recorded. As an incident was documented every minute for 70 stations in San Francisco over 2 years, this dataset contains a large number of recorded incidents. This dataset was exposed to a change detection process to determine times when a change in bike count occurred in each station. From this dataset, as a result of pre-processing, the station ID, number of bikes available, month, day-of-the-week, and time-of-day were extracted for use as a feature. Subsequently, each station’s zip code was assigned and input to the set. A histogram of the bike counts is shown for all stations resulting from the change detection process (Figure 3-2). The histogram is considerably skewed to the right, which means that the mean, median, and mode are markedly different, indicating a dispersion in the counts.

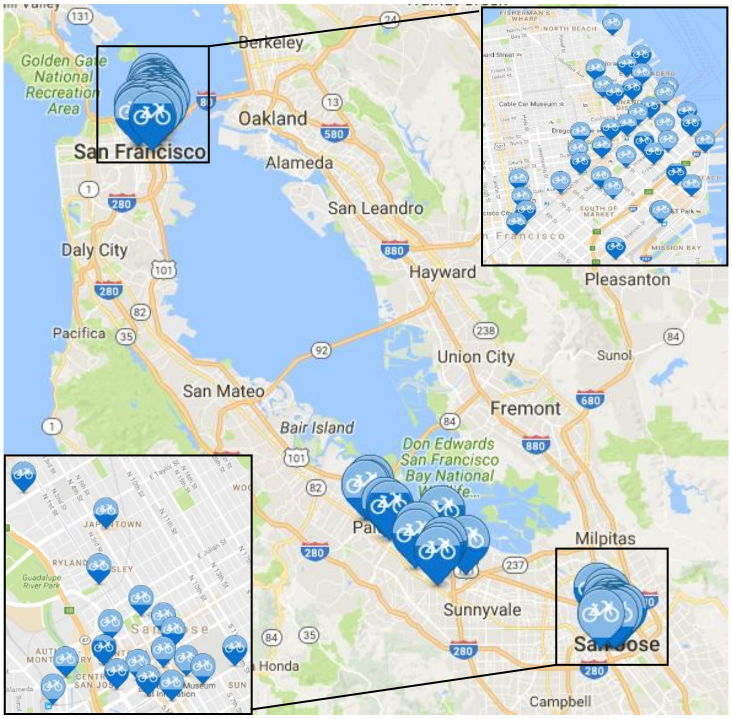


Figure 3-1. Locations of the 70 stations covering five cities: San Francisco, Palo Alto, Mountain View, Redwood City, and San Jose (2016).

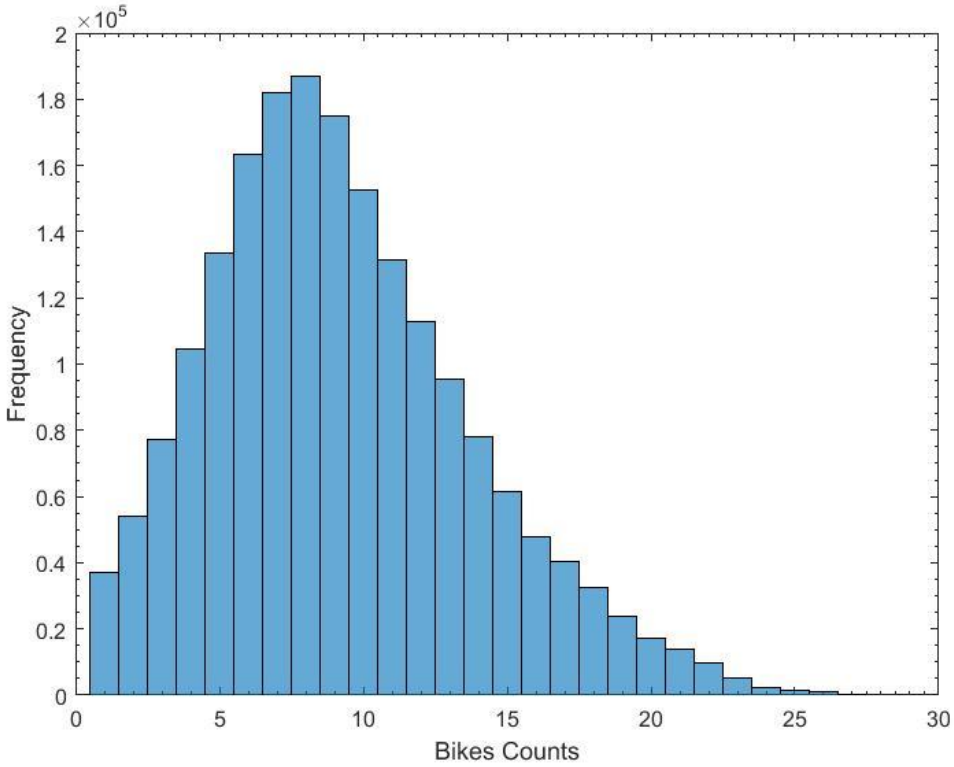


Figure 3-2. Histogram of bike counts.

The second dataset contains different attributes: the date (in month/day/year format), zip code, and other variables describing the daily weather for each zip code over the 2-year period. Daily weather data at each zip code contains information about temperature, humidity, dew, sea level pressure, visibility, wind speed and degree, precipitation, cloud cover, and events for that day (i.e., rainy, foggy or sunny). The minimum, maximum, and mean of the first six attributes of the weather information are recorded in this dataset. This dataset was used to match the daily weather attributes with the first dataset utilizing the two mutual attributes between them: date and zip code. The matched weather data was concatenated with the first dataset.

3.5 Data Analysis and Results

The following subsections present the methodology and the results of the data analysis. MATLAB was used in implementing the count regression models—Poisson and NB, RF, and BIC.

3.6 Problem Definition and Formulation

We assumed there was no interaction between the 70 stations and thus, we used station dummies in modeling for two reasons: (1) the main goal of this chapter was to introduce an effective and fast, but also accurate and reasonable, approach to quantifying the effect of various features on bike counts at different stations. Investigating other variables, such as the relations between different stations (station neighbors), would have required a great deal of effort and may have added some distraction to the goal of the analysis. (2) It was suggested that one of the goals of the analysis was studying the possibility of pooling all of the variables in one model instead of dealing with 70 models for each station. This method could be reasonable and effective in cases of large networks and would not require high prediction accuracy at specific stations.

As we assumed there was no interaction between the 70 stations, the $\log(\mu)$ of the bike count in each station might be represented as parallel hyperplanes. In order to construct one model containing all the stations instead of a model for each station, 69 indicator variables were coded as the 70 stations in the network, which implies that Station 1 is the reference in the model intercept. Similarly, 11 indicators were coded for the 12 months with January as a reference, six indicators for the seven days of the week were coded with Sunday as a reference, and two indicators for the events in the day were coded with sunny as a reference. All of these indicators were pooled in one model. If there was no significant difference between two of the parameters (say for example β_1 and β_2), this meant that the corresponding two parallel hyperplanes (Station 1 and Station 2) were very close to each other and the predicted $\log(\mu)$ of the bike count was the same for the two stations to an acceptable level of accuracy.

The first step in understanding the bike count's behavior was to regress all the available predictors to generate a full model. To that end, the PRM and NBRM were applied. The next step was using RF to rank the predictors in the full model based on the OOB error. Forward step-wise regression was then used to fit several models that were constructed as a result of RF. Finally, BIC was used to select the best model, or, in other words, the best subset of predictors to construct this model.

However, this subset of predictors still had to be evaluated to determine whether they were reasonable. To accomplish this, all the parameters were examined and it was determined which were most acceptable. Different stations, month-of-the-year, day-of-the-week, and time-of-day were all determined to be reasonable parameters that might affect the model. From the weather information, mean temperature, mean humidity, mean visibility, mean wind speed, precipitation, and events were selected for further investigation. These parameters were selected based on subject-matter expertise, previous related studies (see for example (Gallop et al., 2011; Rudloff & Lackner, 2013a), and to avoid multicollinearity between two or more predictors. Once again, RF and forward step-wise regression were repeated and BIC was used to compare the built models. We chose the model with the best compromise between the minimum BIC value and the consideration of the effective parameters.

3.7 Count Regression Models

As described earlier, two count models were used: Poisson and NB. To compare them, log-likelihood was estimated to determine goodness of fit. The likelihood of a set of parameter values is equal to the probability of the observed outcomes given those parameter values (Johansen & Juselius, 1990). The following table shows the log-likelihood of Poisson and NB for the full model (Table 3-1). As NB was able to accommodate the over-dispersion/under-dispersion in the data, its log-likelihood was higher than Poisson's. This meant that NB was better than Poisson at describing the available bikes in the network. As a result, the NBRM was selected for use in all following steps in the analysis.

Table 3-1. Log-likelihood of Poisson and NB models.

	Poisson	**NB**
Log-likelihood	-5.95E+06	-5.61E+06

3.8 Random Forest and Bayesian Information Criterion

Both RF and BIC were applied twice in this study. RF was applied on all the available predictors, constructing 111 different models. Basically, RF was implemented to sort the predictors in descending order of their "importance." MATLAB's manual describes this RF measurement as

> an array containing a measure of importance for each predictor variable (feature). For any variable, the measure is the increase in prediction error if the values of that variable are permuted across the out-of-bag observations. This measure is computed for every tree, then averaged over the entire ensemble and divided by the standard deviation over the entire ensemble (2016).

Importance was utilized as a guide in forward step-wise regression using the NBRM, and computing the log-likelihood following each addition. BIC was then computed from the log-likelihood.

The BIC results of this first process are presented as the orange line shown in the following figure (Figure 3-3). As the number of inserted predictors increased in the model, the BIC value decreased, indicating a better model. The BIC curve was used to select the most influential predictors resulting in the lowest BIC value. There was no specific rule for selecting those predictors, but rather it was a trade-off between the best and most simple model. The elbow in the curve, which corresponds to 45 predictors, was chosen to achieve the best compromise. The selected subset contained features of 31 stations, 7 months, 5 days, time-of-day, and one weather variable (wind direction degree). Based on subject-matter expertise and knowledge gained from related studies, it was determined that this subset was largely unacceptable. For example, temperature, not included in the subset, was found to be significant in previous studies of modeling bike counts in (Rudloff & Lackner, 2013a).

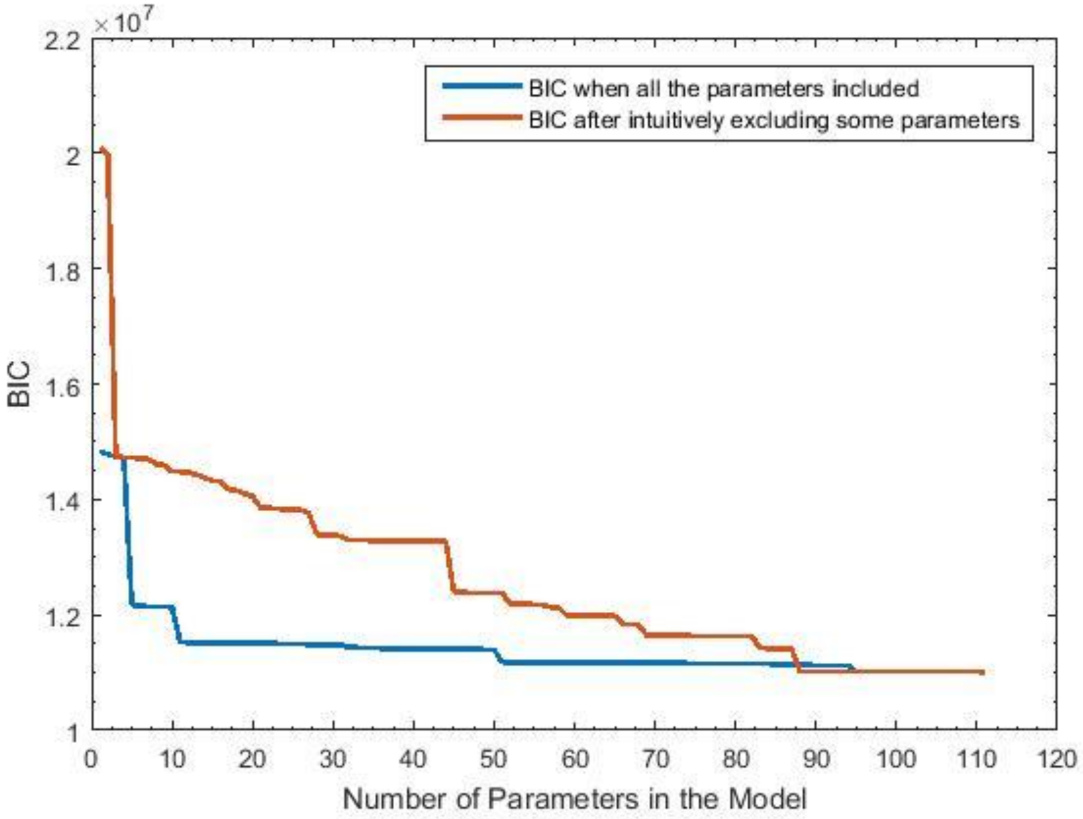


Figure 3-3. BIC before and after feature selection process.

This first conclusion led to a re-evaluation of the predictors by closely examining the weather information variables to determine any correlation among them. Again, based on expertise and related studies, mean temperature, mean humidity, mean visibility, mean wind speed, precipitation, and events were selected as predictors. RF and BIC were again applied after the predictor selection process. The importance of the predictors resulting from the RF is shown in the following figure (Figure 3-4[a]) and the result of the BIC following forward step-wise regression is represented by the blue line in the previous figure (Figure 3-3). As the previous figure illustrates, selecting these features improves BIC values remarkably. This is mainly because RF obtained a different order of predictors after neglecting any features that might correlate with other parameters. For example, maximum and minimum temperatures were correlated with the mean temperature. Maximum and minimum temperatures were neglected and the mean temperature remained.

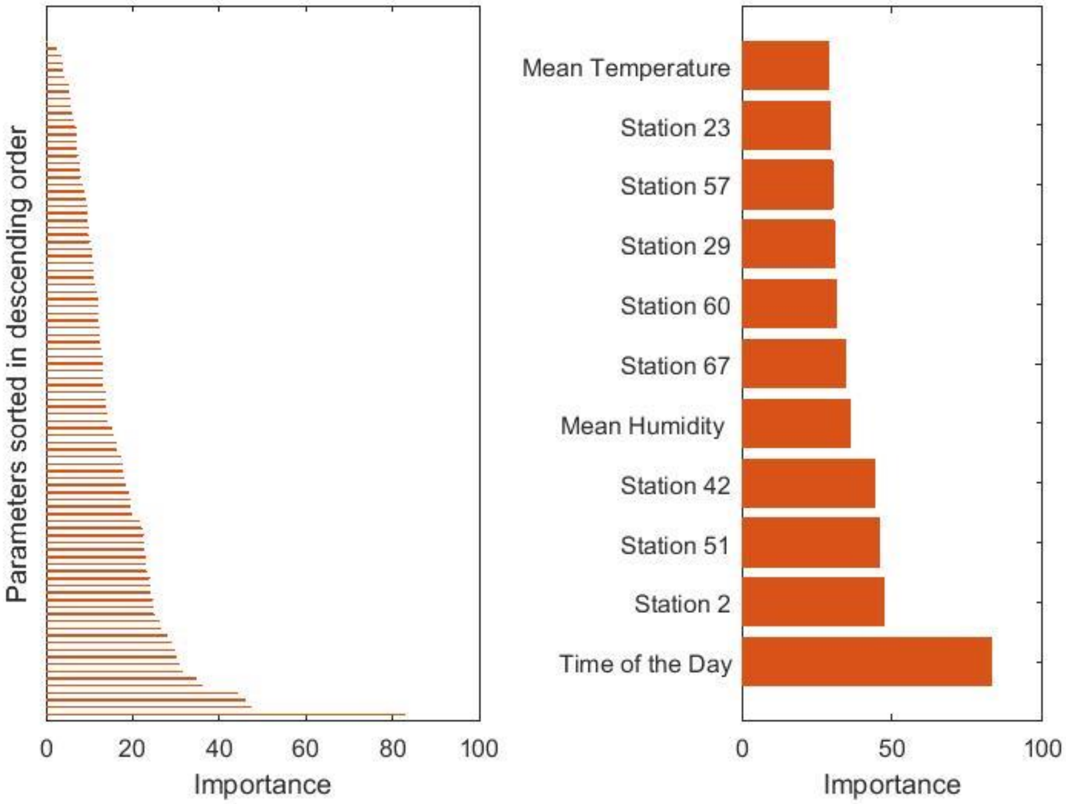


Figure 3-4. Importance of the predictors (a) after feature selection (b) of the first proposed solution.

The BIC curve after feature selection revealed that two elbows could be selected as two proposed solutions that might achieve the best compromise: the first using 11 predictors, the second using 51 predictors. As the simplest explanation is preferable, the first solution was selected as the final model. The figure above (Figure 3-4[b]) shows the importance of these 11 predictors, which are clearly reasonable. Temperature and humidity turned out to be important features and have significant effects in predicting bike availability in the Bay Area Bike Share network. San Francisco is one of the most humid cities in the U.S., with an average humidity of nearly 74% ("Most Humid Cities in USA - Current Results," 2016). Humidity has been proven to be a discomfort to people, particularly during physical activities like riding a bicycle.

Although we chose the first solution, it is worth noting that if we had selected the second solution, another two weather variables (visibility and wind speed), some days of the week, and some months would be included in the 51 most important predictors. All of these predictors are also reasonable and important in predicting bike availability in the San Francisco network.

The final model was constructed for the first solution, applying NBRM with a log-likelihood and BIC of -5.56E+06 and 1.12E+07 respectively. The following table shows the estimated parameter values for the NB Model of bike availability in the San Francisco network (Table 3-2). It also shows that all the parameters are significant since the p-values are approximately equal to zero.

Table 3-2. Estimated parameter values for the NB model for bike availability in the network.

	Estimate	**P-value**
Intercept	2.226865	< 0.0001
Time-of-day	-0.00050	< 0.0001
Station 2	0.467929	< 0.0001
Station 51	0.411846	< 0.0001
Station 42	0.290969	< 0.0001
Humidity	0.000516	< 0.0001
Station 67	0.428846	< 0.0001
Station 60	0.186177	< 0.0001
Station 29	2.56E-01	< 0.0001
Station 57	0.217112	< 0.0001
Station 23	0.290833	< 0.0001
Temperature	-0.0013	< 0.0001

3.9 Conclusions and Recommendations for Future Work

In this chapter, we described the development of a bike availability model for the San Francisco Bay Area Bike Share program. Since the demand of bikes in stations is still not well studied, this chapter introduced an effective and fast, but also accurate and reasonable, approach to quantifying the effect of various features on bike counts at different stations. The results revealed that the bike count changes with the month-of-the-year, day-of-the-week, time-of-day, and some weather variables. This model could also be used to improve the redistribution of bicycles, which is important for rebalancing the network over a period of time.

NBRM and PRM were performed on the bike count data. NBRM was ultimately chosen, as it was found to best fit the count data. However, the significance measure in NBRM (i.e., p-value) resulting from the regression process was not always an adequate measure, especially when there were a large number of features and if there was a possible correlation. As a result, this study adopted a new method consisting of feature selection, RF to run the predictors, guided forward step-wise regression of these predictors, and BIC to compare between models. This method turned out to be an effective and reasonable approach to identify critical predictors of bike counts.

The final results reveal interesting new insights. Firstly, mean humidity as a predictor for bike counts has not been investigated in previous studies. Results of this study demonstrate that humidity is a significant predictor in the Bay Area Bike Share program. Further, although precipitation has been shown to be significant in many previous studies, the results of this study demonstrate that precipitation is not a significant predictor in San Francisco. Over the entire year, the most common forms of precipitation in San Francisco were light rain, moderate rain, and drizzle, none of which appeared to have a major effect on Bay Area Bike Share use. The contrast between this finding and that of previous studies indicates that particular weather information may have different significance depending on the studied geographic area.

Secondly, eight indicator variables corresponding to eight stations and one variable serving as a reference in the intercept were selected as final predictors in the model. This implies that the bike count data for the remaining 61 indicator variables corresponding to 61 stations were not significantly different from the bike count data for the reference station. The variability in bike counts of these 61 stations would not be influential if the data were employed as predictors in the regression. Nonetheless, the eight stations were different from the reference station to an extent that might largely affect the prediction if not considered as predictors at all. This is because of these station locations. For example, one station is near the main train station in Palo Alto, which is the second busiest station in the Caltrain system; another is near Yerba Buena Center for the Arts in San Francisco; one is at Union Square, which is a busy public square in the center of San Jose; and one is at the San Antonio Caltrain station in Mountain View.

Finally, time-of-day was found to be one of the most important predictors. This means that the bike count fluctuates over the course of the day (i.e., during peak and off-peak hours).

The adopted approach needs to be further validated by applying it to other bike count data in different geographic areas. It is also important to investigate other variables, such as bikes coming from other stations and the relative location of each station.

Chapter 4. Identifying Optimum Bike Station Initial Conditions using Markov Chain Modeling

4.1 Introduction

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 aforementioned 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.

BSSs suffer from a central recurring imbalance problem, meaning many bike stations either become empty or full during their daily operation. We hypothesize that the cost of balancing the bike stations can be reduced by optimizing the number of bikes at each station at the start of the day, thus reducing the need for a dynamic balancing system (Lu, 2016; Raviv & Kolka, 2013; J. Schuijbroek, R. C. Hampshire, & W. J. van Hoeve, 2017). We formulate our hypothesis by modeling each station using a Markov chain.

4.2 Methods And Data

This study uses the San Francisco Bay Area Bike Share (now Ford GoBike) docking station data collected from August 2013 to August 2015 in the San Francisco Bay Area (see Figure 3-1 in the previous chapter). The dataset is described in detail in Section 3.4 of this report.

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) was conducted per minute, making the movement between states as smooth as possible throughout the hour.

The probability distribution of the available bikes at a particular station at the end of the day is shown in (4-1).

$$\mathrm{P}\left(\mathrm{x}^{\text{end of the day}} = \mathrm{q} \middle| \mathrm{x}^{\text{start of the day}} = \mathrm{m}\right) =$$

$$\left(\mathrm{P}\left(\mathrm{x}^{\text{end of the first hour}}=\mathrm{q}\middle|\mathrm{x}^{\text{start of the first hour}}=\mathrm{m}\right)\right)*\prod_{\mathrm{h}=2}^{\text{last hour of the day}}\mathrm{P}_{\mathrm{h}} \qquad (4\text{-}1)$$

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

Equation (4.1) 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 (4-2).

$$max_i \sum_{h=1}^{24} W_h \sum_{j=N_{min}}^{N_{max}} P_{ijh} \qquad (4\text{-}2)$$

Where i is the initial condition of station s, h is the hour of the day (considered only the hours from 6 a.m. to 8 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.

4.3 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 (Ashqar et al., 2017).

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 BSS stations 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 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. Modeling Bike Availability in a Bike-Sharing System Using Machine Learning

In this chapter, we provide a toolbox 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 using three different approaches as follows.

5.1 Introduction

A growing population, with more people living in cities, has led to increased pollution, noise, congestion, and greenhouse gas emissions. One possible approach to mitigating these problems is encouraging the use of BSSs. BSSs are an important part of urban mobility in many cities and are sustainable and environmentally friendly. As urban density and its related problems increase, it is likely that more BSSs will exist in the future due to relatively low capital and operational costs, ease of installation, pedal assistance for people who are physically unable to pedal for long distances or on difficult terrain, and better tracking of bikes (DeMaio, 2009).

A detailed description of BSSs implementation over time and the structure and history of the San Francisco Bay Area Bike Share program, from which the data used in this work was collected, is provided in the introduction to Chapter 3 of this report.

This chapter proposes an approach to modeling the number of available bikes at a BSS using machine learning. Since the number of available bikes at a station, which has a finite number of docks, fluctuates, a repositioning (or redistribution) operation must be performed periodically. Coordinating such a large operation is complicated, time-consuming, polluting, and expensive (DeMaio, 2009). Predicting the number of available bikes in each station over time is one of the key tasks to making this operation more efficient. In this study, RF and least-squares boosting (LSBoost) algorithms were used to build univariate prediction models for available bikes at each Bay Area Bike Share station. However, to reduce the number of required prediction models for the entire BSS network, we also used partial least-squares regression (PLSR) as a multivariate regression algorithm.

Following the introduction, this chapter is organized into five sections. Section 5.2 briefly discusses related work from the literature, focusing on the methods proposed in previous studies. Next, a background of the regression models used is presented in Section 5.3. In Section 5.4, the different datasets used in this study are described. The details of the data analysis used to construct predictive models of the number of available bikes are provided in Section 5.5. Finally, the chapter concludes with a summary of new insights and recommendations for future research on modeling the number of available bikes.

5.2 Related Work

The modeling of bike sharing data is an area of significant research interest. Proposed models have relied on various features, including time, weather, the built environment, and transportation infrastructure. In general, the main goals of these models have been to boost the redistribution operation (Contardo et al., 2012; Raviv et al., 2013; Schuijbroek et al., 2013), to gain new insights into and correlations between bike demand and other factors (David William Daddio, 2012; Rixey, 2013; Rudloff & Lackner, 2013b; X. Wang et al., 2015), and to support policy makers and managers in making optimized decisions (David William Daddio, 2012; Vogel et al., 2011b).

