Towards Efficient Sharing: A Usage Balancing Mechanism for Bike Sharing Systems

Shuai Wang
Key Lab of Computer Network and Information Integration
Southeast University
shuaiwang@seu.edu.cn

Tian He
University of Minnesota
tianhe@umn.edu

Desheng Zhang
Rutgers University
desheng.zhang@cs.rutgers.edu

Yunhuai Liu
Peking University
yunhuai.liu@gmail.com

Sang H. Son
Daegu Gyeongbuk Institute of Science and Technology
don@dgist.ac.kr

ABSTRACT

With the rapid development of sharing economy, massive sharing systems such as Uber, Airbnb, and bikeshare have percolated into people's daily life. The sharing economy, at its core, is to achieve efficient use of resources. The actual usage of shared resources, however, is unclear to us. Little measurement or analysis, if any, has been conducted to investigate the resource usage status with the large-scale data collected from these sharing systems. In this paper, we analyze the bike usage status in three typical bikeshare systems based on 140-month multi-event data. Our analysis shows that the most used 20% of bikes account for 45% of usage, while the least used 20% of bikes account for less than 1% of usage. To efficiently utilize shared bikes, we propose a usage balancing design called eShare which has three components: (i) a statistical model based on archived data to infer historical usage; (ii) an entropy-based prediction model based on both real-time and archived data to infer future usage; (iii) a model-driven optimal calibration engine for bike selection to dynamically balance usage. We develop an ID swapping based evaluation methodology and measure the efficiency of eShare with data from three systems including the world's largest bikeshare system with 84,000 bikes and 3,300 stations. Our results show that eShare not only fully utilizes shared bikes but also improves service quality.

CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile computing**.

KEYWORDS

Bike sharing; usage balancing; sharing economy; urban data

ACM Reference Format:
Shuai Wang, Tian He, Desheng Zhang, Yunhuai Liu, and Sang H. Son. 2019. Towards Efficient Sharing: A Usage Balancing Mechanism for Bike Sharing Systems. In *Proceedings of the 2019 World Wide Web Conference (WWW'19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313441

WWW'19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313441

1 INTRODUCTION

Bike sharing is widely adopted in major cities around the world as a new way of travel [7]. In a bikeshare system, people pick up a bike from one station and drop off it at another station, providing great convenience to people's daily commute. As of June 2014, bikeshare systems have been launched in 712 cities, operating approximately 806,200 bicycles at 37,500 stations [22].

As a sharing platform, one of the key concerns is to efficiently utilize the shared bike resources. However, our data analysis on three representative bikeshare systems shows that inefficient bike usage is a common phenomenon in existing bikeshare systems. For example, the statistic results on NiceRide bikeshare system show that the most used 20% of bikes account for 45% of the total rentals, while the least used 20% of bikes account for less than 1% of the total rentals. This is because bikes in current systems are arbitrarily assigned to users, ignoring both cumulative service time of bikes and potential rental duration of users.

The current bike sharing mechanism has several drawbacks. First, a large amount of bikes in the system are rarely utilized but continually introducing maintenance costs while the unsatisfied status of the frequently used bikes leads to complaints on service quality. Second, the irregular or excessive bike usage contributes to bike depreciation, leading to a huge amount of damaged bikes. Per the statistics of Capital bikeshare system, 9,284 bikes are damaged because of heavy riding from Nov 2014 to Oct 2015, which costs about 3 million US dollars in inspecting/repairing those bikes [5]. In addition, massive damaged bikes appear in busy seasons (e.g., from May to August in Capital bikeshare) and operators are too busy to inspect/repair them, which further reduce the service quality.

To overcome these drawbacks, a straightforward strategy is to manage bikes to be balanced used. This strategy helps existing bikeshare systems make full use of all bikes. In addition, while bikes are balanced used, we should avoid the situation that all bikes wear out at the same time which overwhelms system operators. We thus propose *batch balancing* which gracefully degrades bikes in batches so that (i) shared bikes are fully utilized and (ii) the usage

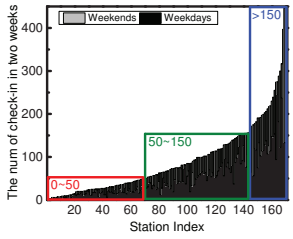

Figure 1: Demand Dist.