Froehlich, Neumann, and Oliver used four predictive models to predict the number of available bikes at each station: last value, historical mean, historical trend, and Bayesian network (Froehlich et al., 2009b). Two methods for time series analysis, autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA), have also been used to predict the number of available bikes/docks for each bike station. Kaltenbrunner, Meza, Grivolla, Codina, and Banchs adopted ARMA (Kaltenbrunner et al., 2010); Yoon, Pinelli, and Calabrese proposed a modified ARIMA model considering spatial interaction and temporal factors (Yoon, Pinelli, & Calabrese, 2012). However, Gallop, Tse, and Zhao used continuous and year-round hourly bicycle counts and weather data to model bicycle traffic in Vancouver, Canada (Gallop et al., 2011). That study used seasonal autoregressive integrated moving average analysis to account for the complex serial correlation patterns in the error terms and tested the model against actual bicycle traffic counts. The results demonstrated that the weather had a significant and important impact on bike usage. The authors found that the weather data (temperature, rain, humidity, and clearness) were generally significant; temperature and rain, specifically, had an important effect.

A multivariate linear regression analysis was used by Rixey to study station-level BSS ridership (2013). That study investigated the correlation between BSS ridership and the following factors: population density; retail job density; bike, walk, and transit commuters; median income; education; presence of bikeways; nonwhite population (negative association); days of precipitation (negative association); and proximity to a network of other BSS stations. The author found that demographics, the built environment, and access to a comprehensive network of stations were critical factors in supporting ridership.

This chapter makes two major contributions to the literature. First, the univariate response models that have been used previously to predict the number of available bikes at each station ignore the correlation between stations and might become hard to implement when applied to relatively large networks. Thus, this chapter investigates the use of multivariate response models to predict the number of available bikes in the network. Second, station neighbors, which are determined by a trip's adjacency matrix, are considered as significant predictors in the regression models.

5.3 Methods

In this section, we will briefly describe the three machine learning algorithms used in this chapter: RF, LSBoost, and PLSR.

5.3.1 Random Forest (RF)

Breiman proposed RF as a new classification and regression technique in supervised learning (Breiman, 2001). RF creates an ensemble of decision trees and randomly selects a subset of features to grow each tree. While the tree is being grown, the data are divided by employing a criterion in several steps or nodes. The correlation between any two trees and the strength of each individual tree in the forest affect the forest error rate in classifying each tree. Practically, the mean squared error of the responses is used for regression.

RF offers several advantages (Breiman, 2001; Loh, 2011). For example, there are very few assumptions attached to its theory; it is considered to be robust against overfitting; it runs efficiently and relatively quickly with a large amount of data and many input variables without the need to create extra dummy variables; it can handle highly nonlinear variables and categorical interactions; and it ranks each variable's individual contributions in the model. However, RF also has a few limitations. For instance, the observations must be independent, which is assumed in our case.

5.3.2 Least-Squares Boosting (LSBoost)

LSBoost is a gradient boosting of regression trees that produces highly robust and interpretable procedures for regression. LSBoost was proposed by Friedman as a gradient-based boosting strategy (J. H. Friedman, 2001), using square loss $L\ (y, F) = \ (y\ -\ F)^2/2$, where F is the actual training and y is the current cumulative output $y_i\ =\ \beta_0\ + \sum_{j=1}^{i-1} \beta_j h_j\ + \beta_i h_i\ =\ y_{i-1} + \beta_i h_i$. The new added training $\hat{F}$ is set to minimize the loss, in which the training error is computed as in (Barutçuoğlu & Alpaydın, 2003):

$$E = \sum_{t=1}^{N} [\beta_i h_i^t - \hat{F}^t] \tag{5-1}$$

where $\hat{F}$ is the current residual error and the combination coefficients β_i are determined by solving $\partial E / \partial \beta_i = 0$.

In this chapter, RF and LSBoost were used as univariate regression techniques to model the number of available bikes in each station at any time t. RF and LSBoost are ensemble learning algorithms, which integrate multiple decision trees to produce robust models. However, the main difference between these two algorithms is the order in which each component tree is trained. Using randomness, RF trains each tree independently, whereas LSBoost trains one tree at a time and each new added tree is set to correct errors made by previously trained trees. The ensemble model is produced by synthesizing results from the individual trees.

5.3.3 Partial Least-Squares Regression (PLSR)

PLSR was recently developed as a multivariate regression algorithm (Geladi & Kowalski, 1986; Höskuldsson, 1988; H. Wold, 1982; S. Wold, Ruhe, Wold, & Dunn, 1984; S. Wold, Sjöström, & Eriksson, 2001). PLSR finds a linear regression model by projecting the predicted variables Y and the observable variables X to a new space. The basic model in the PLSR method consists of a regression between two blocks (i.e. $X \text{ and } Y$). Furthermore, this model contains outer relations for each of the X and Y blocks, and an inner relation that links both blocks. PLSR has several advantages. For example, it is suitable when the matrix of predictors Y has more variables than observations, and when there is multicollinearity among observable variable X values. Moreover, the PLSR method outperforms multiple linear regressions because implementing PLSR develops stable predictors. In this chapter, PLSR was used as multivariate regression to reduce the number of required prediction models for the number of available bikes at any time t for the entire BSS network.

5.4 Dataset

This study used anonymized bike trip data collected from August 2013 to August 2015 in the San Francisco Bay Area (refer to Figure 3-1) (Hamner, 2016). This study used two datasets. See Section 3.4 of this report for a detailed description of the first dataset.

The second dataset contained different attributes: the date (in month/day/year format), zip code, and 22 other variables describing the daily weather for each zip code over the 2-year period. The number of available bikes at station i at time t, the number of available bikes at its neighbors at the same time t, month of the year, day of the week, and time of day were all extracted from the two datasets as parameters that affect the model. Specifically, the neighbors of a station i were defined based on the number of trips originated from station j, in which $j \neq i$, and ended at station i. In that sense, we generated the adjacency matrix of the BSS network and found the highest 10 in-degree stations for station i, which were assigned as neighbors of station i. In addition, an unpublished work by the authors (Huthaifa I. Ashqar, Elhenawy, Ghanem, Almannaa, & Rakha, 2016) investigated various weather data as predictors to determine the reasonable parameters that mainly affect the prediction models. From the weather information, mean temperature, mean humidity, mean visibility, mean wind speed, precipitation, and events in a day (i.e., rainy, foggy, or sunny) were all selected. These parameters were selected based on subject-matter expertise and previous related studies (Gallop et al., 2011; Rudloff & Lackner, 2013b), and they were found to be significant in predicting the number of available bikes at Bay Area Bike Share stations (Huthaifa I. Ashqar et al., 2016).

5.5 Data Analysis and Results

5.5.1 Univariate Models

RF and LSBoost algorithms were applied to create univariate models to predict the number of available bikes at each of the 70 stations of the Bay Area Bike Share network. The two algorithms were applied to investigate the effect of several variables on the prediction of the number of available bikes in each station i in the network, including the available bikes at station i at time t, the available bikes at its neighbors at the same time t, the month of the year, day of the week, time of day, and various selected weather conditions. The predictors' vector for station i at time t, denoted by X_t^i, was used in the built models to predict the log of the number of available bikes at station i at time t and at a prediction horizon time, denoted by $\log\left(y_{t+\Delta}^i\right)$, where $i = 1, 2, \ldots, 70$. The effect of different prediction horizons, Δ (range 15–120 minutes), on the performance of both algorithms was investigated by finding the MAE per station (i.e., bikes/station), which can be described as the prediction error. Moreover, as the number of generated trees by RF and LSBoost is an important parameter in implementing both algorithms, we investigated its effect by changing the number of generated trees from 20 trees to 180 trees with a 40-tree step.

As shown in Figure 5-1 and Figure 5-2, the prediction errors of RF and LSBoost increase as the prediction horizon Δ increases. The lowest prediction error for both algorithms occurred at a 15-minute prediction horizon. Moreover, the prediction error of RF and LSBoost decreases as the number of trees increases until it reaches a point where increasing the number of trees will not significantly improve the prediction accuracy. Figure 5-1 and Figure 5-2 also show that a model consisting of 140 trees yields a relatively sufficient accuracy.

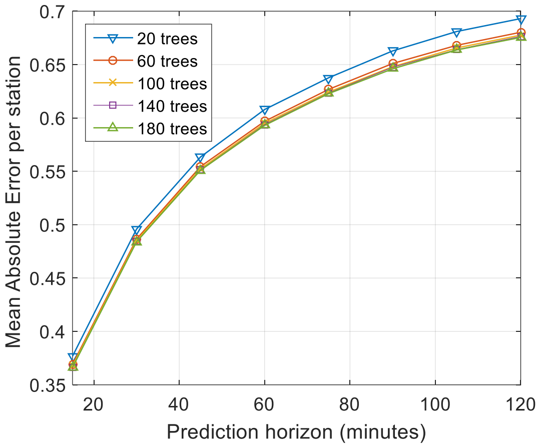


Figure 5-1. RF MAE at different prediction horizons and number of trees.

Comparing the two algorithms, the models produced by RF generally have a smaller prediction error than those produced by LSBoost. LSBoost is a gradient-boosting algorithm, which usually requires various regularization techniques to avoid overfitting (Ganjisaffar, Caruana, & Lopes). As Figure 5-2 clearly shows, as the prediction horizon time increases, the prediction error increases (this is also clearly shown in Figure 5-4 in the next section).

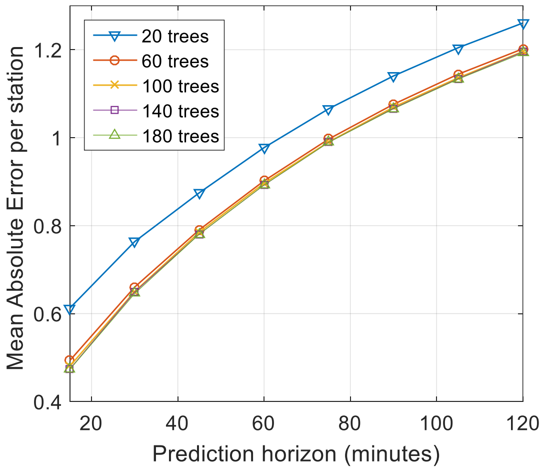


Figure 5-2. LSBoost MAE at different prediction horizons and number of trees.

5.5.2 Multivariate Models

PLSR was used as a multivariate regression to reduce the number of required prediction models for bike stations in the BSS network. When a BSS network has a relatively large number of stations, tracking all the specified models for each bike station becomes complex and time-consuming. For that reason, we examined the adjacency matrix of the Bay Area BSS network and found that the network can be divided into five regions as shown in Figure 5-3. In fact, the bike stations that resulted from the adjacency matrix in each region were found to share the same zip code. This means that the majority of bike trips occurred within the same region and very few trips went from one region to another.

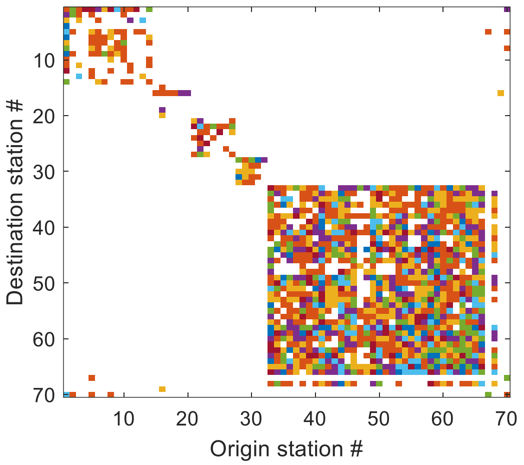


Figure 5-3. Adjacency matrix of the Bay Area Bike Share network.

Using PLSR as a regression algorithm can build prediction models for multivariate response. Therefore, PLSR was applied to reduce the number of models to five, each of which is specified for one region (i.e., one zip code) to reflect the spatial correlation between stations. The input predictors' vector is X_t^i, which consists of the available bikes at the station i at time t, the available bikes at its neighbors at the same time t, the month of the year, day of the week, time of day, and various selected weather conditions. The response's vector is $\log(Y_{t+\Delta}^i)$, where $i = 1, 2, 3, 4, 5$, which is the log of the number of available bikes at all stations in each of the studied regions at a prediction horizon time Δ (range 15–120 minutes). We found that the prediction errors for PLSR were higher than the RF and LSBoost prediction errors when $\Delta =$ 15 minutes, as shown in Figure 5-4. Although the prediction errors resulting from PLSR were higher than the previous results, the resulting models from PLSR are sufficient and desirable for relatively large BSS networks.

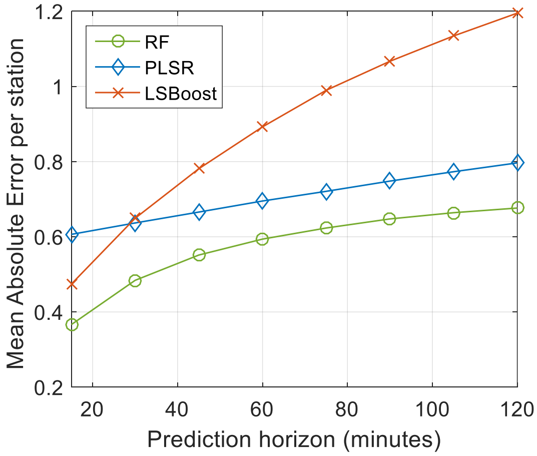


Figure 5-4. PLSR, RF, and LSBoost MAE at different prediction horizons.

5.6 Conclusions and Recommendations for Future Work

In this chapter, we modeled the number of available bikes at San Francisco Bay Area Bike Share stations using machine learning algorithms. The investigation applied two approaches: using univariate regression algorithms, RF and LSBoost, and using a multivariate regression algorithm, PLSR. The univariate models were used to model the available bikes at each station. RF with an MAE of 0.37 bikes/station outperformed LSBoost with an MAE of 0.58 bikes/station. On the other hand, the multivariate model, PLSR, was applied to model available bikes at the spatially correlated stations of each region obtained from the trips adjacency matrix. Results clearly show that the univariate models produced lower error predictions compared to the multivariate model, in which the MAE was approximately 0.6 bikes/station. However, the multivariate model's results might be acceptable and reasonable when modeling the number of available bikes in BSS networks with a relatively large number of stations.

Investigating BSS networks in terms of determined regions gives new insights to policy makers. The fact that stations in each region derived by the multivariate analysis share the same zip code implies that most of the trips were short distance, which may be influenced by the overtime fees applied when trips are longer than 30 minutes. The results also illustrate that station neighbors, prediction horizon time, and weather variables (e.g., temperature and humidity) were found to be significant in modeling the number of available bikes. Specifically, when the prediction horizon time increases, the prediction error increases, with the most effective prediction horizon being 15 minutes. Determining prediction horizon is beneficial to policy makers and technicians to learn how to manage the BSS more responsively, and achieve better performance in prediction. Future work could model the number of available bikes by adding memory as a predictor to handle information related to the number of available bikes in the past.

Chapter 6. Dynamic Linear Models to Predict Bike Availability in a Bike Sharing System

6.1 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 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 concept of BSSs started over five decades ago in Europe, and has since bloomed in 50 countries, growing to more than 37,000 stations by 2000 (DeMaio, 2009). 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, thus making it possible for residents and visitors of the city to access restricted traffic zones with a priority for pedestrians and cyclists over cars.

The significant increase in the use of BSSs raises the issue of imbalance in the distribution of bikes, where 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 (Alvarez-Valdes et al., 2016; Espegren, Kristianslund, Andersson, & Fagerholt, 2016; Schuijbroek et al., 2013). There are three major ways to address the rebalancing issue: static, dynamic and incentivized. The incentivized approach makes includes users in the balancing efforts, as they are offered incentives by the operating company 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. 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 6-1).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 help keep the BSS balanced over time without a need for relocating 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.

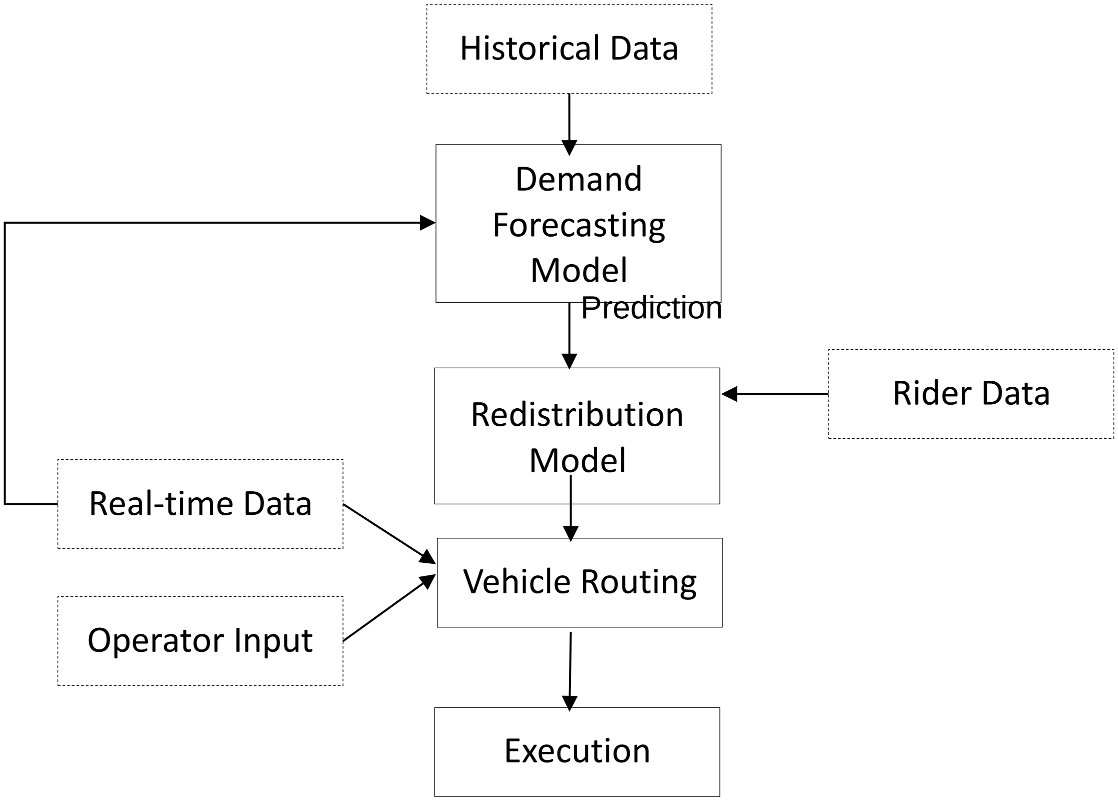


Figure 6-1. Model interactions (Regue & Recker, 2014).

Many researchers use statistical models to predict the demand at any given station, while others use clustering algorithms, such as traditional and non-traditional clustering (M. Almannaa, Elhenawy, & Rakha, 2019; Côme Etienne & Oukhellou Latifa, 2014). 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 (D. W. Daddio & and Mcdonald, 2012; Come Etienne & Oukhellou Latifa, 2014b; X. Wang et al., 2015).

In (Huthaifa I Ashqar, Elhenawy, Ghanem, Almannaa, & Rakha, 2018), the authors developed a bike count model to quantify the effect of weather conditions on the prediction of bike counts at stations using Poisson and NB regression models. 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 (Ashqar H. et al., 2017), the authors used machine learning algorithms—RF, LSBoost, and 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 (B. Wang & Kim, 2018). Chengcheng et al. adapted the long short-term memory neural networks (LSTM NN) for predicating bike prediction and attraction at traffic analysis zones (C. Xu, Ji, Liu, & Peng, 2018). 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 (Pfrommer, Warrington, Schildbach, & Morari, 2014; Singla et al., 2015). The second drawback of the existing machine learning approaches mentioned above is they are sophisticated models using too many variables—there are 19 variables in (Huthaifa I Ashqar, Mohammed Elhenawy, Ahmed Ghanem, et al., 2018) 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 (Jerez et al., 2010). 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 (Harrison & West, 1999). Unlike other statistical and machine learning algorithms 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.

6.2 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 (2014). Poisson, 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 (X. Wang et al., 2015). The results showed that all 13 variables were significant with a high goodness of fit for both models. Rixey used a multivariate linear regression analysis to find the significant factors for the bike sharing ridership, and then estimate system ridership (Rixey, 2013). 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 (Bar-Hillel et al., 2003; Demiriz, Bennett, & Embrechts, 1999; Froehlich et al., 2009b; Sinkkonen et al., 2002). 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 bicycling 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 (Kaltenbrunner et al., 2010). Temporal and geographic mobility patterns were obtained and analyzed with the goal of detecting imbalances in the BSS. Subsequently, the authors 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 (Vogel et al., 2011a). Cluster analysis was used to group the bike stations with respect to pick-up and return activity. The authors 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 (Feng, Hillston, & Reijsbergen, 2017). Gast and Massonnet et al. used a queuing theoretical time-homogeneous model of BSSs to make probabilistic forecast (Gast, Massonnet, Reijsbergen, &

Tribastone, 2015). They also introduced a new metric to evaluate the proposed model instead of the standard root-mean-square error. 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 (Fricker & Gast, 2016).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 (Froehlich et al., 2009b; Gallop et al., 2011; Kaltenbrunner et al., 2010; Yoon et al., 2012). Although these techniques showed good performance in both explaining the past and predicting the future, they had several limitations. For example, Kaltenbrunner et al. (Kaltenbrunner et al., 2010) used an autoregressive moving average (ARMA) model to predict the bike availability at stations (Kaltenbrunner et al., 2010). 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 (Froehlich et al., 2009b). 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 (Yoon et al., 2012). Seasonal trends and neighboring information were utilized in the model. A small dataset of 3 weeks was used to evaluate the model. The 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.

6.3 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 (Petris, Petrone, & Campagnoli, 2009). 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: $Y_t = F_t\theta_t + \nu_t,$ (6-1)

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

the evolution equation: $\theta_t = G_t\theta_{t-1} + \omega_t,$ (6-2)

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.

6.3.1 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 (Petris et al., 2009). 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:

$Y_t = \mu_t + v_t \qquad v_t \sim N(0, \sigma_v)$ (6-3)

$\mu_t = \mu_{t-1} + w_t \qquad w_t{\sim} N(0, \sigma_w)$ (6-4)

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.

6.3.2 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: 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:

$$Y_t = F_t \mu_t + v_t \qquad v_t \sim \mathrm{N}(0, \sigma_{v_t}) \tag{6-5}$$

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

$$B_t = B_{t-1} + w_{2,t} \qquad w_{2,t} \sim \mathrm{N}(0, \sigma_{w_{2,t}}) \tag{6-7}$$

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:

$$Y_t = F_t \, \theta_t + v \qquad v_t \sim \mathrm{N}(0, \sigma_{v_t}) \tag{6-8}$$

$$M_t = G_t \, \theta_{t-1} + w \qquad w \sim \mathrm{N}\left(0, \begin{pmatrix} \sigma_{w_1} & 0 \\ 0 & \sigma_{w_2} \end{pmatrix} \right) \tag{6-9}$$

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 $w = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$.

6.4 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 (see Figure 3-1). The dataset included station ID, number of bikes available, number of docks available, and time of recording. Each row had 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.

6.5 Model testing

The first- and second-order polynomial models were coded using R (Petris et al., 2009). 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) were 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.

6.6 Evaluation Criteria

To measure the predictive accuracy of the two models, two different measurements were used: the 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 (6-8). The SMAPE is an accuracy measure and is calculated as shown in (6-9).

$$\text{MAE} = \frac{\sum_{i=1}^{n} |Y_t - A_t|}{n} \qquad (6\text{-}8)$$

$$\text{SMAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \frac{|Y_t - A_t|}{(|A_t| + |Y_t|)/2} \qquad (6\text{-}9)$$

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 was the MAE relevant to the capacity of the station (MAE/C) in which we divided the MAE for each station over its capacity. This was to make the prediction error more informative by considering the capacity of stations.

6.7 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.

6.8 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:

$$\mu_t = E(Y_{t+k} \ / \ Y_{1:t} \ ) \tag{6-10}$$

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:

$$E\left(\frac{Y_{t+k}}{Y_{1:t}}\right) = \mu_t + K \times B_t \tag{6-11}$$

6.9 Results

Table 6-1 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 6-1. 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 6-2, 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, producing a prediction error of 6% and 9.3% bikes/station for the single- and multiple-step approaches, respectively.

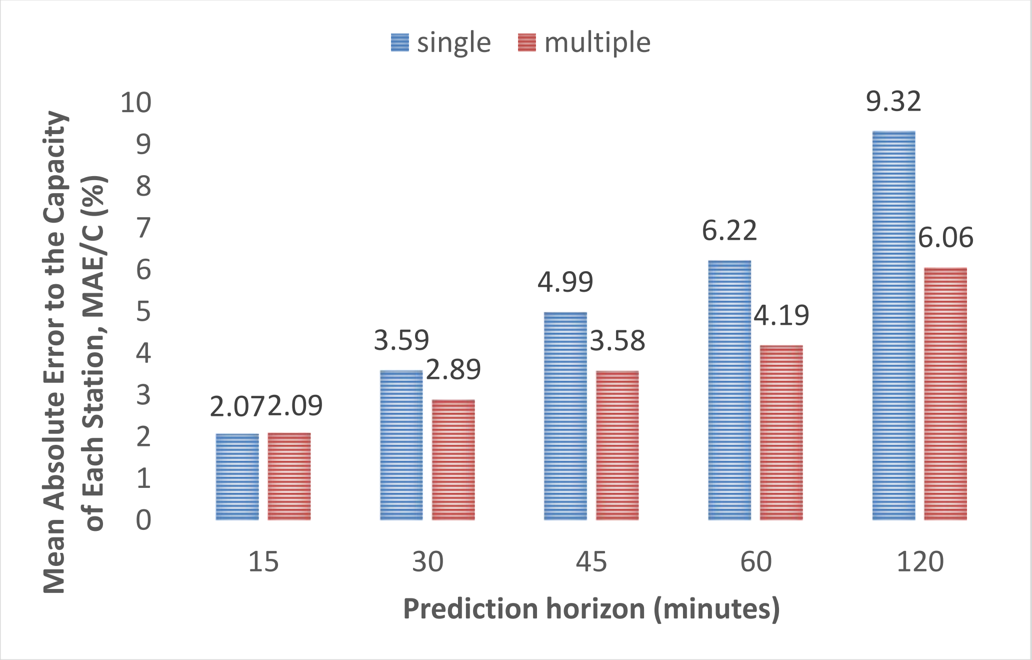


Figure 6-2. Prediction error with respect to the capacity for the single- and multiple-step approaches for the first-order DLM for different prediction horizons.

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 6-3). 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.

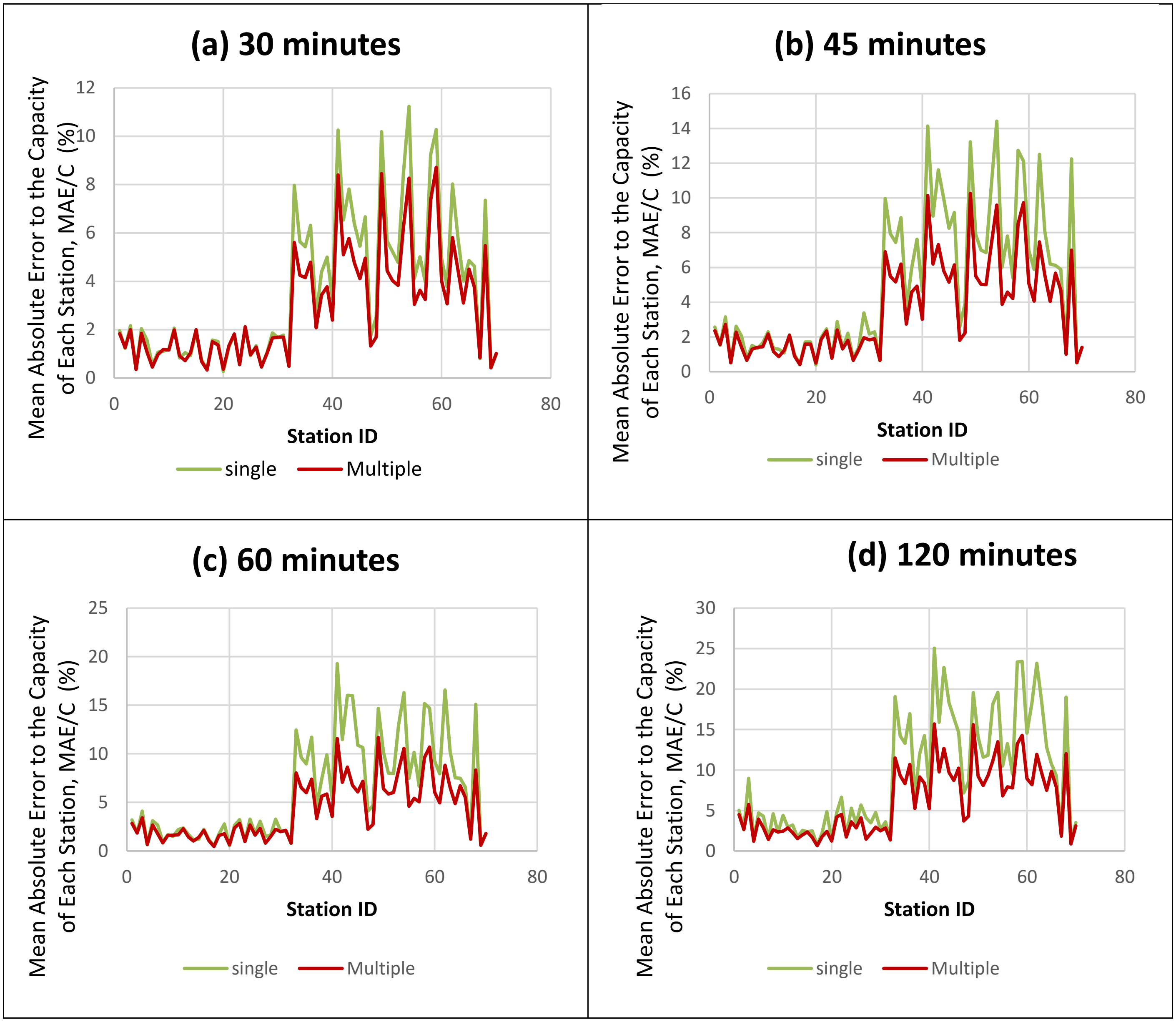
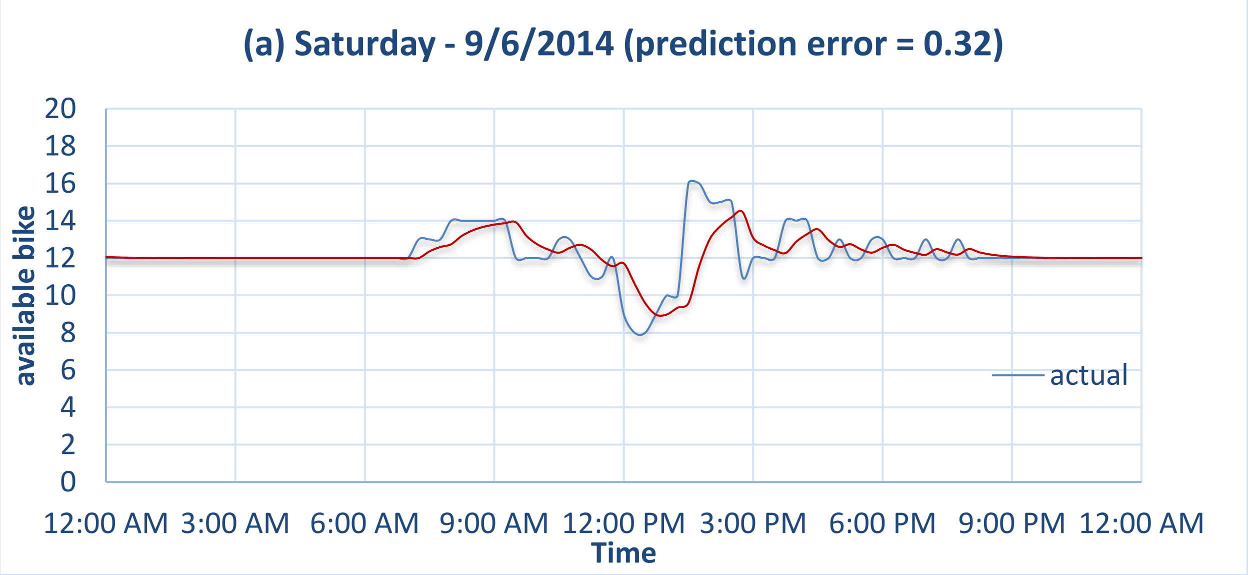


Figure 6-3. MAE/C per station for single-step and multiple-step forecasts at different prediction windows.

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 6-4**,** 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 6-4(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 6-4(b), the station had more bike activity than Figure 6-4(a), but generally the expected pattern follows the actual curve with a small delay in responding to the jumps. In Figure 6-4(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 of 1.86 bikes/station.



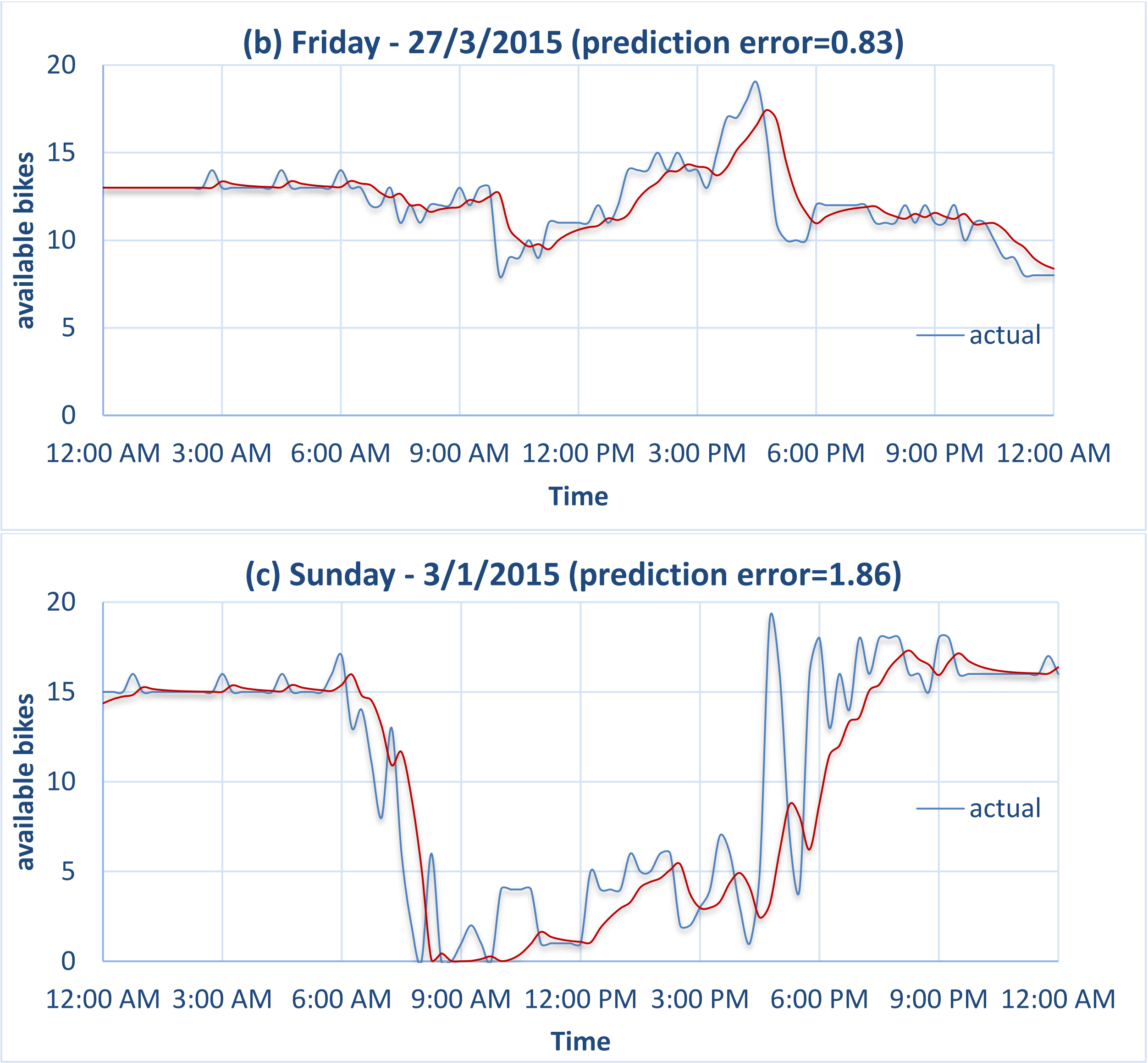


Figure 6-4. Pattern of expected and actual bike availability for three different days of the week at 15-minute prediction window of one station for first-order DLM, multiple-step technique.

6.10 Comparison with Other Machine Learning Algorithms

In (Ashqar H. et al., 2017), two machine learning algorithms were adopted: 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 (Gallop et al., 2011; Rudloff & Lackner, 2013b), and also were found to be significant. We compared the best results 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 6-5). 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.

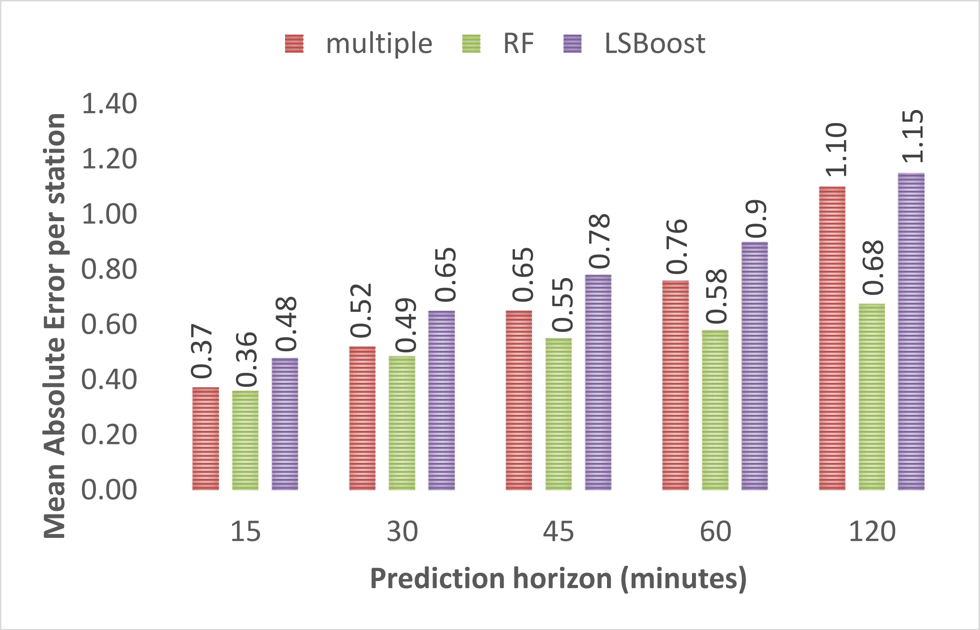


Figure 6-5. Multiple-step approach of the first-order DLM, RF, and LSBoost MAE at different prediction windows.

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 well beyond the results presented here.

6.11 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 in solving 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 from saturated to non-saturated stations using service trucks. 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 of 15, 30, 45, 60, and 120 minutes were used to investigate the effect of the length of the prediction horizon on 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 windows (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 outperformed the RF and LSBoost algorithms for 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

Chapter 7. Incremental Learning Models of Bike Counts at Bike Sharing Systems

7.1 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 (2016).

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 (Ghosh, Varakantham, Adulyasak, & Jaillet, 2017). 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 (Saltzman & Bradford, 2016). 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, the BSS in the San Francisco Bay Area started operating in 2013 with 700 bikes and 70 stations, and now the current operator (Ford, operating the system as GoBike) 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.

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 (Brinkmann et al., 2016; Caggiani & Ottomanelli, 2012; Espegren et al., 2016; Kloimüllner, Papazek, Hu, & Raidl, 2014). The incentivized approach aims to encourage bikers to change either their origin or destination in favor of balancing the system (Fricker & Gast, 2016).

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 (Rixey, 2013; X. Wang et al., 2015), clustering, and exploring algorithms (Froehlich et al., 2009b; Kaltenbrunner et al., 2010), machine learning algorithms (Huthaifa I Ashqar et al., 2017; H. Yang, Wang, Xie, Ozbay, & Ma, 2018), and time series techniques (Kaltenbrunner et al., 2010; Yoon et al., 2012). 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 with up to 137 features (Dressel & Farid, 2018). 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 dataset 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.

7.2 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, NB, and hurdle models to predict bike demand using temperature, precipitation, and neighboring stations as predictors (Rudloff & Lackner, 2014). 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 (X. Wang et al., 2015). 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 (Rixey, 2013). The authors 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 RF found that time-of-day, temperature, and humidity level were significant predictors in bike prediction (Huthaifa I Ashqar, Elhenawy, Almannaa, Ghanem, & Rakha, 2018). The authors adopted two count models: Poisson and NB along with RF; their results showed that RF outperformed the other two models.

Due to the size of BSS datasets, 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 (Froehlich et al., 2009b). 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 (Vogel et al., 2011a). 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 dataset and account for many predictors to discover hidden dataset patterns. (Huthaifa I Ashqar et al., 2017; H. Yang et al., 2018). 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 (Huthaifa I Ashqar et al., 2017). 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 (H. Yang et al., 2018). 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 (Saridis & Stein, 1968). 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.

7.3 Methods

7.3.1 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 (West, Harrison, & Migon, 1985). 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:

$$\widehat{y_t} = x_t^T \beta_{t-1} \tag{7-1}$$

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:

$$\beta_t = \beta_{t-1} + \ 2\,\alpha(y_t - x_t^T \beta_{t-1})x_t \tag{7-2}$$

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 (7-2), 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 (7-2) and (7-3).

$$\beta_t = \beta_{t-W} + \ 2\ \sum_{j=0}^{W} \alpha\big(y_{t-j} - x_{t-j}^T \beta_{t-W}\big)x_{t-j} \tag{7-3}$$

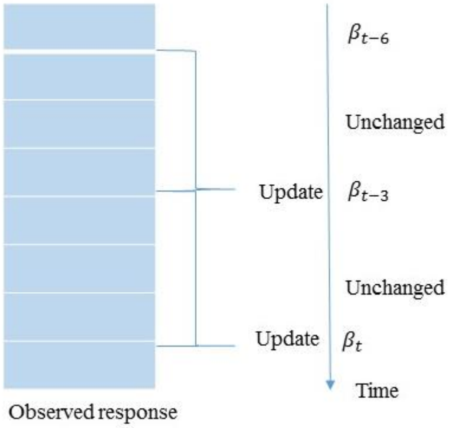


Figure 7-1. Illustration of the regression coefficients updating process (W = 3).

As shown in Figure 7-1, 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.

7.4 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. 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 (7-4).

$$K = diag(e^{-\frac{d_i^2}{2\,\sigma^2}}) \qquad (7\text{-}4)$$

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 (7-5) and (7-6).

$$\beta_{t+1} = inv(X' * W_i * X) * X' * W_i * Y \tag{7-5}$$

$$\hat{y}_{t+1} = x_{t+1}^T \beta_{t+1} \tag{7-6}$$

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.

7.5 Dataset: Case Study of San Francisco

This study used a publicly available BSS docking station dataset. Details of the San Francisco Bay Area Bike Share dataset can be found in Section 3.4 of this report.

Due to the large size of the dataset, we derived a subset of the original dataset 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 dataset. When analyzing the dataset, 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 (now Ford, which operates the BSS as GoBike).

7.6 Results and Discussion

7.6.1 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 started 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 showed a 7-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 showed that W_L starts returning 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.

7.6.2 Evaluation Criteria

To measure the predictive accuracy of the two models, two different measurements were used: The MAE and the 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 (7-7). The SMAPE is an accuracy measure and is calculated as shown in (7-8).

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

$$\text{SMAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \frac{|Y_t - A_t|}{(|A_t| + |Y_t|)/2} \qquad (7\text{-}8)$$

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

7.6.3 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 7-1, 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 7-2, 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 7-1. 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

Prediction Horizons (Minutes)	MBGDLR		LWR	
	MAE	SMAPE	MAE	SMAPE
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).

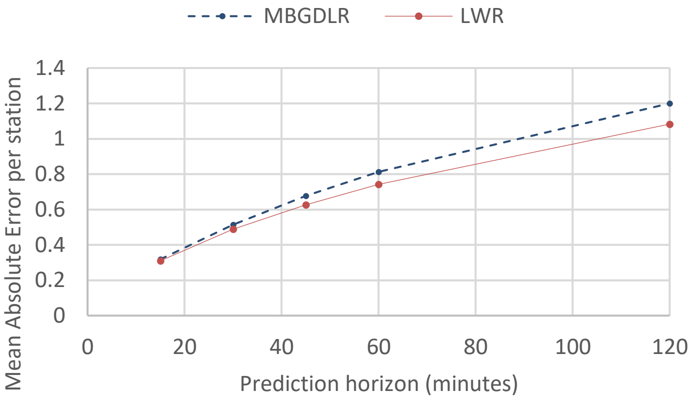


Figure 7-2. Prediction error for MBGDLR and LWR at different prediction horizons.

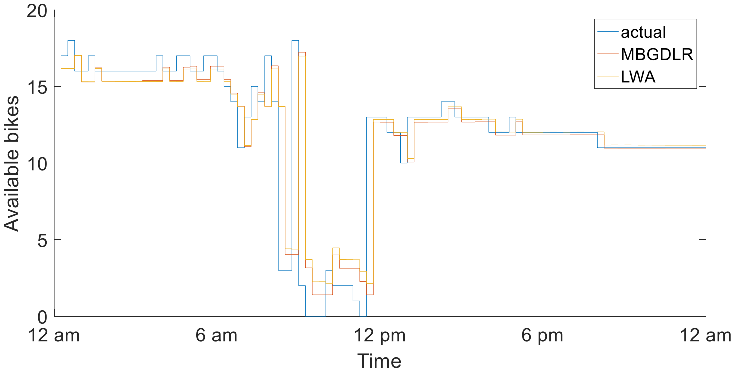


Figure 7-3. One-day pattern of expected and actual bike availability at 15-minute prediction window for MBGDLR and LWR algorithms, station 59.

To investigate why LWR performed slightly better than MBGDLR, though the expected pattern does not differ much (Figure 7-3), we looked at station-level prediction error for all prediction horizons at all stations and compared both algorithms. Results are shown in Figure 7-4 (note: only the 15-minute prediction horizon is presented here as an example). As Figure 7-4 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 7-4, 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 7-5. 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).

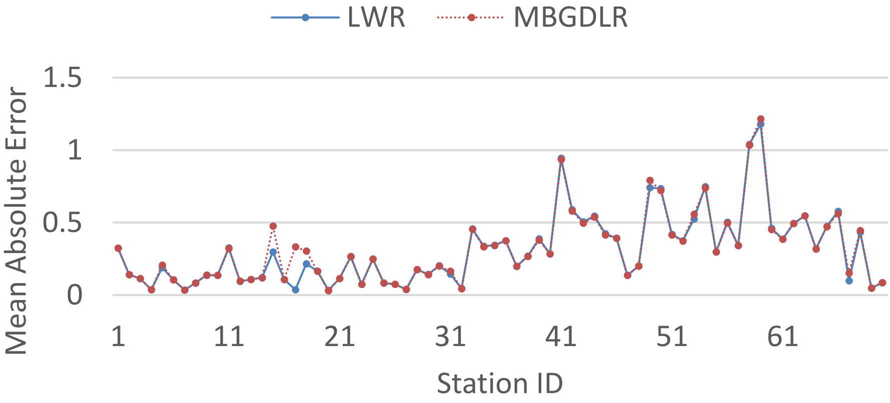


Figure 7-4. MAE per station for MBGDLR and LWR algorithms across stations at a 15-min prediction window.

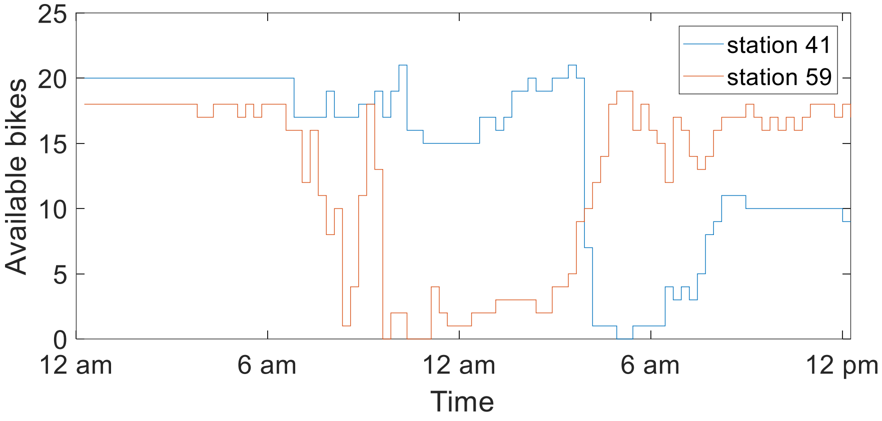


Figure 7-5. Pattern of bike availability for stations 41 and 59.

7.6.4 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 DLM (M. H. Almannaa, Elhenawy, & Rakha, 2018), 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 (Huthaifa I Ashqar, Mohammed Elhenawy, Mohammed H Almannaa, et al., 2018). 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 7-6, 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 MBGDLR).

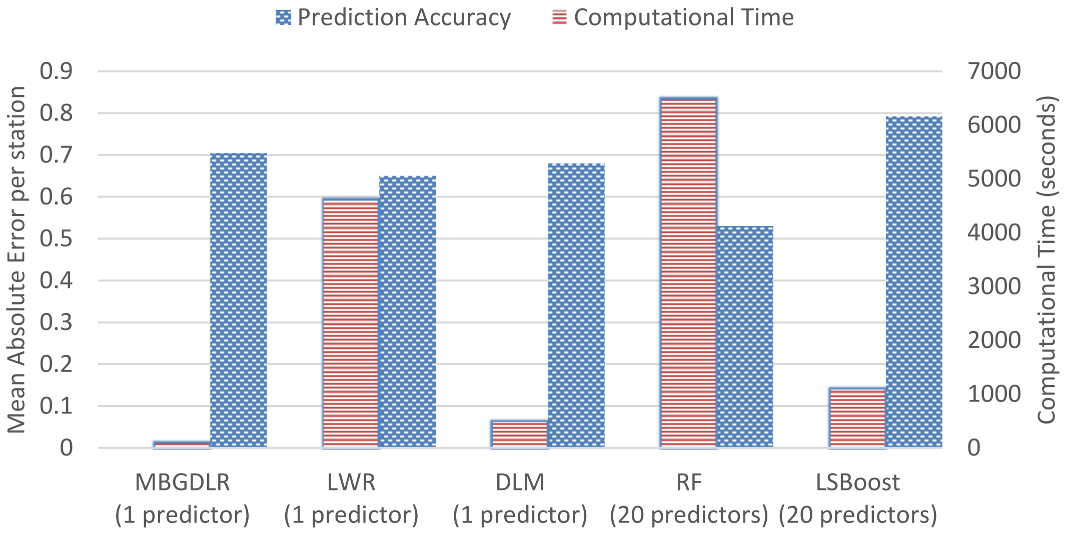


Figure 7-6. Comparison of the average computational time and MAE of all prediction windows for MBGDLR , LWR, DLM, RF, and LSBoost algorithms for all 70 stations.