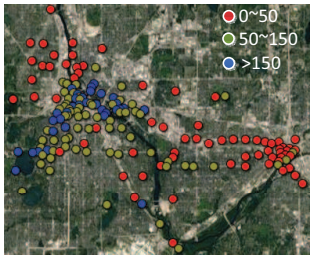

Figure 2: Demand Map

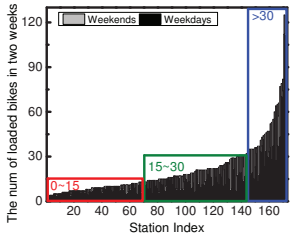

Figure 3: Service Dist.

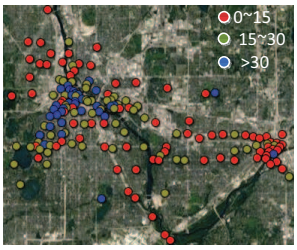

Figure 4: Service Map

status meets operators' demands (e.g., to fit the bike usage status with bike repair/upgrade schedule).

Managing shared bikes to be efficiently used seems straightforward because we can intuitively assign bikes to users based on lengths of their trips when they rent bikes. However, the key challenge is that we can only know their trip lengths when they *return* bikes, not when they *rent* bikes. As a result, naively assigning a newer bike to a passenger may increase usage unbalance because other older bikes may have to be assigned to later passengers with longer trips, and vice versa. Furthermore, it is more challenging to achieve batch balancing which has to meet operators' demands.

Although controlling the usage status is crucial in sharing platforms, most existing work is focused on demand/supply modeling *at station levels* [1, 4, 9, 15] and bike redistribution among stations [16, 17, 25]. To the best of our knowledge, little work, if any, has been conducted to investigate the usage issue *at individual bike levels*. In this paper, we present the first data-driven usage balancing mechanism called eShare to analyze, quantify and (batch) balance the shared usage status at individual bike levels based on large scale data from three bikeshare systems. Specifically, the key contributions of this paper are as follows:

• We conduct the first data analysis on usage balancing for bikeshare systems. To our knowledge, our analysis on bikeshare systems is based on the most comprehensive long-term multi-site data until now, i.e., 140-month multi-event data related to bikes (i.e., rental records, station status, rebalancing and repairs records) for 3 major cities with different system scales and population (i.e., 3.3 million, 6.1 million, and 9.1 million) in both developed and developing countries.

• Built upon the identified features, we design a data-driven usage balancing mechanism, called eShare, which consists of three technical components: (i) a statistical model based on archived data to infer historical usage; (ii) an entropy-based prediction model based on both real-time and archived data to infer future bike usage; (iii) a usage calibration engine based on these two models to select bikes for users to optimize bike usage at individual levels. The calibration engine is the core of achieving efficient usage balancing.

• We develop an ID swapping based measurement methodology and evaluate the performance of eShare through extensive data-driven experiments based on the data from three bikeshare systems including the largest bikeshare system in the world, with 84,000 bikes and 3,300 stations in Hangzhou, as well as the second largest bikeshare system in the US. with 3,700 bikes and 429 stations in Washington D.C. The evaluation results show that the standard deviation of global bike usage is reduced by 90% with our eShare design when the number of batch equals to one.

2 ANALYSIS OF SHARED USAGE

In this section, we first analyze the current usage status of shared bikes in existing systems. Then, we introduce the benefit of usage balancing in efficient sharing.

2.1 Current Bike Sharing Status

By analyzing the usage status in existing bikeshare systems, this subsection shows the need for bike usage balancing. The analysis results from three bikeshare systems in three cities show that inefficient usage is common phenomenon. The basic information about these three systems is shown in Table 1. Details of the data are in Section 3.