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

7.7 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 of 15, 30, 45, 60, and 120 minutes were used. 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 8. Predicting Station Locations in Bike-Sharing Systems Using a Proposed Quality-of-Service Measurement: Methodology and Case Study

8.1 Introduction

A growing population, with more people living in cities, has led to increased pollution, noise, congestion, and greenhouse gas emissions. One possible approach to mitigating these problems is encouraging the use of BSSs. BSSs are an integral part of urban mobility in many cities and are sustainable and environmentally friendly. As urban density increases, it is likely that more BSSs will appear due to their relatively low capital and operational costs, ease of installation, pedal assistance for people who are physically unable to pedal for long distances or on difficult terrain, and the ability to track bikes (DeMaio, 2009).

BSS operators take great efforts to ensure bike and dock availability at each station. This task can be difficult as the movement of users are highly dynamic, difficult to predict, and redistributing bikes is expensive. Recent studies have shown that there are spatial dependencies in bike usage at different stations (Borgnat, Fleury, Robardet, & Scherrer, 2009; Froehlich, Neumann, & Oliver, 2009a; Kaltenbrunner et al., 2010; Vogel et al., 2011b), and that imbalances in the spatial distribution of bikes occur due to one-way use and short rental periods (Vogel et al., 2011b). Thus, it is necessary for operators to understand the spatial dependencies to more effectively manage the system. For example, operators could improve the QoS by identifying the best candidate spots for new stations. However, finding the best QoS measurement for a station in a heterogeneous BSS and using it to study the spatial dependencies in the system is a challenging problem.

We investigated the state-of-art QoS measurement and found it to be largely indiscriminative at the station level. In this study, we propose a new QoS measurement, Optimal Occupancy, to discriminate between different stations in heterogeneous BSSs. We demonstrate that Optimal Occupancy is not only discriminative but can also capture the spatial correlations in a BSS.

8.2 Related Work

Modeling bike sharing data is an area of significant research interest. In general, the main goals of previous studies have been to boost the redistribution operation (Caggiani, Camporeale, Ottomanelli, & Szeto, 2018; Contardo et al., 2012; Liu, Szeto, & Ho, 2018; Pal & Zhang, 2017; Raviv et al., 2013; Schuijbroek et al., 2013), to gain new insights into and correlations between bike demand and other factors (Bordagaray, dell'Olio, Fonzone, & Ibeas, 2016; David William Daddio, 2012; Rixey, 2013; Rudloff & Lackner, 2013b; X. Wang et al., 2015), and to support policy makers and managers in making optimized decisions (David William Daddio, 2012; Vogel et al., 2011b).

Research questions that have been studied previously include the strategic design, operation, and analysis of BSSs. Due to the potential benefits to operators, measuring the level of stations' or the entire system's service (Gunasekaran, Patel, & Tirtiroglu, 2001) has become an appealing issue for researchers. In some cases, operators measure the fraction of time that their stations are full or empty as a measurement of the system’s QoS (Schuijbroek et al., 2013). Similarly, Fricker et al. considered the limiting probability that a station is empty or full as the performance measure. They argued that the optimal proportion of bikes at a station is slightly more than half the capacity of a station in a homogeneous system. In an heterogeneous system, however, they concluded that this performance metric collapses due to the heterogeneity (Fricker, Gast, & Mohamed, 2012).

Lin and Yang (Lin & Yang, 2011) investigated the strategic problems by studying the question of bike stations’ measures of service. They argued that the measures of QoS in the system should include two measurements: the availability rate, which was defined as the proportion of pick-up requests at a bike station that are met by the bicycle stock on hand, and the coverage level, which is the fraction of the total demand at both origins and destinations that is within some specified time or distance from the nearest rental station. Fricker and Gast (Fricker & Gast, 2016) proposed a stochastic model of a homogeneous BSS and investigated the impact of users’ random choices on the number of problematic stations. Problematic stations were defined as stations that, at a given time, have no bikes available or no available spots for bikes to be returned to. Consequently, the performance of the system was determined by the proportion of problematic stations. However, these measures have critical drawbacks: (1) as BSSs usually offer two services—picking up bikes, and returning bikes—these measurements fail to take into account the QoS of returning bikes to stations; (2) some of the studies assume that, in contrast to real systems, the system is homogeneous; and (3) while some studies modeled the system as heterogeneous, they failed to consider the variability of the system parameters (i.e., arrival and pickup rates) throughout the same day or across the different days of the week and their dependency on the individual station.

In any BSS, one of the keys to success is the location and distribution of bike stations (Lin & Yang, 2011). Some studies have worked on locating bike stations using different methods, such as location-allocation models (García-Palomares, Gutiérrez, & Latorre, 2012), and an optimization method that maximizes the demand covered and takes the available budget as a constraint (Frade & Ribeiro, 2015). The spatial distribution of the potential demand is a fundamental element in optimal location modeling. In order to estimate the potential demand, several studies used preference surveys to evaluate both the factors influencing the use of the bicycle mode and choice of routing (Abraham, McMillan, Brownlee, & Hunt, 2002; Dill & Voros, 2007; Meng, 2011; Shafizadeh & Niemeier, 1997). Potential demand has also been estimated by considering the population, employment associated with each building, and the number of trips generated for each transport zone (García-Palomares et al., 2012). However, there are some limitations and drawbacks in the methods previously used to find the optimal station location: these methods are basically used to plan new systems and might not be useful to predict new stations in existing systems; they are aimed at serving the local population on selected days (e.g., workdays); and certain places in the studied area (e.g., large parks) have neither population nor jobs and yet may attract a considerable number of trips.

This chapter makes two major contributions to the literature: (1) we propose a new discriminative QoS measure that reflects the spatial dependencies in a heterogeneous BSS and that considers the variability of arrival and pickup rates; and (2) we use this QoS measure with geo-statistics to model a spatial variogram that could predict the QoS in nearby areas for the purpose of locating new stations in an existing BSS.

8.3 Proposed QoS Measurement

BSSs are highly heterogeneous. The arrival rates, pickup rates, origins, and destinations between stations in diverse areas and topographies are very different. These parameters may also vary with the time of day, day of the week, and season (Borgnat et al., 2011). In this study, we consider that the bike-sharing system has N stations, in which each station i may have a unique capacity C_i (i.e., maximum number of docks). We assume that the dynamics of the system are as follows. Users reach the stations to pick up a bike at varying departure rates $\dot{D}_\iota$ at station i (we name this departure rate as the user's intention is to take the bike and depart to their destination stations). This departure rate $\dot{D}_\iota$ depends on station i and varies throughout the day and with different days of the week. If there are no available bikes, the user leaves the system or waits until another user arrives to return a bike. Users arrive at their destination stations to return the bike at a varying rate $\dot{A}_J$ at station j. Similar to the departure rate, the arrival rate $\dot{A}_J$ depends on station j and varies throughout the day and with different days of the week. If there are less than C_j (i.e., capacity) bikes in this station, the user returns the bike and leaves the system. If the station is full, the user either chooses another station to return the bike or waits until another user reaches the station to pick up a bike.

To consider the impact of system heterogeneity, we introduce a new QoS measurement for each station: Optimal Occupancy. The Optimal Occupancy of a station is formulated in terms of two services: (1) picking up bikes, and (2) returning bikes. As each station i has a finite number of docks (i.e., capacity), two thresholds should be defined. The lower threshold (L_i) is the point when the number of available bikes ($B_{i,t}$) in station i at time t drops low enough that the possibility of a user not finding a bike is very high. The upper threshold (U_i) is the point when the number of bikes ($B_{i,t}$) in a station i at time t is high enough that the possibility of a user not finding a dock to return a bike is very high. For example, if a station's capacity is 25 docks and the number of available bikes at time t is within $[5, 20]$, then the station is considered functional, and otherwise it needs to be rebalanced (i.e., it is a problematic station). In that sense, the Optimal Occupancy (O_{op}) is formulated as the ratio of the total time that a station is functional (t_f) during a given interval to the length of the interval (t_{total}):

$$O_{op_i} = \frac{t_{i,f}}{t_{i,total}} \tag{8-5}$$

where $t_{i,f} = \sum_{t=0}^{t=t_f} X_i(t)$ where $X_i(t)$ is the status function and defined as

$$X_i(t) = \begin{cases} 1, & \text{station } i \text{ is functional} \\ 0, & \text{station } i \text{ is problematic} \end{cases} \tag{8-6}$$

and station i is functional if $B_{i,t} \in [L_i, U_i]$ at any given time t. (8-7)

As the two thresholds L_i and U_i define the functionality of the station, L_i and U_i are correlated with the departure rate $\dot{D}_\iota$ and arrival rate $\dot{A}_\iota$, respectively. Both $\dot{D}_\iota$ and $\dot{A}_\iota$ randomly vary throughout the day, with different days of the week, and different months of the year. However, in this study, we assume that $\dot{D}_\iota$ and $\dot{A}_\iota$ vary only with different days of the week (DoW), and different months of the year (M) to be consistent with the length of the study interval (see Analysis and Results section of this chapter). In fact, $\dot{D}_\iota$ and $\dot{A}_\iota$ are the bike counts picked up (D_i) or returned (A_i), respectively, per unit time ($t_{i,total}$). In that sense and to reflect the stochastic phenomenon in the system, $\dot{D}_\iota$ and $\dot{A}_\iota$ were modeled using a PRM with an exposure variable. Exposure is a measure of how the bike counts are divided. Since both rates are bike counts per unit time, time is considered as the exposure. The model contains a $log(t_{i,total})$, called the offset variable, as a term that could be added to the regression coefficients:

$$D_i \text{ or } A_i \sim Poisson\left(\theta_i^{(D) \text{ or } (A)}\right) \tag{8-8}$$

$$\text{where } \theta_i^{(D)} = \frac{\mu_i}{t_{i,total}}, \text{ and } \theta_i^{(A)} = \frac{\lambda_i}{t_{i,total}} \tag{8-9}$$

$$\log\left(\frac{\mu_i \text{ or } \lambda_i}{t_{i,total}}\right) = \beta_0 + \beta_1 DoW + \beta_2 M \tag{8-10}$$

$$L_i = \mu_i \tag{8-11}$$

$$U_i = C_i - \lambda_i \tag{8-12}$$

In that sense, problematic stations can be redefined as stations that, at any given time t, have fewer bikes available than the expected bike counts to be picked up during analysis discretization duration or more bikes than the difference between capacity and the expected bike counts to be returned during analysis discretization duration. The next sections in this study will further explain the concept of the proposed Optimal Occupancy QoS measurement by applying it to a real BSS dataset and comparing the new definition of problematic stations with the one previously used.

8.4 Dataset

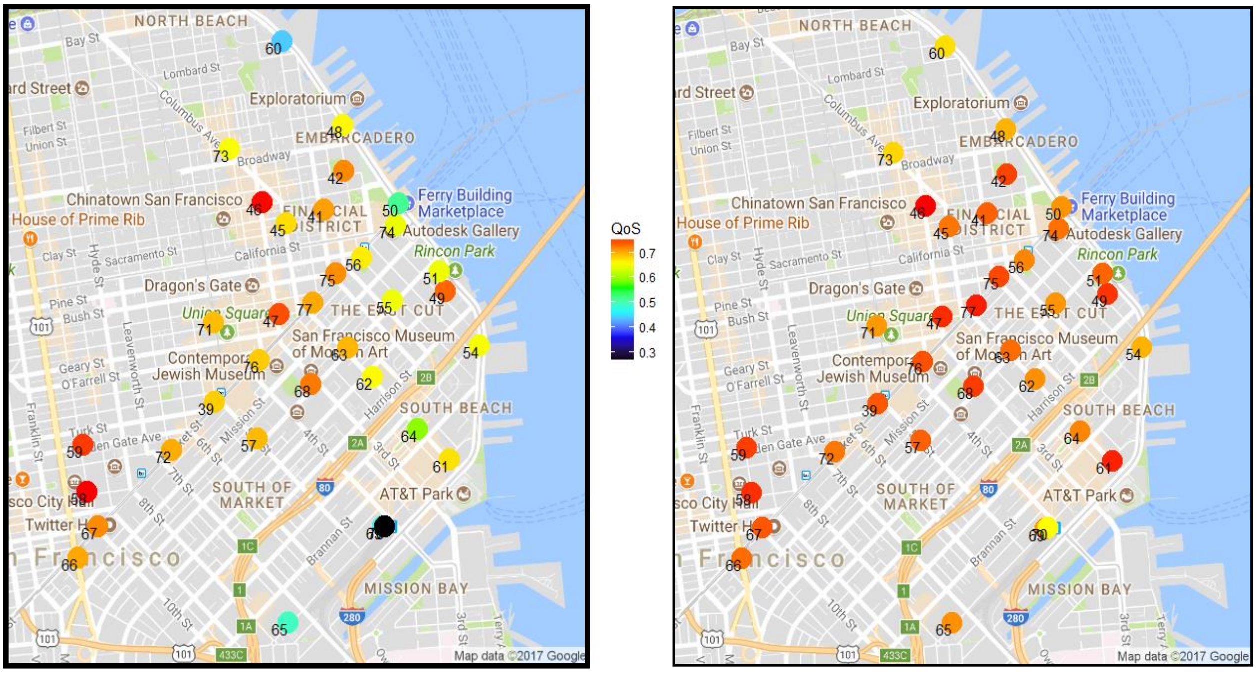
One of the first BSSs in the U.S. was established in 1964 in Portland, with 60 bicycles available for public use. Although BSSs are still relatively limited, at present many cities, such as San Francisco and New York, have launched BSS programs. These programs implement different payment structures, conditions, and logistical strategies. In 2013, San Francisco launched the Bay Area Bike Share System (now operated by Ford under the name GoBike), a membership-based system providing 24-hours-per-day, 7-days-per-week self-service access to short-term rental bicycles. A detailed description of this system is provided in section 3.4 of this report.

This study used anonymized bike trip data collected from August 2013 to August 2015 in San Francisco (Hamner, 2016). This study used two datasets of 34 stations in downtown San Francisco (**Error! Reference source not found.**). The 34 stations have different capacities, ranging from 15 to 27 docks, which means the system is heterogeneous. The first dataset includes station ID, number of available bikes, number of available docks, and time of recording. The time data include the year, month, day of month, day of week, time of day, and minute at which a record was documented. As the database was updated every minute for 34 stations in San Francisco over 2 years, this dataset contains a large number of recorded incidents. The second dataset consists of the station ID, name of station, latitude and longitude of each station, the maximum number of docks, and the installation date. The latitude and longitude of each station were converted to the Universal Transverse Mercator Coordinate (UTM) system, which is expressed as a two-dimensional projection on the surface of the Earth (National Geodetic Survey, 2017).

8.5 Analysis and Results

In a BSS, the QoS measurement should reflect the spatial dependencies of BSS stations in addition to describing the performance of a station's service. Consequently, we investigated the traditionally-known QoS measurement using the Bay Area BSS dataset in San Francisco. We found that it was neither satisfying in exposing the spatial dependencies between stations nor adequate in describing the performance of the service.

The first QoS measurement presented in different studies, such as in (Fricker & Gast, 2016; Fricker et al., 2012; Schuijbroek et al., 2013), is that of problematic stations, defined as stations that, at a given time, have no bikes available or no available spots for bikes to be returned to. This definition has been mainly used to describe the overall performance of the system. However, we used that definition to find a QoS measurement for a specific station by computing the ratio of the total time that a station is not problematic during a given interval to the length of the interval. The second measurement is our proposed QoS measurement, Optimal Occupancy (O_{op}), which redefines problematic stations as stations that, at any given time t, have fewer bikes available than the expected bike counts to be picked up during analysis discretization duration or more bikes than the difference between capacity and the expected bike counts to be returned during analysis discretization duration. Similarly, we used our definition to find the Optimal Occupancy for a specific station by computing the ratio of the total time that a station is not problematic (i.e., functional) during a given interval to the length of the interval. For this specific dataset, and to effectively represent the service in the system, we defined the length of the study interval in both definitions as running from 8 a.m. to 5 p.m., the interval that was found to be the peak hours for the system (M. H. Almannaa, M. Elhenawy, A. Ghanem, H. I. Ashqar, & H. A. Rakha, 2017). Figure 8-1 shows the locations of the stations with the corresponding results of the two QoS average measurements (over 2 years) of 34 stations in the Bay Area Bike Share in San Francisco. The measurements were first found for every 15 minutes at each station then averaged over the interval of the peak hours for the system.



(a) (b)

Figure 8-1. The locations, and the values of the (a) proposed QoS, and (b) traditionally-known QoS measurements.

8.5.1 Analysis of Variance (ANOVA)

Analysis of variance (ANOVA) was used to determine whether there were any statistically significant differences between the means of the two QoS measurements. Before interpreting the results of the hypothesis tests, we checked the ANOVA assumptions, and the hypothesis test results were found to be trustworthy. BSSs are highly heterogeneous, with arrival rates and pickup rates between stations in diverse areas and topographies varying with the time of day, day of the week, and season (M. H. Almannaa et al., 2017; Borgnat et al., 2011). Therefore, to fairly compare the two measurements, we compared the daily values for specific months and days. ANOVA was used to analyze the differences among four group means for all 34 stations: (1) Tuesdays of February, (2) Tuesdays of July, (3) Mondays of February, and (4) Mondays of July. The p-values resulting from testing the groups of traditionally-known QoS measurements were $0.7704, 0.8400, 0.5099,$ and 0.7443, respectively. This means that the null hypothesis is true and there are no significant differences ($p > 0.05$). On the other hand, the p-values resulting from testing the groups of the proposed QoS measurements (O_{op}) were $2.73E-29, 3.25E-36, 7.34E-41,$ and $1.42E-30$, respectively. This means that the null hypothesis is rejected and that there were significant differences between the measurements of the stations ($p < 0.05$). Figure 8-2 shows the differences among the Tuesdays of February group means for all 34 stations, clearly demonstrating that the traditionally-known QoS cannot be used to discriminate between the stations, while the proposed Optimal Occupancy is discriminative to a sufficient extent. In that sense, recognition of the differences between the QoS of stations is not required in and of itself but because it is necessary for operators to effectively manage the system and it appears to reflect the dynamics of the BSS. Although we present the results of only four groups, in fact we examined the ANOVA test for other groups that cover most of the days of the week and months of the year. The results were found to be consistent with the results presented here.

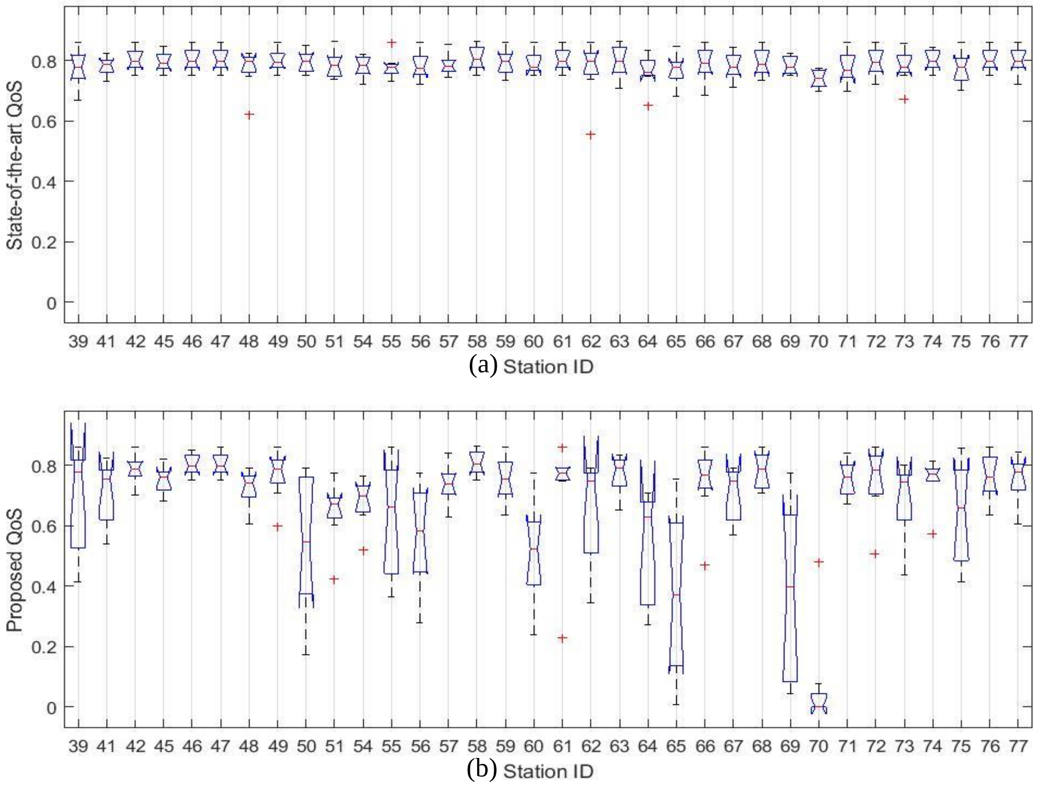


Figure 8-2. ANOVA test for Tuesdays of February for the 34 stations for (a) traditionally-known QoS, and (b) proposed QoS.

8.5.2 Spatial Analysis

We applied geo-statistics to explore the spatial configuration of Optimal Occupancy variations. We used two packages in R: geoR to analyze geostatistical data (Ribeiro Jr & Diggle, 2001) and gstat to perform geostatistical modelling and prediction (Pebesma, 2004). The analysis was performed to assess whether the proposed Optimal Occupancy measurements can reflect the spatial dependencies and be used to predict the QoS in nearby areas. This would allow operators to determine candidate spots for new stations in the BSS, increasing the overall QoS of the system.

Spatial statistics attempt to develop inferential methods to properly account for the spatial dependences in the presence of georeferenced observations. Spatial modeling typically contains a specification of a mean function and a model of the correlation structure (i.e., variogram), which is a description of the spatial continuity of the data. The variogram is the key function in geostatistics, as it is used to fit a model of the spatial correlation of the observed phenomenon (Banerjee, Carlin, & Gelfand, 2014). A variogram model is chosen by plotting the empirical variogram, which is a simple nonparametric estimate of the variogram, and then comparing it to various theoretical shapes available. A variogram could be mathematically defined as (Banerjee et al., 2014):

$$\gamma(\Delta x, \Delta y) = \frac{1}{2}\varepsilon\,[\{Z(x+\Delta x, y+\Delta y\} - Z(x,y)\}^2] \tag{8-13}$$

where $Z(x,y)$ is the value of the variable of interest at location (x,y), and $\varepsilon\,[\,]$ is the statistical expectation operator. The variogram, $\gamma()$, is a function of the separation between points $(\Delta x, \Delta\, y)$, and not a function of the specific location (x,y). However, one common assumption of the spatial analysis is that it is isotropic. An isotropic variogram means that the correlation between any two observations depends only on the distance between those locations and not on their relative direction; otherwise, it is anisotropic (Maity & Sherman, 2012).

A series of directional empirical variograms (including directions between 0° and 180°) was investigated to highlight the main observations' directions and check the spatial isotropy in the proposed QoS measurements data. The results illustrate that we cannot assume isotropy and that the directional empirical variogram for 45° outperforms other variograms, as it reflects the correlation between the observations and the distance. The empirical variogram for 45° using transformed coordinates was estimated and is illustrated in Figure 8-3. It shows a steady increase in the semi-variance over increasing distance intervals to an absolute maximum between 1.0 and 1.5 km. For greater distances, Figure 8-3 displays an oscillatory state with a second maximum around 2.5 and 3 km.

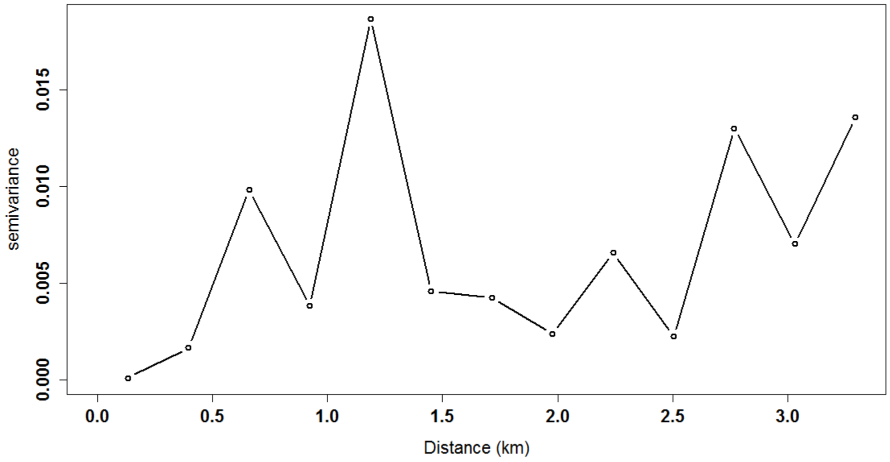


Figure 8-3. The empirical variogram for 45° using transformed coordinates.