Unbalanced Demand: We first introduce the unbalanced demand phenomenon in existing bikeshare systems. Figure 1 plots each station's total number of rentals in two weeks in NiceRide bikeshare system. We find that (i) about 17.6% of the stations (marked as "blue " in Figure 2) are extremely busy which have more than 150 rentals in two weeks; (ii) about 40% of the stations (marked as "red" in

Table 1: Statistic of Three Bikeshare Systems

	Capital	Hangzhou	NiceRide
Launched Time	2010.9	2008.5	2010.6
Num of Stations	429	3.4×10^3	190
Num of Bikes	3,700	8.4×10^4	1,700
Ridership/Day	5.8×10^3	2.3×10^5	1.9×10^3

the demand map in Figure 2) have a low user demand, i.e., fewer than 50 rentals in two weeks. Statistically, 17.6% of the busiest stations satisfy 44.5% of the user demand. In addition, we find that the user demand at some stations varies dramatically on weekdays and weekends, e.g., the user demand at the stations near campus increases on weekdays and drops on weekends.

Unbalanced Service: To meet user demand, the operators in bikeshare systems redistribute bikes among stations. Figure 3 shows the number of bikes loaded to each station in two weeks in NiceRide bikeshare system. We find that the service level at different stations is quite different as in Figure 4: (i) some stations have a high service level with more than 30 loaded bikes (marked as "blue circle"); (ii) however, some stations have a service level with fewer than 15 loaded bikes (marked as "red circle"); (iii) most of the stations with high service levels locate in the downtown and the campus areas.

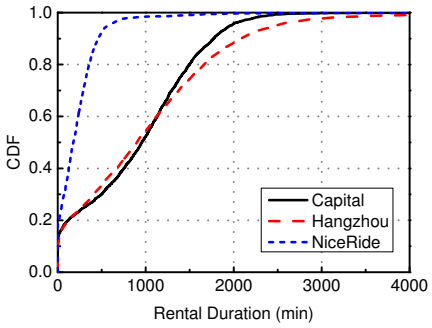

Figure 5: The CDFs of accumulated rental duration

Unbalanced Usage: We investigate the bike usage in Capital, Hangzhou, and NiceRide bikeshare systems. We find that inefficient bike usage is a common phenomenon in existing bikeshare systems. Figure 5 plots the CDF of bikes' rental durations in two weeks in the three bikeshare systems. We find that the bike usage is inefficient in all three systems. For example, in Capital bikeshare system, about 20% of bikes are used more than 1500 minutes while 20% of bikes are used less than 120 minutes in two weeks. As shown in Figure 5, similar results are found in Hangzhou and NiceRide bikeshare systems as well.

2.2 The Benefit of Bike Usage Balancing

From the above discussion, we learn that inefficient bike usage is a common phenomenon in existing bikeshare systems. Bike usage balancing helps existing bikeshare systems improve sharing efficiency. Specifically, with the help of usage balancing, users are able to use those rarely-used new bikes which improves service quality. Inefficient bike usage also causes heavy riding on a portion of bikes which leads to massive damaged bikes. In addition, compared with traditional bikes, shared bikes are designed for sharing purpose and most of them are equipped with special sensors or communication devices. These shared bikes are much more expensive than traditional bikes and need to be batch customized. Bike usage balancing has great potential to help operators achieve efficient system maintenance by letting the system-wide bike usage meet operators' repair/upgrade schedule.

Furthermore, investigating bike usage helps system operators obtain information including but not limited to (i) the current status of a bike, (ii) the residual service time of a bike, (iii) whether a bike needs to be repaired, and (iv) whether bikes are efficiently used. This information can help operators manage bikes and improve the quality of sharing service.

3 ESHARE OVERVIEW

Leveraging upon the share usage analysis results, we aim to develop a novel usage balancing mechanism for efficient sharing. In the following, we first introduce the infrastructure of bikeshare systems and the collected data. Then, we propose design objectives, followed by design overview.