Modeling variograms are usually used for spatial prediction (i.e., interpolation). Most practical studies used exponential, spherical, and Gaussian models. As we assumed anisotropy, we applied the maximum likelihood estimation of spatial regression models to estimate the angle for geometric anisotropy of the three models. The exponential variogram model yields the most beneficial realization of the spatial process in the BSS. While the spherical model yields a decent estimation, the Gaussian model fails to fit a variogram that manifests the spatial correlation. We also applied the maximum likelihood estimation for the same three models to fit the traditionally-known QoS measurement to compare it with the Optimal Occupancy. Similarly, the exponential variogram model outperforms the spherical and the Gaussian models. Results in Table 8-1 show some inferences. According to the BIC of the spatial and non-spatial models, the spatial model for Optimal Occupancy outperforms the non-spatial model, but the traditionally-known QoS non-spatial model outperforms the spatial one. This shows that the traditionally-known QoS cannot expose the spatial dependencies between stations. Therefore, using Optimal Occupancy is more advantageous than using the traditionally-known QoS. As the BIC for the spatial model demonstrates, Optimal Occupancy as a measurement is more gainful and would result in better prediction of the QoS in a BSS.

Table 8-1. Parameters estimation of the exponential model for Optimal Occupancy and traditionally-known QoS.

	BIC for spatial	**BIC for non-spatial**	**Angle**
Optimal Occupancy	-55.46	-52.12	78°
Traditionally-known QoS	-204.50	-211.10	71°

8.5.3 Optimal Location of New Stations

We proposed Optimal Occupancy as a QoS measurement to: (1) allow the operator to keep track of the performance of different stations in a BSS, so, for example, they may increase the number of docks/available bikes in a station; (2) identify the optimal location of new stations in existing systems using a data-driven decision management approach. In the previous section, geo-statistics were used to model a spatial variogram that could predict the QoS in nearby areas for the purpose of locating new stations in an existing BSS. The model was used to produce new QoS datasets in order to build a QoS surface for the case study area. Figure 8-4 shows the QoS surface for the case study area in San Francisco. This surface could be used to quantify and visualize the QoS measurements represented by contours in the surface. Looking at the surface in Figure 8-4, there are four hot spots (red-colored) that could be considered as candidates to add new stations nearby or increase the number of docks in a station. By considering these candidates, we convert the surface into more homogeneous QoS terrain, which means the BSS will be more functional (i.e., less problematic stations at any given time) and easier to rebalance. It is also interesting to note that during our study, Ford GoBike, the operator of the case study BSS, added different “coming-soon” stations near the aforementioned areas or added more docks to others. For example, a coming-soon station is to be built very near to Station 50, which is shown in Figure 8-1 (a). We hypothesize this station was added to increase the functionality of Station 50 (Ford GoBike, 2018).

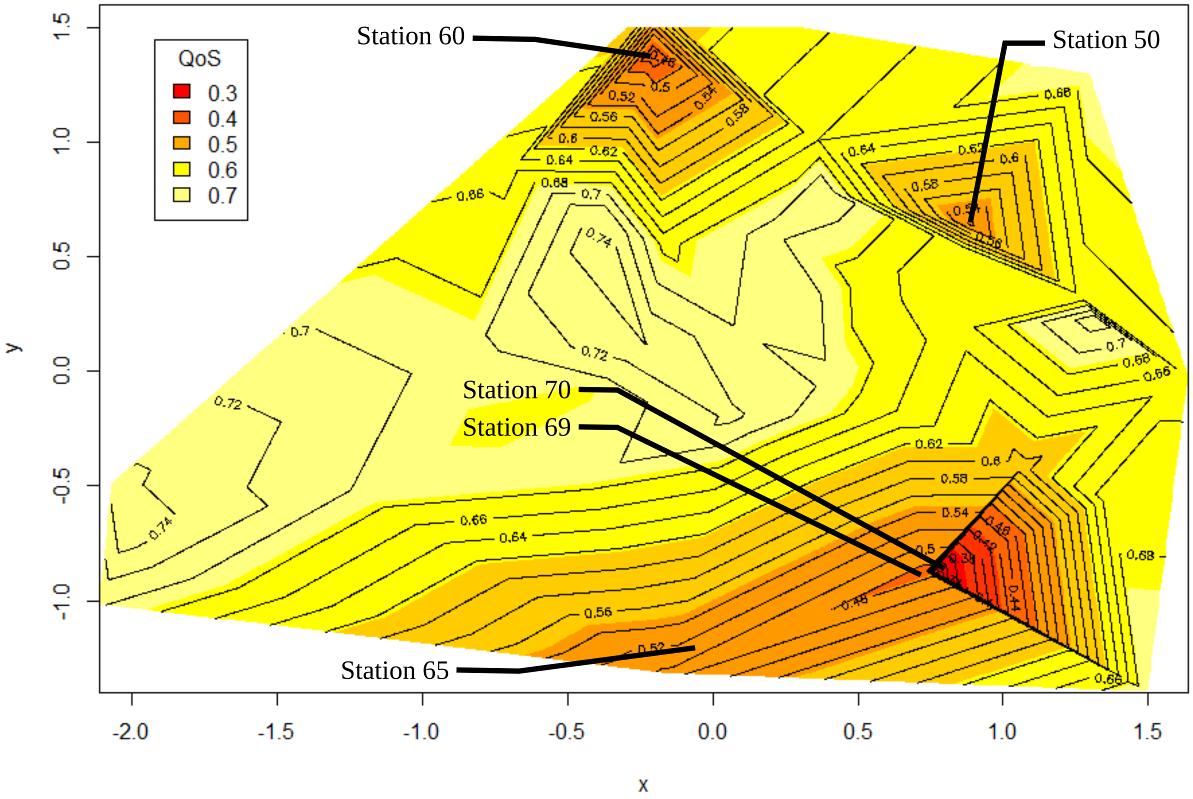


Figure 8-4. Predicted QoS surface for the case study area.

In (H. I. Ashqar, Elhenawy, & Rakha, 2018), a model was developed to predict the bike counts at each station in the Bay Area BSS using RF as a univariate regression algorithm for different prediction horizons. Modeling bike counts using RF produced a MAE of 0.37 bikes/station, which means the model was found to be promising. Station 50, the Harry Bridges Plaza Station, was also found to be one of the highly unpredictable stations due to the large fluctuations in bike counts. When the area around the Harry Bridges Plaza Station was studied, it was hypothesized that this high incoming/outgoing demand is a result of the station being in an open air area at the end of a market and restaurants, where artists, skaters, tourists and others congregate (SF Station, 2017). We used the developed model in (H. I. Ashqar et al., 2018) to prove our hypothesis that adding a new station, for example near Station 50, will increase its functionality. We compared the proposed QoS values for two different days of the week, Monday and Tuesday of July, before and after adding the new suggested station near Station 50. The model was used to predict the bike counts at Station 50 every 15 minutes for each of the selected days to estimate the proposed QoS using Equations (8-5) through (8-12). We assumed that the new station will cover only a third of the two types of services that Station 50 used to serve. The resulting QoS for Station 50 was improved after adding the new suggested station by an increase from 0.52 to 0.84 and from 0.43 to 0.79 for Monday and Tuesday of July, respectively.

8.6 Conclusions

BSS operators tend to spend a great amount of time and effort to satisfy users. Accurately measuring the QoS of each station in a BSS will advance this mission. Moreover, measuring the QoS and using it to study the spatial dependencies in a BSS allows operators to better manage the system. For example, operators can determine candidate spots for new stations that will improve the overall QoS. Consequently, we investigated the traditionally-known QoS measurement and found it to be largely indiscriminative at the station level and not reflective of the spatial correlations. For that reason, we introduced a new QoS measurement, Optimal Occupancy. The Optimal Occupancy at a station is formulated in terms of two types of services: (1) picking up bikes and (2) returning bikes. It is formulated as the ratio of the total time a station is functional during a given interval to the length of the interval. Consequently, we redefined problematic stations as stations that, at any given time, have fewer bikes available than the expected bike counts to be picked up during the analysis discretization duration or more bikes than the difference between capacity and the expected bike counts to be returned during the analysis discretization period.

We further studied the proposed QoS measurement by applying it to a real dataset of 34 stations in San Francisco and also compared the new definition of problematic stations with the previous definition. First, results from ANOVA analysis clearly demonstrate that the traditionally-known QoS cannot be used to discriminate between the stations, whereas the Optimal Occupancy is found to be sufficiently discriminative. Recognition of the differences between the QoS of stations benefits the effective management of the system, and appears to reflect the dynamic nature of the BSS.

Second, we applied geo-statistics to explore the spatial configuration of the Optimal Occupancy variations and model variograms for spatial prediction. The empirical variogram shows a steady increase in semi-variance over increasing distance intervals to an absolute maximum between 1 and 1.5 km. The exponential variogram model was fitted and yields the most beneficial realization of the spatial process in the BSS. Results revealed that the spatial model for Optimal Occupancy outperforms the non-spatial model. Furthermore, Optimal Occupancy as a measurement is more gainful and would result in better prediction for the QoS in nearby locations. The spatial model was used to produce new QoS datasets in order to build a QoS surface for the case study area. Adding new stations near the hot spots in the surface, we were able to convert the surface into a more homogeneous QoS terrain, indicating that the BSS will be more functional and easier to rebalance as a result of this change. For example, the resulting QoS for Station 50 was improved after adding the new suggested station from 0.52 to 0.84 and from 0.43 to 0.79 for Monday and Tuesday of July, respectively.

Chapter 9. A Can Portable Stations Resolve Bike Share System Station Imbalances?

9.1 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 country to totally ban cars in the city center. Instead, public transit and cycling will be supported and encouraged in the banned-car zone, and all 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 ("Proposals to ban cars and taxis from Dublin city centre," 2018).

As an effort by governments to support bicycling and offer alternative transportation modes, BSSs have been introduced over 50 countries (DeMaio, 2009). 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 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 (McNeil, Dill, MacArthur, & Broach, 2017), 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 (Qiu & He, 2018). 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 (Alvarez-Valdes et al., 2016; Angeloudis, Hu, & Bell, 2014; Contardo et al., 2012; Pfrommer et al., 2014; Schuijbroek et al., 2013).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 (Mohammed H Almannaa, Mohammed Elhenawy, Ahmed Ghanem, Huthaifa I Ashqar, & Hesham A Rakha, 2017). 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 (Cui, 2018). 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 9-1), and thus can be easily linked to portable stations. The proposed portable stations can function as either individual stations (standalone) or as an extension of 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 will 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 as much as three times.

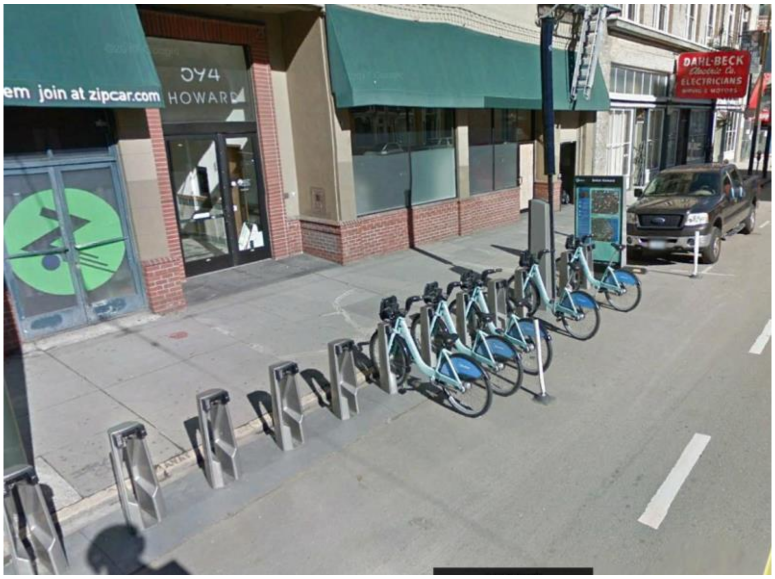


Figure 9-1. Off-street bike station located at 594 Howard St., San Francisco (Source: Google Earth).

9.2 Related Work

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.

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 (Caggiani & Ottomanelli, 2012; Chemla, Meunier, & Calvo, 2013; Espegren et al., 2016). Espegren et al. proposed a model to minimize the deviation from the optimal status of the stations (Espegren et al., 2016). 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 (Caggiani & Ottomanelli, 2012). 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 (Chemla et al., 2013). 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 (Elhenawy & Rakha, 2017b). Kadri and Kacem formulated the balancing problem mathematically with two lower and four upper bonds (Kadri, Kacem, & Labadi, 2018). 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 (Brinkmann, Ulmer, & Mattfeld, 2015; Brinkmann et al., 2016; Chiariotti, Pielli, Zanella, & Zorzi, 2018; Ghosh et al., 2017; Kloimüllner et al., 2014; Regue & Recker, 2014; Vogel & Mattfeld, 2010).

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 (Brinkmann et al., 2016). 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 (Kloimüllner et al., 2014). 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 (Ghosh et al., 2017). The authors 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 (Chiariotti et al., 2018). 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 (Almannaa et al., 2017; Kaltenbrunner et al., 2010). In addition, (Mohammed H Almannaa et al., 2017) 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, a BSS in the San Francisco Bay Area (now operated by Ford as the GoBike BSS) opened in 2013 and the locations of stations have changed significantly since then . 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 (Walteros & Swamy, 2017). In (Walteros & Swamy, 2017) , 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 as well as 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.

9.3 Dataset

This study used the Bay Area'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 (see Figure 3-1): 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 9-2), 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.). 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.

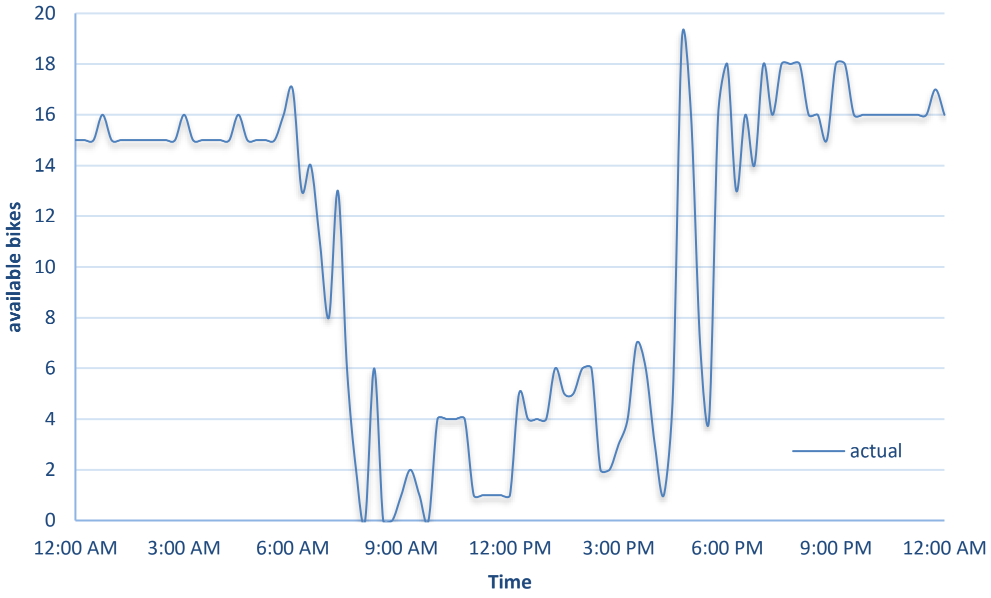


Figure 9-2. Bike counts for randomly selected station during one day.

9.4 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 9-3). 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 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 did not know the arrival rate for other points of interest; we knew 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.

Simulation was conducted every deci-second.

The portable station chooses the next station in a greedy way based on (9-1) and (9-2) as follows:

$$R_{t+1} = S_t - \lambda_{t+1} + {P_{t+1}^{T}}^{T} \lambda_{t+1} \tag{9-1}$$

$$\underset{i}{\text{minarg}}(\left|\frac{Q_j}{2}\right| - H_j + r_i) \tag{9-2}$$

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 (9-1) 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 (9-2), the portable station prefers the station that will keeps it loaded at half capacity after it visits the station.

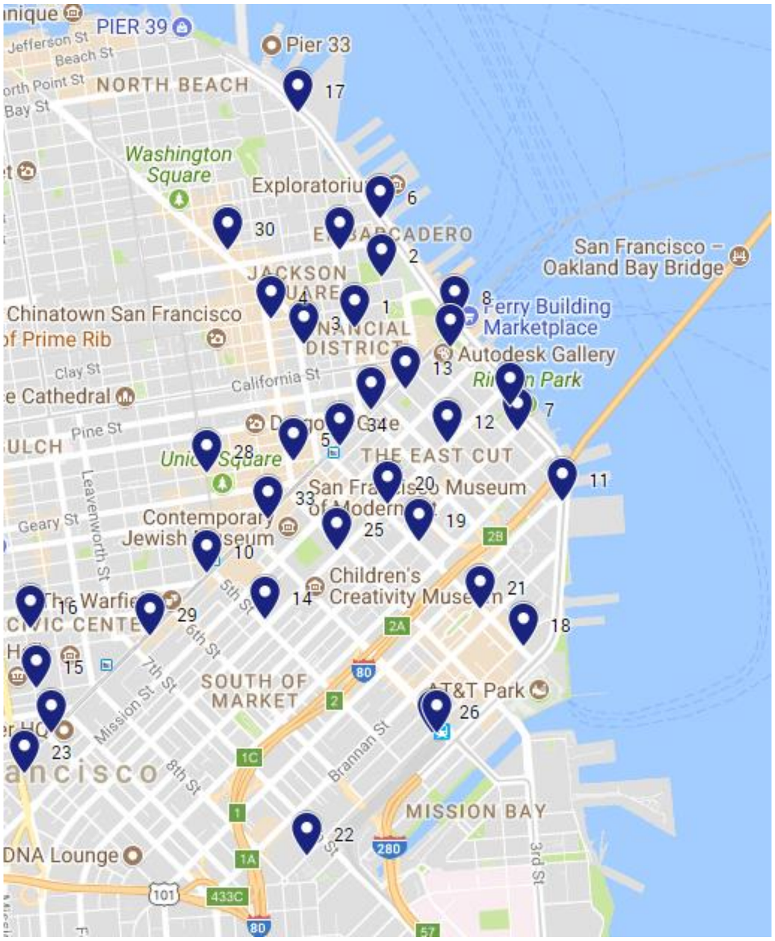


Figure 9-3. Locations of 35 bike stations in downtown San Francisco (Source: Google Maps).

9.5 Model Testing

9.5.1 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 (9-3)

$$\sum_{i=1}^{35} \left| B_i^{start\ of\ the\ day} - B_i^{end\ of\ the\ day} \right| \qquad (9\text{-}3)$$

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.

9.5.2 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 9-4 and Figure 9-5).

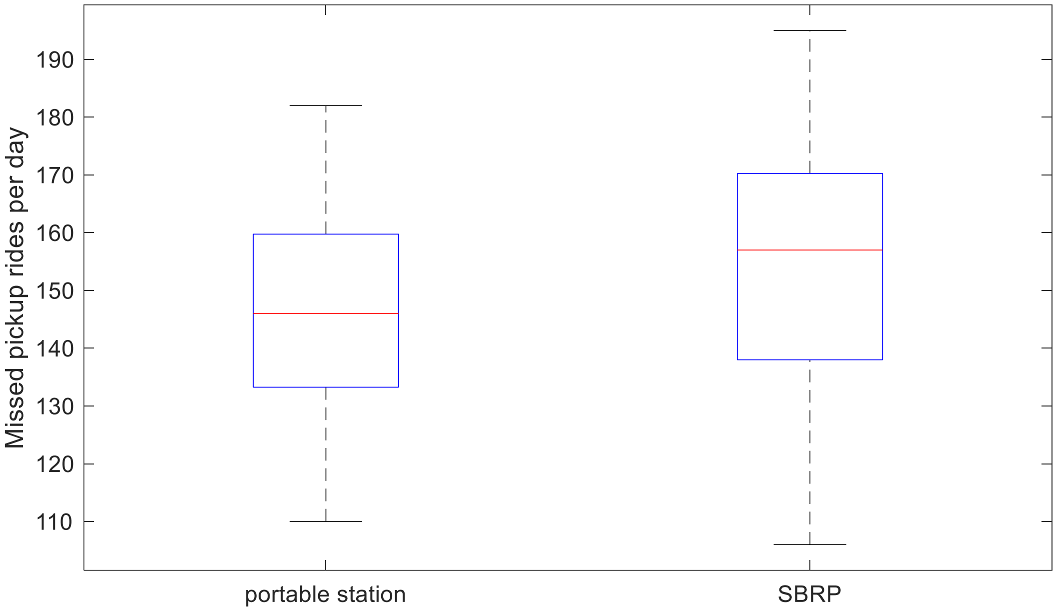


Figure 9-4. Box plot of the average missed bike pick-ups per day for the two approaches: portable stations and SBRP.

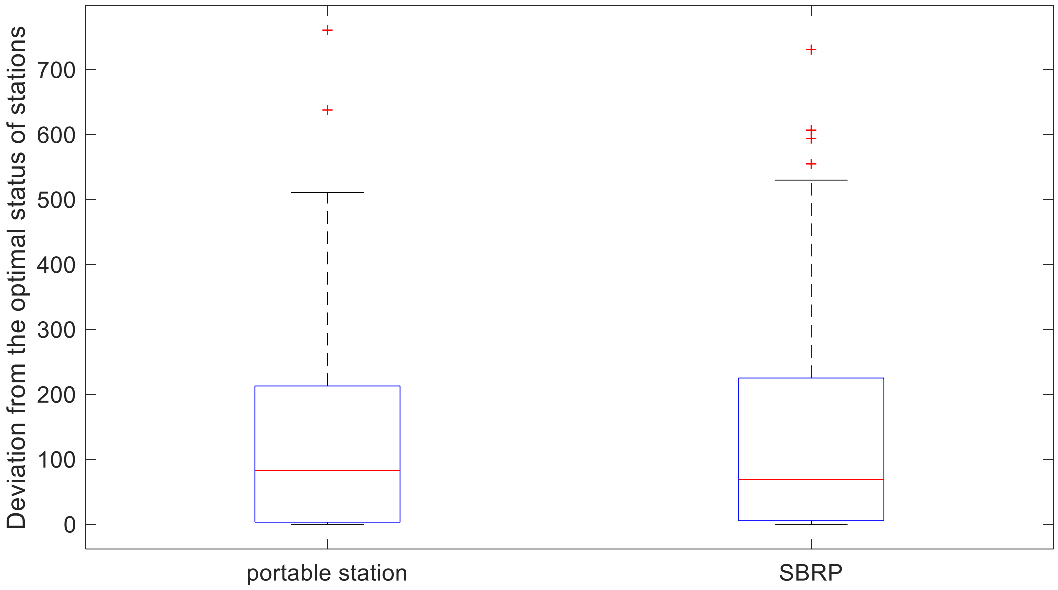


Figure 9-5. Box plot of the average deviation from the optimal status for the two approaches: portable stations and SBRP.

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 did 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 9-6), 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 9-7, 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.

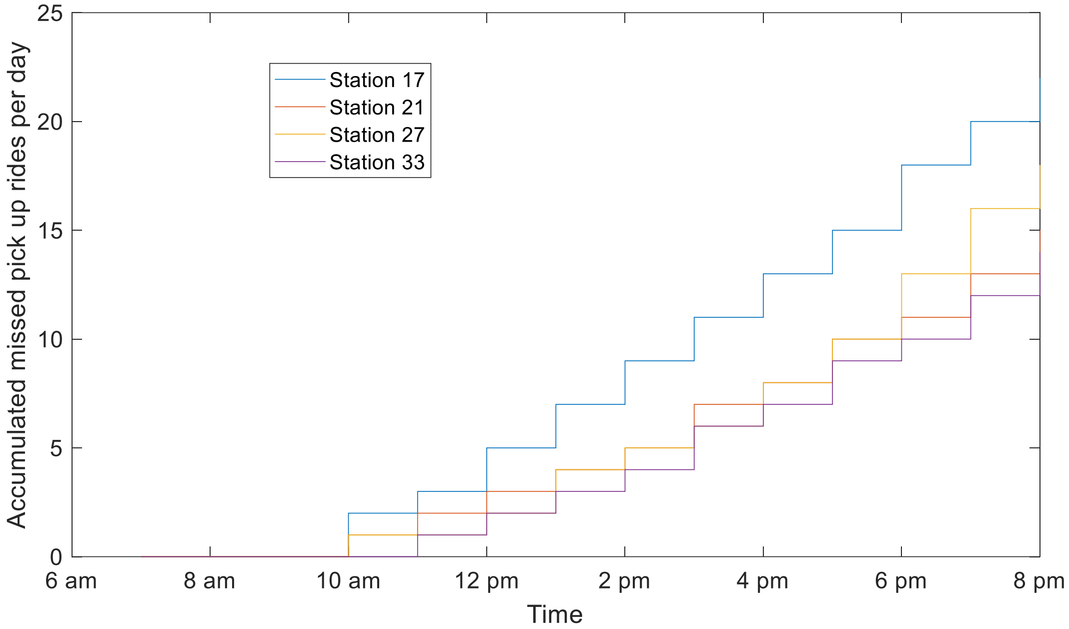


Figure 9-6. Average accumulated missed bike pick-ups per day when using portable station.

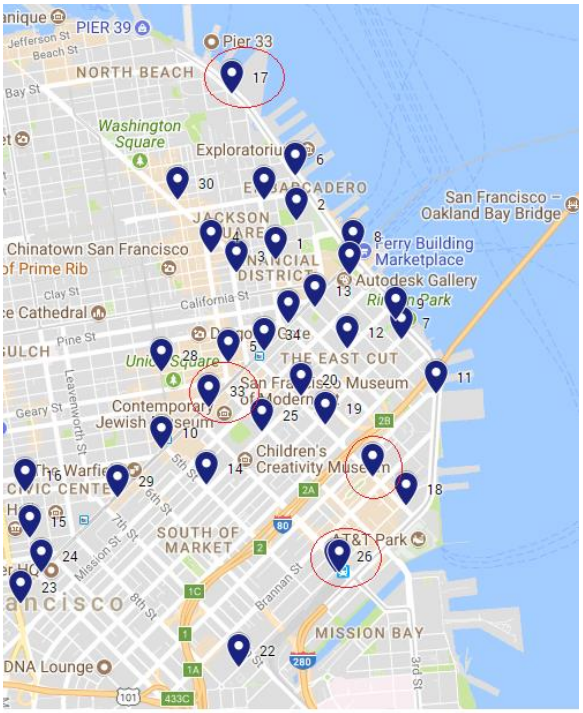


Figure 9-7. Four stations circled in red had 50% of missed bike pick-ups.

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 9-8). The reduction of the missed pick-ups 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 9-9). 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.

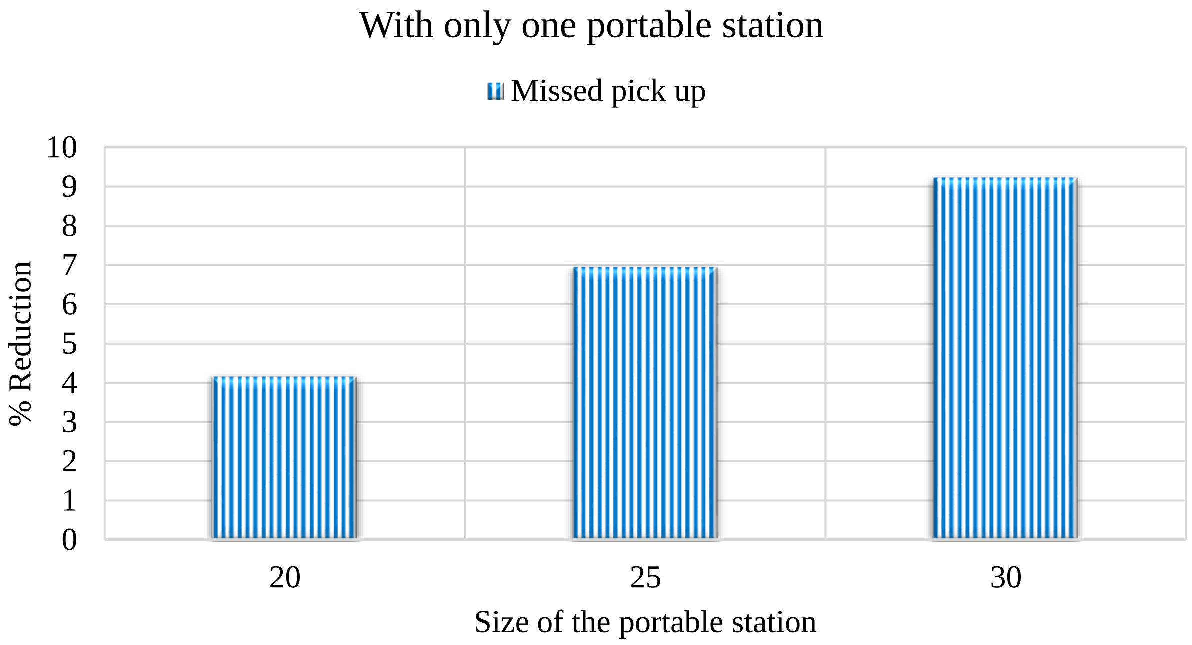


Figure 9-8. The effect of the size of the portable station on the missed bike pick-ups.

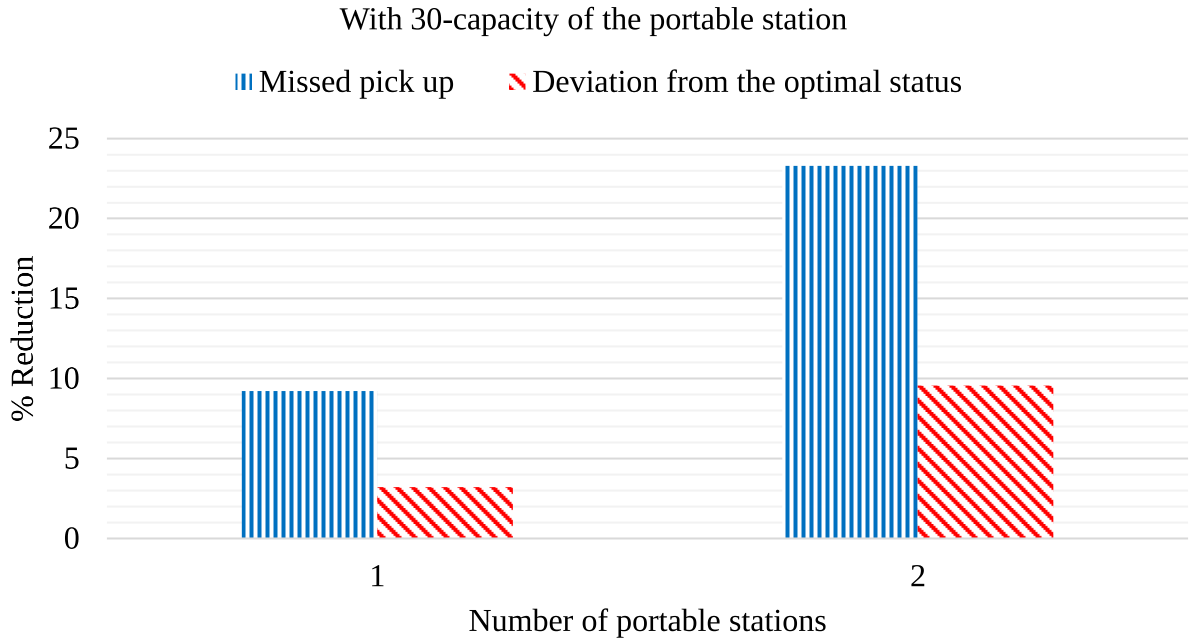


Figure 9-9. The effect of the number of portable stations on missed bike pick-ups and deviation from stations' optimal status.

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.

9.6 Conclusions

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 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 of 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 10. Bike Share Travel Time Modeling: San Francisco Bay Area Case Study

10.1 Introduction and Motivation

BSSs are emerging as a new trend in many urban areas to provide a last-mile solution for short-distance transfers between different private and public transportation modes (Ram et al., 2016). BSSs are not a new service; in fact, they have been in existence for almost five decades (DeMaio, 2009). As of 2014, BSSs were in use in over 700 cities in 50 countries, their numbers having grown rapidly over the past 3 years (Shaheen, Martin, Cohen, Chan, & Pogodzinski, 2014). In the U.S., several major cities, such as New York, NY, Chicago, IL and Seattle, WA, have recently started BSSs. BSSs have also been adopted in populous places such as the Bay Area in CA. These are all examples of BSSs serving areas with high urban densities.

As the number of BSSs across the nation increases, more planning is required to support biking as a trending transportation mode. In order to encourage the increased use of bikes as a mode of transportation, tools, measures, and planning techniques similar to those used for other transportation modes need to be developed. Increased use of bikes and BSSs has many benefits, including lower congestion levels, reduced effects of environmental pollution, and improved BSS user lifestyles.

Roadway congestion levels began to rise again along with the US economy’s recovery from the most recent recession. Congestion levels have not only returned to the pre-recession levels of 2000 and before, but they are now even greater, causing more congestion problems. By 2014, congestion had caused travel delay to increase to 6.9 billion hours per year, up from 5.2 billion hours per year in 2000. Additionally, congestion costs increased by nearly $46 billion between 2000 and 2014, reaching $160 billion in 2014 (Scorecard, 2015). Ideally, the increased presence and use of BSSs will mean decreased congestion levels.

With growing warnings and worries about climate change and increased recommendations to reduce fossil fuel consumption, people are more open to using sustainable transportation modes. Shifting from using motorized transportation modes to the use of sustainable transportation modes, such as driving electric vehicles, biking, and walking, can benefit the environment in many ways, including reducing toxic gas emissions and noise levels. Biking, in particular, can also be an important part of a healthy lifestyle, as it incorporates physical activity into the biker’s daily routine.

Many cities in the US are striving to offer better services to BSS users. They offer cheap rates for trips ranging between 30 and 45 minutes if users subscribe to the system, allowing one bike to serve many users per day. In addition, in order to improve their BSSs, cities and counties are offering public use of their BSS datasets to encourage researchers to analyze them. One of the most important pieces of data in these datasets is travel time.

This paper focuses on predicting BSS travel time considering different predictors, including trip distance, time-of-day, weather conditions, and biker experience. We chose to focus on travel time for a number of reasons. First, many transportation studies addressing multi-modal trip planning and routing try to find a path between points that is optimal according to some criteria. The most obvious of these criteria is optimizing for the shortest path. However, other criteria, including energy cost, number of transfers, and travel time have also been used (Booth, Sistla, Wolfson, & Cruz, 2009). Consequently, predicting bike travel time is vital to these studies. Second, BSS service levels are dependent on the availability of bikes for a pick-up and the availability of docks for a return. In order to maintain the system in a balanced state, bikes need to be moved from stations with more bikes to stations with fewer bikes regularly, in a process called rebalancing. Maintaining a BSS in a balancing state is an NP-hard problem. Many algorithms have been used to solve this problem in an efficient way. However, this requires a great deal of work due to the recurring nature of the problem. Incentivizing BSS users to move bikes between stations as proposed in (Singla et al., 2015) reduces the cost and work of the rebalancing process. Predicting bike travel time can help in the process of incentivized rebalancing by providing an idea about when and where bikes will be available. Third, bike share travel time, to the best of our knowledge, has not been studied a great deal, despite the growing body of literature that focuses on BSSs. Fourth, although some studies have addressed bike speed, none of them have addressed weather conditions, which we argue can have an effect on bike travel time. Going forward, this paper will refer to BSS travel time as bike travel time.

The remainder of this chapter is organized as follows. Related work is discussed in section 2. In section 3, we describe the dataset used. In section 4, we present the methods used for travel time prediction. We show the results and analyses in section 5. Finally, conclusions and future work are discussed in section 6.

10.2 Related Work

Travel time is considered an intuitive performance measure in many Advanced Traveler Information Systems and Advanced Traffic Management Systems. A traveler needs to estimate the trip travel time in order to plan for departure times and make different route decisions by avoiding congested routes. With precise information about the trip travel time, route planner systems can suggest optimal alternative routes. Moreover, transportation agencies can use travel time to manage and control traffic congestion. For the aforementioned reasons and many others, travel time prediction has been considered a hot research topic over the past several decades. Many studies have focused on predicting travel time for vehicular transportation modes (Billings & Yang, 2006; Myung, Kim, Kho, & Park, 2011; Van Lint, Hoogendoorn, & van Zuylen, 2005; C.-H. Wu, Ho, & Lee, 2004; J.-S. Yang, 2005), an obvious choice due to the huge number of vehicles in use around the world. Non-vehicular transport modes, however, have not attracted the same level of interest. To the best of our knowledge, bike travel time has been only sparingly addressed in a few studies in the existing body of relevant literature.

Studies addressing travel time prediction either used travel time as the state variable or used variables such as speed, density or flow as the state variables. El-Geneidy et al. studied bike speed in different types of environments found in urban areas, such as off-street, on-street, and mixed traffic (El-Geneidy, Krizek, & Iacono, 2007). The authors conducted an experiment to collect bike speed data along these different types of facilities, then developed regression models to predict bike speed for different trip characteristics, including biker gender, presence of off-street facility, and biker comfort level. One limitation to their study was that it did not consider the effect of weather conditions on bike speed. Another major drawback was that the study used linear regression models, which assume the normality of the data.

Although the U.S. is considered a relative latecomer to implementing BSSs, according to the Bureau of Transportation Statistics, there were 3,378 BSSs in 104 U.S. cities as of April, 2016 (Firestine, 2016). By comparison, only 54 U.S. cities had deployed BSSs as of April, 2015 (DeMaio, 2009). These numbers show the huge growth in BSSs in the U.S. This growth in BSSs dictates the need for new tools, measures, and planning techniques to be developed for bikes for the benefit of both cities and travelers. First, cities need information about travel demand and bike availability to maintain BSSs in such a way that they offer a certain level of service. The continuously changing nature of a BSS makes maintaining it in a balanced state a difficult task that needs to be addressed periodically by moving bikes between stations, a process referred to as "rebalancing." Second, this information will enable travelers to better plan their trips, make better routing decisions, and reduce associated costs.

The BSS rebalancing process aims at maximizing service availability; i.e., bike availability for pickups and the existence of an empty bike slot for returns. Understanding bike demand and trip patterns is crucial to BSS balancing. Recent studies tried to tackle this problem in different ways. Ram et al. presented a system called SMARTBIKE, which was implemented in Fortaleza, Brazil (Ram et al., 2016). It uses a k-means clustering technique to understand bike demand from historical bike trips and bike station statuses. This system uses a network analysis approach to learn trip flow patterns from historical trip data, then uses these patterns to help BSS managers with the rebalancing process. Singla et al. addressed the same problem by designing a dynamic incentives system for bike rebalancing (Singla et al., 2015). This system uses a smartphone app to get BSS users to engage in the rebalancing process by incentivizing them to move bikes from stations with a higher number of bikes to stations with a lower number of bikes.

Predicting bike travel time is vital to BSSs management. In general, travel time prediction is affected by several predictors, such as traveling speed, traffic flow, and occupancy, all predictors that are highly sensitive to weather conditions and traffic incidents. Bike travel time is also dependent on other important predictors, such as biker experience and physical ability, as shown in (El-Geneidy et al., 2007). For these reasons, predicting travel time accurately, in general, and bike travel time, in particular, is a fairly complex and difficult task requiring a large amount of traffic data. Especially in areas with rapidly changing conditions, an accurate travel time prediction model is essential (van Grol, Lindveld, Manfredi, & Danech-Pajouh, 1999).

In summary, the existing body of literature lacks studies addressing bike travel time, particularly in the context of BSSs. In this paper, we developed different bike travel time prediction models using machine learning techniques. The main contribution of this paper is finding the best predictors to explain bike travel time variability. The techniques used in this paper do not require any assumptions about the data.

10.3 Dataset

As many cities and municipalities adopt BSSs, they are opting to share their datasets publicly to encourage researchers to analyze them. Applying machine and deep learning approaches, artificial intelligence techniques, and statistical analysis are all widely used methods to provide deep insights into the information contained in these datasets. These insights are beneficial for enhancing the BSSs in different ways, including determining the placement of new stations, the number of docks at each station, and the travel patterns associated with different times or days (weekdays or weekends), etc. In the next subsections, we will describe the dataset used in this paper in more detail.

10.3.1 Bay Area BSS Dataset

Among the different BSS datasets that are publicly avail-able, we found the Bay Area BSS dataset to be the richest in content. This dataset is described in detail in Section 3.4 of this report.

In addition to the available trip data, the dataset also includes weather information per day per service area and bike and dock availability per minute per station. This data is available in files named “weather” and “status,” respectively. Additionally, the dataset includes a file named “station,” which contains such information about stations as latitude, longitude, dock count, city, and installation date. Prior to its use, the dataset needed to be preprocessed, cleaned, and imputed for missing data. In the next subsections, we describe how we prepared each file for the analysis stage.

10.3.2 Weather Data

The weather data is only available for five service areas of the Bay Area: San Francisco, San Mateo, Mountain View, Palo Alto, and San Jose. The weather dataset had many discrepancies and missing values. First, we imputed the weather data to fill missing fields. To accomplish this, we used a KNN imputer function written in the R language to impute data with the weighted average of the nearest K = 5 neighbors. Second, we cleaned the precipitation values, as they contained both categorical and numerical data. We replaced all T values with the minimum precipitation value equal to 0.01.

10.3.3 Station Data

The station file contained information about 70 stations distributed among the five aforementioned service areas. Since weather data was only available in these five service areas, denoted by the service area zip code, we assigned each station to one of the five service areas in order to use the weather data at that service area for trips to/from that station. We searched for each station’s zip code in circles of a 2 miles radius around the five service areas. We wanted the radius to be as small as possible so that a single station would be assigned to only one service area. Thus, 67 stations were automatically assigned to one service area. The remaining three stations were assigned manually to their closest service areas.

10.3.4 Trip Data

The trip data includes data from about 669,960 trips. We began by merging the service area weather data to the trips to/from stations within that service area. We found that there were 1,042 trips between stations from different service areas. In this case, we had two sets of weather data (for the source and destination stations) that could potentially be used for these trips. Since we did not have the trajectories of these trips, we could not determine which weather data to use. As a result, we discarded those trips, as they constituted less than 1% of the total trip data. We also excluded trips with the same source and destination, as this would not be relevant to travel time analysis.

Table 10-1. Descriptive statistics of trips data.

	Maximum	**Mean**	**Minimum**
Travel Time (sec)	11455.0	514.1	60.0
Distance (m)	4968	1832	54
Expected Travel Time (sec)	1253.0	479.8	39.0

In order to make an estimation about the trip distance, we used the Google Maps Directions API to get the distance and duration for each trip using bicycle as the travel mode. Of course, this did not actually correspond to the trip route, but was rather used to provide an estimate of the trip distance and duration. We found that there were many trips where the travel time was significantly greater than the estimated duration from Google. Accordingly, we further investigated trips between each source-destination pair of stations. For trips between each pair, we determined the median travel time and the median absolute deviation (MAD) in travel time. We excluded trips whose travel time was greater than (median + 3 × MAD). Using median and MAD is more robust than using mean and standard deviation because the latter pair are more sensitive to outliers, as described in (Leys, Ley, Klein, Bernard, & Licata, 2013). Using median and MAD allowed us to filter outlier trips in which the users kept the bikes for a very long time. These bikes were most likely abandoned before they were restored in the system. We listed more descriptive statistics of the trips data in Table 10-1.

10.4 Methods

To develop the most accurate travel time prediction model, we tried several statistical and machine learning techniques. Initially, we used MLR, then went on to investigate the use of RF, boosting LSBoost, and artificial neural networks (ANN), comparing the techniques in order to find the most accurate model. In this section, we will describe the methods used to analyze the data.

10.4.1 Random Forest

RF, first proposed in (Breiman, 2001; Dressel & Farid, 2018), is one of a number of machine learning ensemble methods. Seni, Elder, & Discovery stated that, “Ensemble methods have been called the most influential development in Data Mining and Machine Learning in the past decade” (Seni, Elder, & Discovery, 2010). In order to get more precise prediction of the target variable, ensembles combine multiple models. RF starts by creating many top-down decision trees. Each tree works on a bootstrapped sample of the original data, and the root node of each tree contains all the data. To grow these trees, RF utilizes a greedy approach referred to as recursive binary splitting. It randomly chooses from the independent variables to split the data in one node to two groups, trying to minimize an objective function. This randomness minimizes the correlation between independent variables. The two groups are considered to be two child nodes of their parent node. This splitting is performed recursively to the largest extent possible. The target variable is predicted by aggregating the predictions of all the trees. In practice, MSE is used for regression.

10.4.2 Least Square Boosting

LSBoost (Freund, Schapire, & Abe, 1999) is also an ensemble method. It refers to a group of algorithms that combine the predictions of individual weak (base) learners into a single strong learner. A weak learner can be as simple as a two terminal node classifiers. LSBoost combines the prediction of weak learners using several methods, such as averaging or weighted averaging. It applies the base learning algorithms iteratively to different versions of the data and builds a learner community $L_k(x), K = 1, 2, ..., K$. LSBoost (J. H. J. A. o. s. Friedman, 2001) fits regression models by minimizing the mean-squared error objective function. At each iteration M, LSBoost fits a new regression model to the difference between the true response and the sum of the prediction of all the $M - 1$ regression models fitted previously.

10.4.3 Artificial Neural Networks

ANNs are used to estimate or approximate unknown linear and non-linear functions that depend on a large number of inputs. ANN is defined in (Hecht-Nielsen, 1992) as a "parallel, distributed information processing structure consisting of processing elements (neurons) interconnected together with unidirectional signal channels called connections." These neurons are organized in layers with the outermost layers being the input layer and the output layer, and the in between layers the hidden layers. Neurons connections allow information to flow in the direction from the input layer to the output layer. As the ANN is trained, the weights of the connections between neurons are modified. ANN uses a learning rule, such as back propagation, to learn these weights.

10.5 Analysis

The Bay Area dataset has 33 different predictors, including weather conditions, distance, time-of-day, day-of-week, and subscription type. We used MLR to model the bike travel time to determine which predictors significantly affect bike travel time. However, we found that the studentized residuals violated the normality assumptions of homoscedasticity. Hence, the model could not be used for making an inference about the predictors. However, it could still be used for prediction purposes. The MLR model could also be used for interpreting the relationships between bike travel time and the predictors. For instance, from physics laws, we know that the distance coefficient should be equal to the reciprocal of the bike speed. To verify the model's correctness, we calculated a distance coefficient of 0.2225 sec/m, which when reciprocated, equals 4.49 m/sec, which is close to the average bike speed used in the literature.

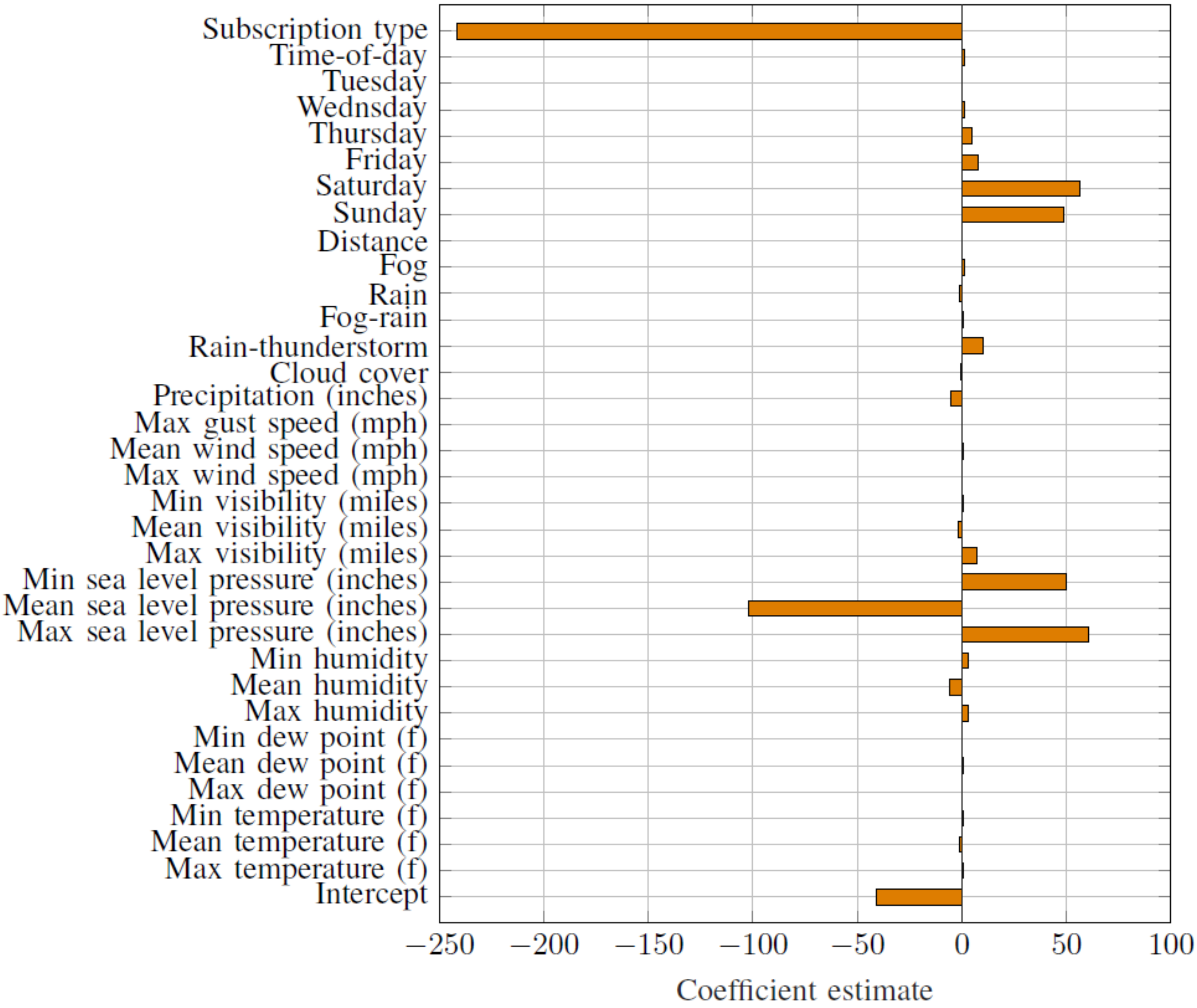


Figure 10-1. Multiple linear regression coefficients estimates.

The estimated coefficients of the MLR model are shown in Figure 10-1. The MLR model has a MAE of 100.36 sec and a MAPE of 21.3%. Its R2 is 0.61, which shows that the model explains only 61% of the bike travel time variability. Due to the MLR model's inadequacy, we decided to adopt different machine learning techniques to predict the bike travel time.