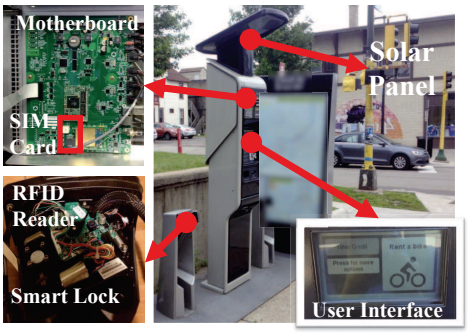

Figure 6: The infrastructure of bikeshare systems

3.1 Bikeshare Infrastructure

Leveraging experiences we gained in our project, we provide our expertise and are part of a team to design, deploy and maintain NiceRide bikeshare system as shown in Figure 6.

From the figure, we can see that passive RFID reader along with smart dock is installed in each dock in a station. When a user comes to borrow a bike, she/he needs to use her/his access key (i.e., RFID tags) to unlock a bike. Once passive RFID reads the tag, the user's information along with current timestamp is sent to central server through the communication device installed in the kiosk (as shown in Figure 6). In our bikeshare system, T-Mobile is adopted to transmit these data. Once user's information is verified by the central server, authorization message is sent back to the station. The smart lock will unlock the bike and the user thus can take the bike.

There is also a RFID tag in the head tube of each bike. When a user finishes her/his trip and returns the bike to a dock, the RFID tag

will be close enough to the passive RFID reader which will recognize the return. The real-time return information is then transmitted to the central server which will compute rental fees based on the rental duration.

3.2 Data Sets

Our infrastructure collects multi-event data as follows.

• **User Trip Data:** Each record includes user ID, borrowed bike ID, trip duration, start/end date and time, and start/end station. These data are used to infer user's habit (e.g., origin/destination and rental time) and bike's usage status.

• **Station Status Data:** Each record includes time, station ID/name, station GPS with available bikes or docks. These data help us obtain real-time station status and decide which bike should be selected for a user.

• **Operators' Rebalancing Data:** A rebalancing record includes operators' ID, bike ID, loading/unloading, station ID, time. Combined with the user trip data, these data help us obtain bikes' IDs in a station, and are also used to infer repaired bikes after operators return them.

• **Bike Repair Data:** Different from the above data sets, bike repair data cannot be extracted from the sensing data directly, but they are essential to our design. We build a new data set by manually recording each repaired bike's ID and its corresponding date from 08/03/15 to 10/05/16 in NiceRide bikeshare system. We use this data set as the ground truth for evaluation purposes. Based on the repaired bike data, we can infer (i) the current status of a bike, (ii) the residual service time of a bike, and (iii) whether a bike needs to be repaired.

The above data are typically collected by most of existing bikeshare systems in order to monitor and improve service quality. As a result, our eShare design is helpful to other bikeshare systems to improve the usage efficiency.

3.3 Design Objectives

Leveraging the multi-event data, we aim to design a data-driven usage balancing mechanism to help operators efficiently maintain the shared bikes. Specifically, eShare is designed to achieve the following two objectives:

• **Better Service Quality:** eShare should avoid excessive wearing on a small portion of bikes and fully exploit those rarely-used bikes to improve the overall service quality. In addition, some bikeshare systems impose a mandatory decommission day for operational and maintenance concerns. For example, the mandatory decommission deadline for sharing bikes is three years in China. In the US, although there is no mandatory decommission deadline, most of bikeshare systems upgrade bikes every two to four years. In these cases, eShare aims at controlling the bike usage, so that all bikes are evenly used before the decommissioning/upgrade day. Such an even usage improves overall service quality for riders.

• **Efficient Maintenance:** While bike usage is controlled to be evenly-used, eShare should avoid the situation that all bikes wear out at the same time which may overwhelm repair staff. The concept of *batch balancing* can gracefully degrade bikes in batches so that the workload of repairs is aligned with the staff capacity. To achieve this batch balancing objective, eShare needs to have the capability to statistically control the number of repairing bikes to meet operators' repair capability. Furthermore, it would be desirable to control the interval between routine repairs, so that bike parts can be purchased in batch and labors can be scheduled efficiently.