10.5.1 Random Forest

RF does not assume normality of the data. RF improves the decision tree by growing more trees, which work on different bootstrapped samples of the data, thus decreasing the correlation between trees, which in turn decreases the variance of the RF model. The number of trees is one of the RF parameters that requires tuning. Breiman recommended using P/3 trees for models with P predictors when RF is used for regression. To identify the optimum number of trees, we varied the number of trees and also tried to enhance the models by using log transformation to reduce the skewness of the bike travel time distribution. We found that the models of the log-transformed data had a lower prediction error. To validate our models, we used K-cross validation with K = 5.

The RF models' MAEs and MAPEs are shown in Figures 2 and 3, respectively. As the number of trees in the model increases, the prediction error decreases. Increasing the number of trees beyond 100 does not seem to decrease the prediction error. As the figures show, the models of the log-transformed bike travel time have a lower prediction error. The RF model of log-transformed data and 100 trees has a MAE and MAPE of 84.01 sec and 16.92%, respectively, whereas the model of the original data and the same number of trees has a MAE and MAPE of 85.28 sec and 18.10%, respectively.

10.5.2 Least Square Boosting

We used Matlab to build the LSBoost models and chose the tree as the weak learner. The number of weak learners is an LSBoost parameter that needs to be tuned. We performed a sensitivity analysis to calibrate the number of trees. As done earlier, we used a log transformation to see whether it would improve the prediction models. We also used 5-fold cross validation.

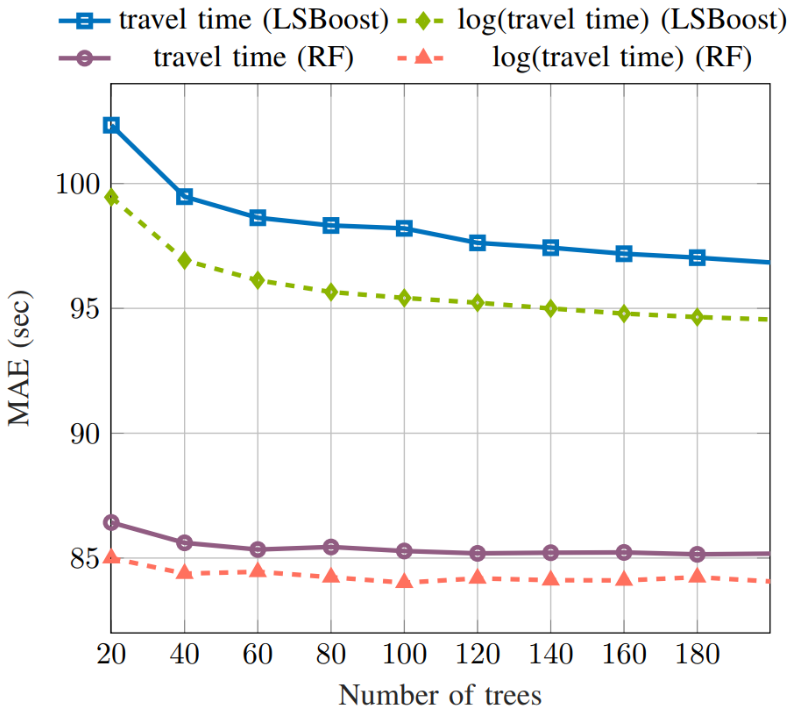


Figure 10-2. Random forest and LSBoost MEA\absolute error at different number of trees.

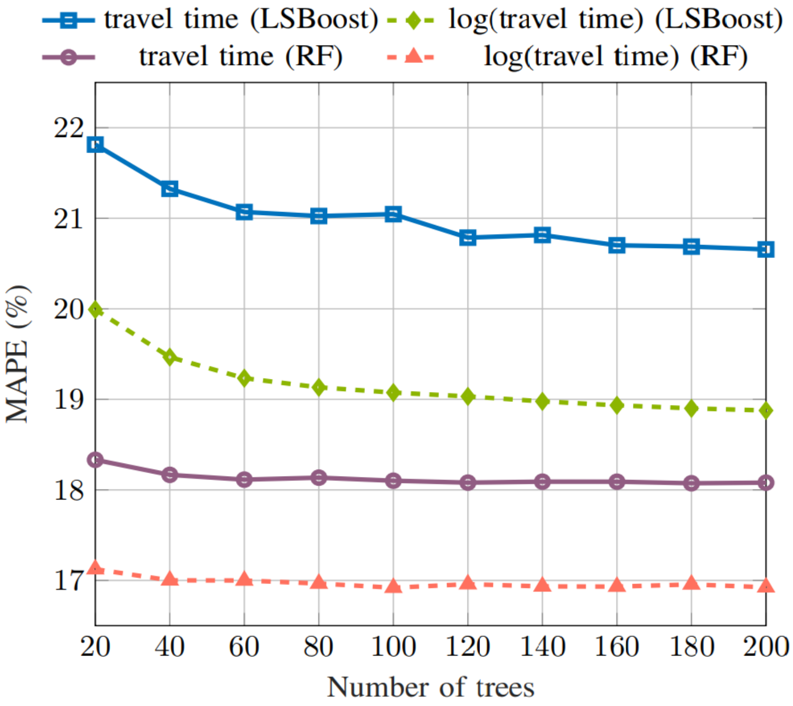


Figure 10-3. RF and LSBoost MAPE at different number of trees.

The LSBoost models' MAEs and MAPEs are shown in Figure 10-2 and Figure 10-3, respectively. Unsurprisingly, the models of log-transformed data outperformed the models of the original data. As indicated in both figures, the RF models have a lower prediction error compared to the LSBoost models. The LSBoost model of log-transformed bike travel time and 100 trees has a MAE and MAPE of 95.42 sec and 19.07%, respectively.

10.5.3 Artificial Neural Network

For the sake of comparison, we implemented a feed-forward ANN with two hidden layers. For simplicity, we used an equal number of neurons in each hidden layer. We varied the number of neurons from two to nine. The implemented ANN was trained using the discriminative pre-training technique (D. Yu, Deng, Seide, & Li, 2016). The prediction error of the ANN for both travel time and the log-transformed travel time as the response are shown in Table 10-2.

Comparing the prediction errors of the previous three learning algorithms, we found that the RF models outperformed the other models. Consequently, we decided to adopt an RF model for bike travel time prediction. The large number of predictors motivated us to reduce the RF model because, most likely, these predictors will not be readily available. In the next subsection, we will describe the model reduction procedure.

Table 10-2. Mean absolute error for artificial neural network model.

MAE (sec)								
Number of Neurons	2	3	4	5	6	7	8	9
Travel Time	99.79	99.55	99.56	99.47	99.04	99.05	100.01	99.18
Log(Travel Time)	97.36	97.05	96.65	96.67	96.61	96.30	96.38	96.25
MAPE (%)								
Travel Time	21.08	21.08	21.13	21.13	20.88	20.94	21.32	21.00
Log(Travel Time)	19.61	19.56	19.48	19.47	19.47	19.39	19.43	19.37

10.5.4 Model Reduction

We noticed that the models developed in the previous subsections used many redundant weather predictors, such as mean, minimum, and maximum temperature, etc. Consequently, we decided to choose the most intuitive weather predictors, such as mean temperature, mean humidity, mean visibility, and precipitation. We used RF to rank the new subset of 18 predictors including only 9 weather predictors. The predictors' importance is shown in Figure 10-4.

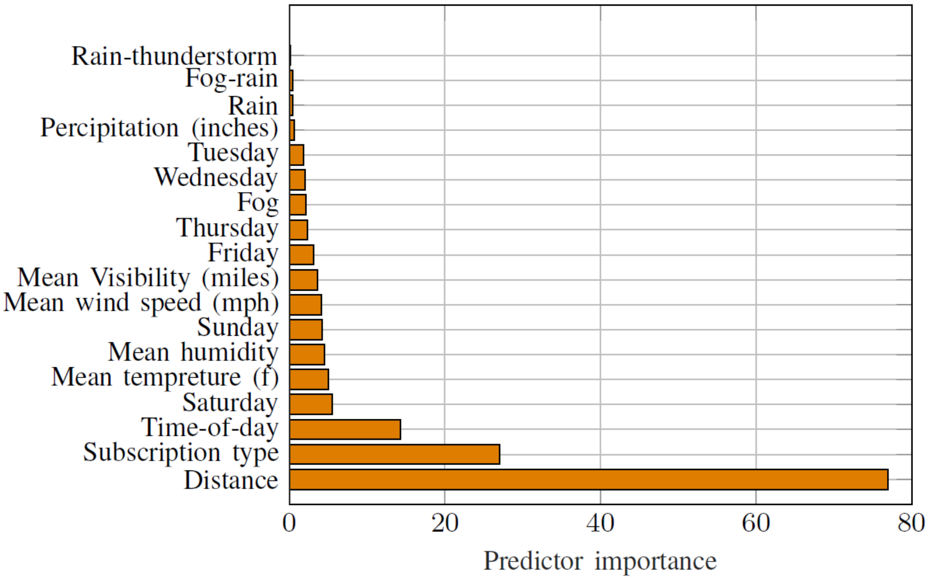


Figure 10-4. Importance of reduced set predictors.

Finally, we used forward stepwise regression to select the best subset of predictors to model the bike travel time. As shown in Figure 10-5, we found that the least error occurred with a subset of seven predictors. Based on our analysis, the best predictors of bike travel time were distance, subscription type, time-of-day, Saturday, mean temperature, mean humidity, and Sunday, respectively. That distance was the most important predictor is clearly explained by the laws of physics. The subscription type predictor is indicative of the bikers' familiarity with the road network and ability to exert power, directly affecting bike travel time. This is consistent with the conclusions of (El-Geneidy et al., 2007) that time-of-day, Saturday, and Sunday predictors affect the travel time can be clearly attributed to traffic conditions.

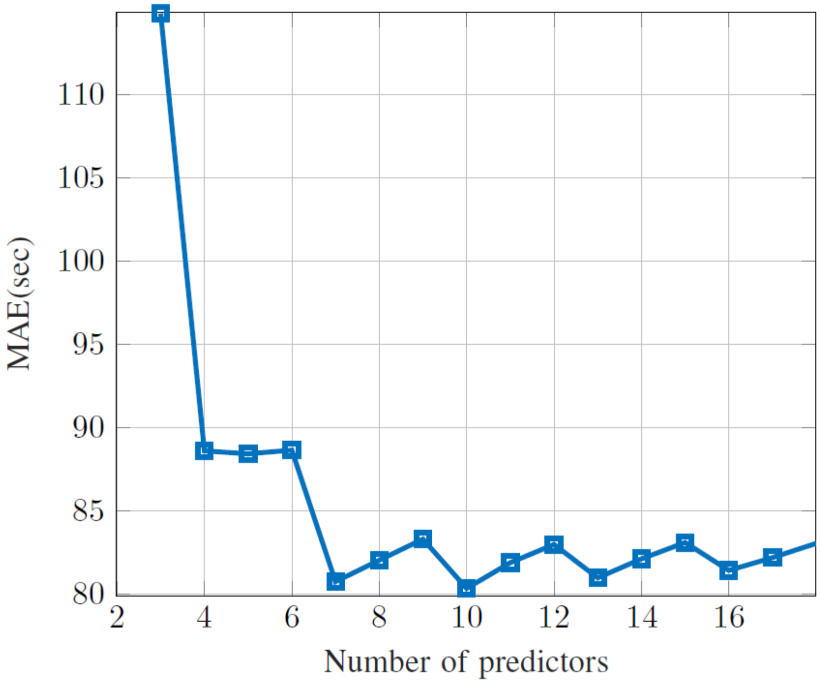


Figure 10-5. Stepwise MAE for different number of predictors.

The model shows that weather conditions, specifically, mean temperature and mean humidity, have a large impact on the bike travel time. This is because the body loses fluids when exerting effort (biking), especially when it's hot and humid. This is in accordance with the findings of different studies (Barr, 1999; Murray, 2007). The model using only these seven predictors reduced the MAE to 82.04 sec and the MAPE to 16.2%.

10.6 Conclusions

We used RF, LSBoost and ANN techniques to build bike travel time prediction models. We determined that RF models outperform the other models. We used RF and forward stepwise regression to reduce the number of predictors that explain the bike travel time variability, and found that the most important subset of predictors includes intuitive predictors. These predictors, ordered by importance, are travel distance, subscription type, time-of-day, Saturday, mean temperature, mean humidity, and Sunday. We also found that different weather conditions, particularly temperature and humidity, have an effect on bike travel time.

Chapter 11. Overall Study Conclusions

In this report, we defined and showed how smart bike sharing systems are important for connecting different transportation networks in a smart city in order to establish smart transportation. Moreover, we made eight contributions toward building a toolbox of models and algorithms to convert a current BSS into a smart bike sharing system.

The first contribution (Chapter 1) proposes a two-layer hierarchical framework classifier to distinguish between five transportation modes using new extracted frequency domain features pooled with traditionally used time domain features. We investigated the possibility of improving the classification accuracy using pooled features in the proposed framework by applying several techniques: KNN, CART, SVM, RF, and RF-SVM. The results show that using pooled features in the proposed framework increased the classification accuracy for all the applied classifiers. For the same data, the highest reported accuracy was 95.10% using the traditional approach for detection, whereas the proposed approach in this study achieved an accuracy of 97.02%.

The second contribution (Chapter 2) proposes a new supervised clustering algorithm that will potentially assist agencies and researchers anticipate bike availability at stations with respect to a time event. The proposed algorithm, tested on a BSS in the San Francisco Bay Area, 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 smart cities.

The third contribution (Chapter 3) describes the development of a bike availability model for the San Francisco Bay Area BSS. Since the demand of bikes in stations is still not well studied, this contribution introduced a fast, effective approach, which is also accurate and reasonable, to quantifying the effect of various features on bike counts at different stations. The results revealed that the bike count changes with the month-of-the-year, day-of-the-week, time-of-day, and some weather variables. This model could also be used to improve the redistribution of bicycles, which is important for rebalancing the network over a period of time.

The fourth contribution (Chapter 4) builds a Markov chain model for each station and day of the week. We investigated the daily imbalances and identified an optimal inventory level to minimize the probability of a station reaching an empty or full state. Our analysis showed 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. 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 the San Francisco BSS stations 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.

The fifth contribution (Chapter 5, Chapter 6, Chapter 7) adapts state-of-the-art machine learning and statistical algorithms to model the number of available bikes. We applied these algorithms to the Bay Area Bike Share stations in San Francisco. First, we tried two approaches: using univariate regression algorithms, RF and LSBoost, and using a multivariate regression algorithm, PLSR. The univariate models were used to model the number of available bikes at each station. RF, with an MAE of 0.37 bikes/station, outperformed LSBoost, with an MAE of 0.58 bikes/station. On the other hand, the multivariate model, PLSR, was applied to model available bikes at spatially correlated stations of each region obtained from the trip's adjacency matrix. Results clearly showed that the univariate models produced lower error predictions compared to the multivariate model, in which the MAE was approximately 0.6 bikes/station. However, the multivariate model results might be acceptable and reasonable when modeling the number of available bikes in BSS networks with a relatively large number of stations. Investigating BSS networks in terms of determined regions gives new insights to policy makers. The fact that stations in each region derived by the multivariate analysis share the same zip code implies that most of the trips were short distance trips. This may be influenced by the overtime fees applied when trips are longer than 30 minutes. With the most effective prediction horizon being 15 minutes, determining prediction horizon is beneficial to policy makers and technicians for learning how to manage BSSs more responsively, and achieving better prediction performance.

Second, we adapted dynamic linear and incremental learning models 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. We compared the results of the online and incremental learning algorithms with the machine learning algorithms RF and 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 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). 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.

The sixth contribution (Chapter 8) investigates the traditionally-known QoS measurement. We found QoS to be largely indiscriminate at the station level and not reflective of the spatial correlations. For that reason, we introduced a new QoS measurement: Optimal Occupancy. The Optimal Occupancy at a station is formulated in terms of two types of services: (1) picking up bikes and (2) returning bikes. Our results from ANOVA analysis clearly demonstrate that the traditionally-known QoS cannot be used to discriminate between the stations, whereas the Optimal Occupancy is found to be sufficiently discriminative. Recognition of the differences between the QoS of stations benefits the effective management of the system, and appears to reflect the dynamic nature of the BSS.

The seventh contribution (Chapter 9) investigates the advantage of having portable bike stations, using an agent-based simulation approach as a proof-of-concept. Our 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.

The eight contribution (Chapter 10) builds bike travel time prediction models using RF, LSBoost and ANN techniques. We determined that RF models outperform the other models. We used RF and forward stepwise regression to reduce the number of predictors that explain bike travel time variability, and found that the most important subset of predictors includes intuitive predictors. These predictors, ordered by importance, are travel distance, subscription type, time-of-day, Saturday, mean temperature, mean humidity and Sunday. We also found that different weather conditions, particularly temperature and humidity, have an effect on bike travel time.

References

Abraham, J. E., McMillan, S., Brownlee, A. T., & Hunt, J. D. (2002, 2002). *Investigation of cycling sensitivities*.

Almannaa, M., Elhenawy, M., & Rakha, H. (2019). A Novel Supervised Clustering Algorithm for Transportation System Applications. *IEEE transactions on intelligent transportation systems*.

Almannaa, M. H., Elhenawy, M., Ghanem, A., Ashqar, H. I., & Rakha, H. A. (2017, 26-28 June 2017). *Network-wide bike availability clustering using the college admission algorithm: A case study of San Francisco Bay area.* Paper presented at the 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS).

Almannaa, M. H., Elhenawy, M., Ghanem, A., Ashqar, H. I., & Rakha, H. A. (2017). *Network-wide bike availability clustering using the college admission algorithm: A case study of San Francisco Bay area.* Paper presented at the Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2017 5th IEEE International Conference on.

Almannaa, M. H., Elhenawy, M., & Rakha, H. A. (2018). *Predicting Bike Availability in Bikesharing Systems Using Dynamic Linear Models*. Retrieved from

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

Angeloudis, P., Hu, J., & Bell, M. G. (2014). A strategic repositioning algorithm for bicycle-sharing schemes. *Transportmetrica A: Transport Science, 10*(8), 759-774.

Arbelaez, P., Maire, M., Fowlkes, C., & Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence, 33*(5), 898-916.

Ashqar, H., Elhenawy M., Almannaa M., Ghanem A., H., R., & L., a. H. (2017). *Modeling bike availability in a bike-sharing system using machine learning*. Paper presented at the 5th IEEE International Conference on MODELS AND TECHNOLOGIES FOR INTELLIGENT TRANSPORTATION SYSTEMS, Napoli, Italy.

Ashqar, H. I., Elhenawy, M., Almannaa, M. H., Ghanem, A., & Rakha, H. A. (2018). *Quantifying the Effect of Various Features on the Modeling of Bike Counts in a Bike-Sharing System*. Paper presented at the 97th Transportation Research Board Annual Meeting, Washington DC.

Ashqar, H. I., Elhenawy, M., Almannaa, M. H., Ghanem, A., Rakha, H. A., & House, L. (2017). *Modeling bike availability in a bike-sharing system using machine learning.* Paper presented at the Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2017 5th IEEE International Conference on.

Ashqar, H. I., Elhenawy, M., Ghanem, A., Almannaa, M. H., & Rakha, H. A. (2016). *Modeling Bike Counts in a Bike-Sharing System Considering the Effect of Weather Conditions*.

Ashqar, H. I., Elhenawy, M., Ghanem, A., Almannaa, M. H., & Rakha, H. A. (2018). *Quantifying the Effect of Various Features on the Modeling of Bike Counts in a Bike-Sharing System*. Retrieved from

Ashqar, H. I., Elhenawy, M., & Rakha, H. A. (2018). Network and Station-Level Bike-Sharing System Prediction: A San Francisco Bay Area Case Study. *unpublished*.

Awasthi, P., & Zadeh, R. B. (2010). *Supervised clustering.* Paper presented at the Advances in neural information processing systems.

Banerjee, S., Carlin, B. P., & Gelfand, A. E. (2014). *Hierarchical modeling and analysis for spatial data*: Crc Press.

Bar-Hillel, A., Hertz, T., Shental, N., & Weinshall, D. (2003). *Learning distance functions using equivalence relations.* Paper presented at the Proceedings of the 20th International Conference on Machine Learning (ICML-03).

Barr, S. I. J. C. J. o. A. P. (1999). Effects of dehydration on exercise performance. *24*(2), 164-172.

Barutçuoğlu, Z., & Alpaydın, E. (2003). A comparison of model aggregation methods for regression. In *Artificial Neural Networks and Neural Information Processing—ICANN/ICONIP 2003* (pp. 76-83): Springer.

Basu, S., Banerjee, A., & Mooney, R. (2002). *Semi-supervised clustering by seeding*. Paper presented at the In Proceedings of 19th International Conference on Machine Learning (ICML-2002.

Basu, S., Bilenko, M., & Mooney, R. J. (2003). *Comparing and unifying search-based and similarity-based approaches to semi-supervised clustering.* Paper presented at the Proceedings of the ICML-2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining.

Bay Area Bike Share. Introducing Bay Area Bike Share, Your New Regional Transit

System. Retrieved from www.bayareabikeshare.com/faq#BikeShare101

Biljecki, F., Ledoux, H., & Van Oosterom, P. (2013). Transportation mode-based segmentation and classification of movement trajectories. *International Journal of Geographical Information Science, 27*(2), 385-407.

Billings, D., & Yang, J.-S. (2006). *Application of the ARIMA models to urban roadway travel time prediction-a case study*. Paper presented at the Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on.

Bolbol, A., Cheng, T., Tsapakis, I., & Haworth, J. (2012). Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification. *Computers, Environment and Urban Systems*.

Booth, J., Sistla, P., Wolfson, O., & Cruz, I. F. (2009). *A data model for trip planning in multimodal transportation systems.* Paper presented at the Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology.

Bordagaray, M., dell'Olio, L., Fonzone, A., & Ibeas, Á. (2016). Capturing the conditions that introduce systematic variation in bike-sharing travel behavior using data mining techniques. *Transportation Research Part C: Emerging Technologies, 71*, 231-248. Retrieved from http://www.sciencedirect.com/science/article/pii/S0968090X16301176. doi:https://doi.org/10.1016/j.trc.2016.07.009

Borgnat, P., Abry, P., Flandrin, P., Robardet, C., Rouquier, J.-B., & Fleury, E. (2011). Shared bicycles in a city: A signal processing and data analysis perspective. *Advances in Complex Systems, 14*(03), 415-438.

Borgnat, P., Fleury, E., Robardet, C., & Scherrer, A. (2009, 2009-09-21). *Spatial analysis of dynamic movements of Vélo'v, Lyon's shared bicycle program.* Paper presented at the ECCS'09, Warwick, United Kingdom.

Breiman, L. (2001). Random forests. *Machine learning, 45*(1), 5-32.

Brinkmann, J., Ulmer, M. W., & Mattfeld, D. C. (2015). Short-term strategies for stochastic inventory routing in bike sharing systems. *Transportation Research Procedia, 10*, 364-373.

Brinkmann, J., Ulmer, M. W., & Mattfeld, D. C. (2016). Inventory Routing for Bike Sharing Systems. *Transportation Research Procedia, 19*, 316-327. Retrieved from http://www.sciencedirect.com/science/article/pii/S2352146516308778. doi:https://doi.org/10.1016/j.trpro.2016.12.091

Caggiani, L., Camporeale, R., Ottomanelli, M., & Szeto, W. Y. (2018). A modeling framework for the dynamic management of free-floating bike-sharing systems. *Transportation Research Part C: Emerging Technologies, 87*, 159-182. Retrieved from http://www.sciencedirect.com/science/article/pii/S0968090X18300020. doi:https://doi.org/10.1016/j.trc.2018.01.001

Caggiani, L., & Ottomanelli, M. (2012). A modular soft computing based method for vehicles repositioning in bike-sharing systems. *Procedia-Social and Behavioral Sciences, 54*, 675-684.

Calafate, C. T., Soler, D., Cano, J.-C., & Manzoni, P. (2015). Traffic Management as a Service: The Traffic Flow Pattern Classification Problem. *Mathematical Problems in Engineering, 2015*, 14. Retrieved from http://dx.doi.org/10.1155/2015/716598. doi:10.1155/2015/716598

Cameron, A. C., & Trivedi, P. K. (1986). Econometric models based on count data. Comparisons and applications of some estimators and tests. *Journal of applied econometrics*, *1*(1), 29-53.

Cameron, A. C., & Trivedi, P. K. (2013). *Regression analysis of count data* (Vol. 53): Cambridge university press.

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

Chen, N., Chen, W.-N., Gong, Y.-J., Zhan, Z.-H., Zhang, J., Li, Y., & Tan, Y.-S. (2015). An evolutionary algorithm with double-level archives for multiobjective optimization. *IEEE transactions on cybernetics, 45*(9), 1851-1863.

Chiariotti, F., Pielli, C., Zanella, A., & Zorzi, M. (2018). A dynamic approach to rebalancing bike-sharing systems. *Sensors, 18*(2), 512.

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

Cui, W. (2018). *The Effects of Urban Density on the Efficiency of Dockless Bike Sharing System-A Case Study of Beijing, China.* Arizona State University,

Daddio, D. W. (2012). Maximizing Bicycle Sharing: an empirical analysis of capital bikeshare usage.

Daddio, D. W., & and Mcdonald, N. (2012). Maximizing bicycle sharing: an empirical analysis of capital bikeshare usage.

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

Demiriz, A., Bennett, K. P., & Embrechts, M. J. (1999). Semi-supervised clustering using genetic algorithms. *Artificial neural networks in engineering (ANNIE-99)*, 809-814.

Demiryurek, U., Pan, B., Banaei-Kashani, F., & Shahabi, C. (2009). *Towards modeling the traffic data on road networks*. Paper presented at the Proceedings of the Second International Workshop on Computational Transportation Science.

Dill, J., & Voros, K. (2007). Factors affecting bicycling demand: initial survey findings from the Portland, Oregon, region. *Transportation Research Record: Journal of the Transportation Research Board*(2031), 9-17.

Dressel, J., & Farid, H. (2018). The accuracy, fairness, and limits of predicting recidivism. *Science Advances, 4*(1). Retrieved from http://advances.sciencemag.org/content/advances/4/1/eaao5580.full.pdf. doi:10.1126/sciadv.aao5580

Eick, C. F., Zeidat, N., & Zhao, Z. (2004). *Supervised clustering-algorithms and benefits*. Paper presented at the Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on.

El-Geneidy, A. M., Krizek, K. J., & Iacono, M. (2007). *Predicting bicycle travel speeds along different facilities using GPS data: A proof of concept model.* Paper presented at the Proceedings of the 86th Annual Meeting of the Transportation Research Board, Washington, DC, USA.

Elhenawy, M., Chen, H., & Rakha, H. A. (2014). Dynamic travel time prediction using data clustering and genetic programming. *Transportation Research Part C: Emerging Technologies, 42*, 82-98. Retrieved from http://www.sciencedirect.com/science/article/pii/S0968090X14000588. doi:http://dx.doi.org/10.1016/j.trc.2014.02.016

Elhenawy, M., Jahangiri, A., & Rakha, H. A. (2016). *Smartphone Transportation Mode Recognition using a Hierarchical Machine Learning Classifier*. Paper presented at the 23rd ITS World Congress, MELBOURNE AUSTRALIA. https://www.researchgate.net/publication/301338082

Elhenawy, M., & Rakha, H. (2017a). *Applying Cluster Analysis Techniques to Traffic Operations*. Retrieved from http://vtrc.virginiadot.org/ProjDetails.aspx?Id=613

Elhenawy, M., & Rakha, H. (2017b). *A heuristic for rebalancing bike sharing systems based on a deferred acceptance algorithm.* Paper presented at the Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2017 5th IEEE International Conference on.

Elhenawy, M., & Rakha, H. A. (2015). Automatic Congestion Identification with Two-Component Mixture Models. *Transportation Research Record: Journal of the Transportation Research Board, 2489*, 11-

19. Retrieved from http://trrjournalonline.trb.org/doi/abs/10.3141/2489-02. doi:doi:10.3141/2489-02

Espegren, H. M., Kristianslund, J., Andersson, H., & Fagerholt, K. (2016). *The Static Bicycle Repositioning Problem-Literature Survey and New Formulation.* Paper presented at the International Conference on Computational Logistics.

Etienne, C., & Latifa, O. (2014a). Model-Based Count Series Clustering for Bike Sharing System Usage Mining: A Case Study with the Velib' System of Paris. *ACM Transactions on Intelligent Systems and Technology, 5*(3), 1-21. doi:10.1145/2560188

Etienne, C., & Latifa, O. (2014b). Model-Based Count Series Clustering for Bike Sharing System Usage Mining: A Case Study with the Velib' System of Paris. *ACM Trans. Intell. Syst. Technol., 5*(3), 1-21. doi:10.1145/2560188

Etienne, C., & Latifa, O. (2014). 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), 5*(3), 39.

Fedor-Freybergh, P. G., & Mikulecký, M. (2005). From the descriptive towards inferential statistics. Hundred years since conception of the Student's t-distribution. *Neuroendocrinol Lett, 26*, 167-171.

Feng, C., Hillston, J., & Reijsbergen, D. (2017). Moment-based availability prediction for bike-sharing systems. *Performance Evaluation, 117*, 58-74.

Finley, T., & Joachims, T. (2005). *Supervised clustering with support vector machines.* Paper presented at the Proceedings of the 22nd international conference on Machine learning.

Firestine, T. (2016). Bike-Share Stations in the United States.

Ford GoBike. (2018). Station map. Retrieved from https://member.fordgobike.com/map/

Forestier, G., Gançarski, P., & Wemmert, C. (2010). Collaborative clustering with background knowledge. *Data & Knowledge Engineering, 69*(2), 211-228. Retrieved from http://www.sciencedirect.com/science/article/pii/S0169023X09001463. doi:http://dx.doi.org/10.1016/j.datak.2009.10.004

Frade, I., & Ribeiro, A. (2015). Bike-sharing stations: A maximal covering location approach. *Transportation Research Part A: Policy and Practice, 82*(Supplement C), 216-227. Retrieved from http://www.sciencedirect.com/science/article/pii/S0965856415002487. doi:https://doi.org/10.1016/j.tra.2015.09.014

Freund, Y., Schapire, R., & Abe, N. J. J.-J. S. F. A. I. (1999). A short introduction to boosting. *14*(771-780), 1612.

Fricker, C., & Gast, N. (2016). Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity. *EURO Journal on Transportation and Logistics, 5*(3), 261-291. Retrieved from http://dx.doi.org/10.1007/s13676-014-0053-5. doi:10.1007/s13676-014-0053-5

Fricker, C., Gast, N., & Mohamed, H. (2012). *Mean field analysis for inhomogeneous bike sharing systems*.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232.

Friedman, J. H., Baskett, F., & Shustek, L. J. (1974). A relatively efficient algorithm for finding nearest neighbors. *IEEE Trans. Comput., 24*(SLAC-PUB-1448), 1000-1006.

Friedman, J. H. J. A. o. s. (2001). Greedy function approximation: a gradient boosting machine. 1189-1232.

Froehlich, J., Neumann, J., & Oliver, N. (2009a, 2009). *Sensing and Predicting the Pulse of the City through Shared Bicycling*.

Froehlich, J., Neumann, J., & Oliver, N. (2009b). *Sensing and Predicting the Pulse of the City through Shared Bicycling.* Paper presented at the IJCAI.

Gale, D., & Shapley, L. S. (1962). College admissions and the stability of marriage. *The American Mathematical Monthly, 69*(1), 9-15.

Gallop, C., Tse, C., & Zhao, J. (2011). A seasonal autoregressive model of Vancouver bicycle traffic using weather variables. *i-Manager's Journal on Civil Engineering, 1*(4), 9.

Ganjisaffar, Y., Caruana, R., & Lopes, C. V. (2011). *Bagging gradient-boosted trees for high precision, low variance ranking models*.

García-Palomares, J. C., Gutiérrez, J., & Latorre, M. (2012). Optimizing the location of stations in bike-sharing programs: A GIS approach. *Applied Geography, 35*(1), 235-246. Retrieved from http://www.sciencedirect.com/science/article/pii/S0143622812000744. doi:https://doi.org/10.1016/j.apgeog.2012.07.002

Gast, N., Massonnet, G., Reijsbergen, D., & Tribastone, M. (2015). *Probabilistic forecasts of bike-sharing systems for journey planning.* Paper presented at the Proceedings of the 24th ACM International on Conference on Information and Knowledge Management.

Geladi, P., & Kowalski, B. R. (1986). Partial least-squares regression: a tutorial. *Analytica Chimica Acta, 185*, 1-17. Retrieved from http://www.sciencedirect.com/science/article/pii/0003267086800289. doi:http://dx.doi.org/10.1016/0003-2670(86)80028-9

Ghosh, S., Varakantham, P., Adulyasak, Y., & Jaillet, P. (2017). Dynamic Repositioning to Reduce Lost Demand in Bike Sharing Systems. *Journal of Artificial Intelligence Research, 58*, 387-430.

Gunasekaran, A., Patel, C., & Tirtiroglu, E. (2001). Performance measures and metrics in a supply chain environment. *International journal of operations & production Management, 21*(1/2), 71-87.

Hamner, B. (2016). SF Bay Area Bike Share | Kaggle. Retrieved from https://www.kaggle.com/benhamner/sf-bay-area-bike-share

Handl, J., & Knowles, J. (2007). An evolutionary approach to multiobjective clustering. *IEEE transactions on Evolutionary Computation, 11*(1), 56-76.

Harrison, J., & West, M. (1999). *Bayesian forecasting & dynamic models* (Vol. 1030): Springer New York City.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics), 28*(1), 100-108.

Hecht-Nielsen, R. (1992). Theory of the backpropagation neural network. In *Neural networks for perception* (pp. 65-93): Elsevier.

Höskuldsson, A. (1988). PLS regression methods. *Journal of chemometrics, 2*(3), 211-228.

Hsu, C.-W., & Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on, 13*(2), 415-425.

Jahangiri, A., & Rakha, H. A. (2015). Applying machine learning techniques to transportation mode recognition using mobile phone sensor data. *IEEE transactions on intelligent transportation systems, 16*(5), 2406-2417.

Jahangiri, A., Rakha, H. A., & Dingus, T. A. (2015). *Adopting Machine Learning Methods to Predict Red-light Running Violations.* Paper presented at the Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on.

Jerez, J. M., Molina, I., García-Laencina, P. J., Alba, E., Ribelles, N., Martín, M., & Franco, L. (2010). Missing data imputation using statistical and machine learning methods in a real breast cancer problem. *Artificial intelligence in medicine, 50*(2), 105-115.

Johansen, S., & Juselius, K. (1990). Maximum likelihood estimation and inference on cointegration—with applications to the demand for money. *Oxford Bulletin of Economics and statistics, 52*(2), 169-210.

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika, 32*(3), 241-254.

Kadri, A. A., Kacem, I., & Labadi, K. (2018). Lower and upper bounds for scheduling multiple balancing vehicles in bicycle-sharing systems. *Soft Computing*, 1-22.

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

Kloimüllner, C., Papazek, P., Hu, B., & Raidl, G. R. (2014). *Balancing bicycle sharing systems: an approach for the dynamic case.* Paper presented at the European Conference on Evolutionary Computation in Combinatorial Optimization.

Kwapisz, J. R., Weiss, G. M., & Moore, S. A. (2011). Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter, 12*(2), 74-82.

Law, M. H., Topchy, A. P., & Jain, A. K. (2004). *Multiobjective data clustering*. Paper presented at the Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on.

Leys, C., Ley, C., Klein, O., Bernard, P., & Licata, L. J. J. o. E. S. P. (2013). Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *49*(4), 764-766.

Lin, J.-R., & Yang, T.-H. (2011). Strategic design of public bicycle sharing systems with service level constraints. *Transportation Research Part E: Logistics and Transportation Review, 47*(2), 284-294.

Liu, Y., Szeto, W. Y., & Ho, S. C. (2018). A static free-floating bike repositioning problem with multiple heterogeneous vehicles, multiple depots, and multiple visits. *Transportation Research Part C: Emerging Technologies, 92*, 208-242. Retrieved from http://www.sciencedirect.com/science/article/pii/S0968090X18301761. doi:https://doi.org/10.1016/j.trc.2018.02.008

Loh, W. Y. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1*(1), 14-23.

Long, J. S., & Freese, J. (2006). *Regression models for categorical dependent variables using Stata*: Stata press.

Lu, C.-C. (2016). Robust multi-period fleet allocation models for bike-sharing systems. *Networks and Spatial Economics, 16*(1), 61-82.

MacDonald, D., & Fyfe, C. (2000). *The kernel self-organising map*. Paper presented at the Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000. Proceedings. Fourth International Conference on.

Maity, A., & Sherman, M. (2012). Testing for Spatial Isotropy Under General Designs. *Journal of statistical planning and inference, 142*(5), 1081-1091. Retrieved from http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3275644/. doi:10.1016/j.jspi.2011.11.013

Manzoni, V., Maniloff, D., Kloeckl, K., & Ratti, C. (2010). *Transportation mode identification and real-time CO2 emission estimation using smartphones*. Retrieved from

Marcu, D. (2005). A Bayesian model for supervised clustering with the Dirichlet process prior. *Journal of Machine Learning Research, 6*(Sep), 1551-1577.

McNeil, N., Dill, J., MacArthur, J., & Broach, J. (2017). Breaking Barriers to Bike Share: Insights from Bike Share Users.

Meng, L. D. O. (2011). Implementing bike-sharing systems. *Proceedings of the Institution of Civil Engineers, 164*(2), 89.

Mohanty, S. P., Choppali, U., & Kougianos, E. (2016). Everything you wanted to know about smart cities: The internet of things is the backbone. *IEEE Consumer Electronics Magazine, 5*(3), 60-70.

Monti, S., Tamayo, P., Mesirov, J., & Golub, T. (2003). Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine learning, 52*(1-2), 91-118.

Most Humid Cities in USA - Current Results. (2016). Retrieved from https://www.currentresults.com/Weather-Extremes/US/most-humid-cities.php

Murray, B. J. J. o. t. A. C. o. N. (2007). Hydration and physical performance. *26*(sup5), 542S-548S.

Myung, J., Kim, D.-K., Kho, S.-Y., & Park, C.-H. J. T. R. R. (2011). Travel time prediction using k nearest neighbor method with combined data from vehicle detector system and automatic toll collection system. *2256*(1), 51-59.

National Geodetic Survey. (2017). Universal Transverse Mercator Coordinates. Retrieved from https://geodesy.noaa.gov/TOOLS/utm.shtml#

Ngai, E., Hu, Y., Wong, Y., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems, 50*(3), 559-569.

Nham, B., Siangliulue, K., & Yeung, S. (2008). *Predicting mode of transport from iphone accelerometer data*. Retrieved from

Nick, T., Coersmeier, E., Geldmacher, J., & Goetze, J. (2010). *Classifying means of transportation using mobile sensor data*. Paper presented at the Neural Networks (IJCNN), The 2010 International Joint Conference on.

Olshen, L., & Stone, C. J. (1984). Classification and regression trees. *Wadsworth International Group, 93*(99), 101.

Pal, A., & Zhang, Y. (2017). Free-floating bike sharing: Solving real-life large-scale static rebalancing problems. *Transportation Research Part C: Emerging Technologies, 80*, 92-116. Retrieved from http://www.sciencedirect.com/science/article/pii/S0968090X17300992. doi:https://doi.org/10.1016/j.trc.2017.03.016

Pebesma, E. J. (2004). Multivariable geostatistics in S: the gstat package. *Computers & Geosciences, 30*(7), 683-691.

Petris, G., Petrone, S., & Campagnoli, P. (2009). *Dynamic Linear models with R*. University of Arkansas, Fayetteville AR: Springer.

Pfrommer, J., Warrington, J., Schildbach, G., & Morari, M. (2014). Dynamic vehicle redistribution and online price incentives in shared mobility systems. *IEEE transactions on intelligent transportation systems, 15*(4), 1567-1578.

Proposals to ban cars and taxis from Dublin city centre. (2018). Retrieved from https://www.joe.ie/news/proposals-to-ban-cars-and-taxis-from-dublin-city-centre-499064

Qiu, L.-Y., & He, L.-Y. (2018). Bike Sharing and the Economy, the Environment, and Health-Related Externalities. *Sustainability, 10*(4), 1145.

Ram, S., Dong, F., Currim, F., Wang, Y., Dantas, E., & Sabóia, L. A. (2016). *Smartbike: Policy making and decision support for bike share systems*. Paper presented at the Smart Cities Conference (ISC2), 2016 IEEE International.

Raviv, T., & Kolka, O. (2013). Optimal inventory management of a bike-sharing station. *IIE Transactions, 45*(10), 1077-1093.

Raviv, T., Tzur, M., & Forma, I. A. (2013). Static repositioning in a bike-sharing system: models and solution approaches. *EURO Journal on Transportation and Logistics, 2*(3), 187-229. Retrieved from http://dx.doi.org/10.1007/s13676-012-0017-6. doi:10.1007/s13676-012-0017-6

Reddy, S., Mun, M., Burke, J., Estrin, D., Hansen, M., & Srivastava, M. (2010). Using Mobile Phones to Determine Transportation Modes. *Acm Transactions on Sensor Networks, 6*(2). Retrieved from <Go to ISI>://WOS:000275163100004. doi:10.1145/1689239.1689243

Reddy, S., Mun, M., Burke, J., Estrin, D., Hansen, M., & Srivastava, M. (2010). Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks (TOSN), 6*(2), 13.

Regue, R., & Recker, W. (2014). Proactive vehicle routing with inferred demand to solve the bikesharing rebalancing problem. *Transportation Research Part E: Logistics and Transportation Review, 72*, 192-209.

Ribeiro Jr, P. J., & Diggle, P. J. (2001). geoR: a package for geostatistical analysis. *R news, 1*(2), 14-18.

Rixey, R. (2013). Station-level forecasting of bikesharing ridership: station network effects in three US systems. *Transportation Research Record: Journal of the Transportation Research Board*(2387), 46-55.

Roberts, J. A. (1995). Profiling levels of socially responsible consumer behavior: a cluster analytic approach and its implications for marketing. *Journal of marketing Theory and practice, 3*(4), 97-117.

Rudloff, C., & Lackner, B. (2013a). Modeling demand for bicycle sharing systems–neighboring stations as a source for demand and a reason for structural breaks. *Transportation Research Record: Journal of the Transportation Research Board*(2430), 1-11.

Rudloff, C., & Lackner, B. (2013b, 2013). *Modeling demand for bicycle sharing systems–neighboring stations as a source for demand and a reason for structural breaks*.

Rudloff, C., & Lackner, B. (2014). 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*(2430), 1-11.

Saltzman, R. M., & Bradford, R. M. (2016). Simulating a More Efficient Bike Sharing System. *Journal of Supply Chain and Operations Management, 14*(2), 36.

Saridis, G., & Stein, G. (1968). Stochastic approximation algorithms for linear discrete-time system identification. *IEEE Transactions on Automatic Control, 13*(5), 515-523.

Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation, 10*(5), 1299-1319.

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

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

Schuijbroek, J., Hampshire, R. C., & van Hoeve, W. J. (2017). Inventory rebalancing and vehicle routing in bike sharing systems. *European Journal of Operational Research, 257*(3), 992-1004. Retrieved from http://www.sciencedirect.com/science/article/pii/S0377221716306658. doi:https://doi.org/10.1016/j.ejor.2016.08.029

Scorecard, U. M. J. I. G. S. (2015). The Texas A&M Transportation Institute and Inrix.

Şenbabaoğlu, Y., Michailidis, G., & Li, J. Z. (2014). Critical limitations of consensus clustering in class discovery. *Scientific reports, 4*, 6207.

Seni, G., Elder, J. F. J. S. L. o. D. M., & Discovery, K. (2010). Ensemble methods in data mining: improving accuracy through combining predictions. *2*(1), 1-126.

SF Station. (2017). Harry Bridges Plaza. Retrieved from https://www.sfstation.com/harry-bridges-plaza-b7616

Shafizadeh, K., & Niemeier, D. (1997). Bicycle journey-to-work: travel behavior characteristics and spatial attributes. *Transportation Research Record: Journal of the Transportation Research Board*(1578), 84-90.

Shaheen, S. A., Martin, E. W., Cohen, A. P., Chan, N. D., & Pogodzinski, M. (2014). Public Bikesharing in North America During a Period of Rapid Expansion: Understanding Business Models, Industry Trends & User Impacts, MTI Report 12-29.

Singla, A., Santoni, M., Bartók, G., Mukerji, P., Meenen, M., & Krause, A. (2015). *Incentivizing Users for Balancing Bike Sharing Systems.* Paper presented at the AAAI.

Sinkkonen, J., Kaski, S., & Nikkilä, J. (2002). *Discriminative clustering: Optimal contingency tables by learning metrics.* Paper presented at the European Conference on Machine Learning.

Spinelli, V. (2017). Supervised box clustering. *Advances in Data Analysis and Classification, 11*(1), 179-204.

Stenneth, L., Wolfson, O., Yu, P. S., & Xu, B. (2011a). *Transportation mode detection using mobile phones and GIS information.* Paper presented at the Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.

Stenneth, L., Wolfson, O., Yu, P. S., & Xu, B. (2011b). *Transportation mode detection using mobile phones and GIS information.* Paper presented at the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM SIGSPATIAL GIS 2011, November 1, 2011 - November 4, 2011, Chicago, IL, United states.

Susan, S., Stacey, G., & Hua, Z. (2010). Bikesharing in Europe, the Americas, and Asia. *Transportation Research Record*, 159-167.

Susi, M., Renaudin, V., & Lachapelle, G. (2013). Motion Mode Recognition and Step Detection Algorithms for Mobile Phone Users. *Sensors, 13*(2), 1539-1562. Retrieved from http://dx.doi.org/10.3390/s130201539. doi:10.3390/s130201539

Tan, P.-N. (2006). *Introduction to data mining*: Pearson Education India.

Thiprungsri, S., & Vasarhelyi, M. A. (2011). Cluster analysis for anomaly detection in accounting data: An audit approach.

van Grol, R., Lindveld, K., Manfredi, S., & Danech-Pajouh, M. (1999). *DACCORD: On-line travel time estimation/prediction results.* Paper presented at the Proceedings of Sixth World Congress on Intelligent Transport Systems (ITS), Toronto.

Van Lint, J., Hoogendoorn, S., & van Zuylen, H. J. J. T. R. P. C. E. T. (2005). Accurate freeway travel time prediction with state-space neural networks under missing data. *13*(5-6), 347-369.

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

Vogel, P., Greiser, T., & Mattfeld, D. C. (2011b). Understanding Bike-Sharing Systems using Data Mining: Exploring Activity Patterns. *Procedia - Social and Behavioral Sciences, 20*, 514-523. Retrieved from http://www.sciencedirect.com/science/article/pii/S1877042811014388. doi:http://dx.doi.org/10.1016/j.sbspro.2011.08.058

Vogel, P., & Mattfeld, D. C. (2010). *Modeling of repositioning activities in bike-sharing systems.* Paper presented at the World conference on transport research (WCTR).

Walteros, J., & Swamy, R. (2017). Locating Portable Stations to Support the Operation of Bike Sharing Systems.

Wang, B., & Kim, I. (2018). Short-term prediction for bike-sharing service using machine learning. *Transportation Research Procedia, 34*, 171-178.

Wang, X., Lindsey, G., Schoner, J. E., & Harrison, A. (2015). Modeling bike share station activity: Effects of nearby businesses and jobs on trips to and from stations. *Journal of Urban Planning and Development, 142*(1), 04015001.

Weijermars, W., & van Berkum, E. (2005). *Analyzing highway flow patterns using cluster analysis.* Paper presented at the Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.

West, M., Harrison, P. J., & Migon, H. S. (1985). Dynamic generalized linear models and Bayesian forecasting. *Journal of the American Statistical Association, 80*(389), 73-83.

Widhalm, P., Nitsche, P., & Brandie, N. (2012). *Transport mode detection with realistic Smartphone sensor data.* Paper presented at the 2012 21st International Conference on Pattern Recognition (ICPR 2012), 11-15 Nov. 2012, Piscataway, NJ, USA.

Wit, E., Heuvel, E. v. d., & Romeijn, J. W. (2012). 'All models are wrong...': an introduction to model uncertainty. *Statistica Neerlandica, 66*(3), 217-236.

Wold, H. (1982). Soft modelling: the basic design and some extensions. *Systems under indirect observation, Part II*, 36-37.

Wold, S., Ruhe, A., Wold, H., & Dunn, I. W. (1984). The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing, 5*(3), 735-743.

Wold, S., Sjöström, M., & Eriksson, L. (2001). PLS-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems, 58*(2), 109-130. Retrieved from //www.sciencedirect.com/science/article/pii/S0169743901001551. doi:http://dx.doi.org/10.1016/S0169-7439(01)00155-1

Wu, C.-H., Ho, J.-M., & Lee, D.-T. J. I. t. o. i. t. s. (2004). Travel-time prediction with support vector regression. *5*(4), 276-281.

Wu, Z.-d., Xie, W.-x., & Yu, J.-p. (2003). *Fuzzy c-means clustering algorithm based on kernel method.* Paper presented at the Computational Intelligence and Multimedia Applications, 2003. ICCIMA 2003. Proceedings. Fifth International Conference on.

Xu, C., Ji, J., Liu, P., & Peng, L. (2018). *Forecasting the Travel Demand of the Station-Free Sharing Bike Using a Deep Learning Approach*. Retrieved from

Xu, D., & Tian, Y. (2015). A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science, 2*(2), 165-193. Retrieved from https://doi.org/10.1007/s40745-015-0040-1. doi:10.1007/s40745-015-0040-1

Yang, H., Wang, Z., Xie, K., Ozbay, K., & Ma, Y. (2018). *Use of Deep Learning to Predict Daily Usage of Bike Sharing Systems*. Retrieved from

Yang, J.-S. (2005). *Travel time prediction using the GPS test vehicle and Kalman filtering techniques.* Paper presented at the American Control Conference, 2005. Proceedings of the 2005.

Yoon, J. W., Pinelli, F., & Calabrese, F. (2012). *Cityride: a predictive bike sharing journey advisor*.

Yu, D., Deng, L., Seide, F. T. B., & Li, G. (2016). Discriminative pretraining of deep neural networks. In: Google Patents.

Yu, X., Low, D., Bandara, T., Pathak, P., Hock Beng, L., Goyal, D., . . . Ben-Akiva, M. (2012, 14-17 Jan. 2012). *Transportation activity analysis using smartphones.* Paper presented at the Consumer Communications and Networking Conference (CCNC), 2012 IEEE.

Zhang, L., Qiang, M., & Yang, G. (2013). Mobility transportation mode detection based on trajectory segment. *Journal of Computational Information Systems, 9*(8), 3279-3286.

Zheng, Y., Liu, L., Wang, L., & Xie, X. (2008). *Learning transportation mode from raw gps data for geographic applications on the web*. Paper presented at the Proceedings of the 17th international conference on World Wide Web, Beijing, China.