3.4 Design Overview

Keeping these design objectives in mind, we establish eShare – a usage balancing mechanism which consists of three modules. Although the description of these modules is specific to the bikeshare system, our approach is generically applicable to other sharing systems (e.g., rideshare).

• A historical usage model: It is a statistical model responsible for quantifying the bike historical usage with archived data.

• A future usage model: It is a predictive model responsible for inferring bike usage associated with future rentals from potential passengers.

• A usage calibration engine: It is based on the historical usage as well as the potential usage of the incoming rental, and chooses a bike for coming users to minimize the system-wide usage variance. The usage calibration engine is the core of achieving batch balancing.

eShare is designed to achieve batch balancing control, e.g., controlling the usage of bikes in the same batch to reach the usage threshold for routine repair. In the following, we introduce these three modules as well as how to achieve batch balancing with these modules in detail.

For the sake of clarity, in Section 4, we first introduce our basic design where we use rental duration to quantify bike usage. Thus, our historical usage model is simply to add all cumulated historical rental durations for the same bike together as its historical usage. We directly introduce the other two modules, i.e., the predictive future usage model and the system-wide usage calibration model in Section 4 to show our key idea. Then, in Section 5, we propose an advanced design to quantify bike usage by incorporating more factors (e.g., origin and destination, terrain features, etc.) in addition to rental duration.

4 BASIC DESIGN

This section introduces future usage prediction model, followed by the system-wide usage calibration model. Then, we introduce how to achieve usage balancing with these models.

4.1 Future Usage Prediction

This section describes the predictive future usage model which estimates coming users' rental duration, thus to obtain potential bike usage. Given both historical and real-time user trip data, we propose a fine-grained Bayesian model to estimate users' rental duration. In existing bikeshare systems, when users swipe their cards to rent a bike, the bikeshare system obtains users' ID, start station, and start time. Based on users' historical rental records, we use this real-time information as a condition to predict users' rental duration.

In the following, we first demonstrate that a user's rental duration between a pair of origin (i.e., start station) and destination (i.e., end station) is stable. Then we show that the entropy of a user's destination is low, especially when the start station and start time are given.

• **Property 1: Stable Rental Duration.** For a user u renting a bike at start station S_u and returning the bike at destination D_u, the rental duration R_u is stable. Moreover, if the hour of the start time t_u is given, the standard deviation of the rental duration, denoted by $\sigma(R_u)$, will be further reduced.

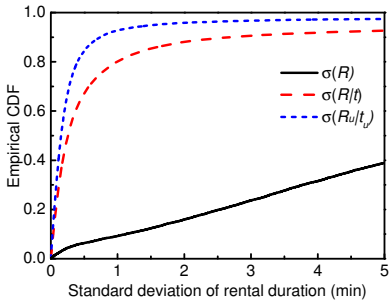

Figure 7: The CDFs of the standard deviation of rental duration with the same origin and destination given start time and user information.

Figure 7 shows the CDFs of (i) $\sigma(R)$, i.e., the standard deviation of rental duration with the same origin and destination, (ii) $\sigma(R|t)$, i.e., the standard deviation of all users' rental duration with the same origin and destination given the hour of the rental start time t, and (iii) $\sigma(R_u|t_u)$, i.e., the standard deviation of user u's rental duration with the same origin and destination pair given user u's start time t_u. The statistical result is based on 58,330 users' 483,239 rental records from 04/01/2015 to 11/01/2015 in NiceRide bikeshare system. From the figure, we can clearly see that the rental duration becomes more stable when the hour of the rental start time and the user information is given. With the start time information, 90% of $\sigma(R|t)$ are lower than three minutes. With both the start time and user information, 90% of $\sigma(R_u|t_u)$ are lower than one minute.

• **Property 2: Low Entropy Destination** The entropy of user u's destination $H(D_u)$ is low. The conditional entropy $H(D_u|S_u, t_u)$ is much lower when the start station S_u and start time t_u are given.

Figure 8 plots user u's CDFs of $H(D_u)$, $H(D_u|S_u)$, and $H(D_u|S_u, t_u)$ based on all her/his trip records during eight months, which are computed as follows,

$$H(D_u) = -\sum_{D_u \in \Psi} p(D_u) \log p(D_u),$$
$$H(D_u|S_u) = \sum_{D_u, S_u \in \Psi} p(S_u, D_u) \log \frac{p(S_u)}{p(S_u, D_u)},$$
$$H(D_u|S_u, t_u) = \sum_{\substack{D_u, S_u \in \Psi, \\ t_u \in \chi}} p(S_u, D_u, t_u) \log \frac{p(S_u, t_u)}{p(S_u, D_u, t_u)},$$

where Ψ is the set of bikeshare stations, and $\chi = \{[00:00:00, 01:00:00), [01:00:00, 02:00:00), \cdots, [23:00:00, 24:00:00)\}$ is the set of hours in a day.

Figure 8 plots the CDFs of (i) $H(D_u)$, i.e., the entropy of user u's destination, (ii) $H(D_u|S_u)$, i.e., the conditional entropy of user u's destination when the start station S_u is given, and (iii) $H(D_u|S_u, t_u)$, i.e., the conditional entropy of user u's destination when both start station S_u and start time t_u are given. The statistical result is based on 43,316 users' rental data who have at least two rental records.

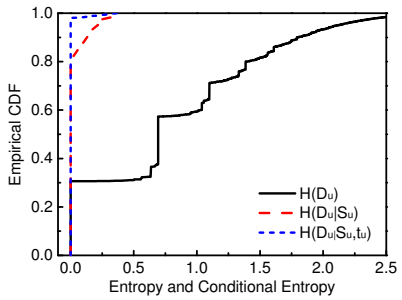

Figure 8: The CDFs of entropy & conditional entropy of destination given start station and time.

15,014 users' rental data are removed in this statistics since these users have only one rental record and thus the entropy of their destination is always zero. As shown in Figure 8, with user's start station and start time information, the values of conditional entropy $H(D_u|S_u)$ and $H(D_u|S_u, t_u)$ are greatly reduced. $H(D_u|S_u, t_u)$ is almost zero, which means user u's destination is almost fixed when his/her start station S_u and start time t_u are given. From the result in Figure 8, we find the destination distribution of a user's coming rental can be accurately predicted given historical user trip records and real-time start station and time.

With the stable duration and low-entropy destination distribution, user u's expected duration $E(R_u)$ is predicted by

$$E(R_u) = \sum_{D_u^i \in \Psi} p(D_u^i) R_u(D_u^i), \quad (1)$$

where $D_u^i \in \Psi$ is user u's possible end station given the start station S_u and start time t_u. $p(D_u^i)$ is the probability that user u will go to end station D_u^i, and $R_u(D_u^i)$ is the rental duration between start station S_u and end station D_u^i.

We use station-wide average rental duration to predict a user's rental duration when the user information is unknown. The predicted rental duration $E(R)$ is given by

$$E(R) = \frac{1}{|U_t^S|} \sum_{u_j \in U_t} E(R_{u_j}), \quad (2)$$

where $E(R_{u_j})$ is computed using Eq.(1), and U_t^S is the set of users who rent bikes in start station S at start time t.

4.2 Usage Calibration Engine

After obtaining the historical bike usage and the usage for the coming rental, this section introduces the system-wide usage calibration engine which is designed to select bikes for users to minimize system-wide usage variance. Although the usage calibration design itself only balances the system-wide usage, it can be used as an engine to achieve batch balancing by introducing weight to historical usage, which will be introduced in Section 4.3. In the following, we first propose the rationale behind our design. Then, we introduce the calibration algorithm.

4.2.1 Design Rationale. When a user u comes to a station to rent a bike, let the available bikes in this station be $\{b_1, b_2, ..., b_m\}$. We calculate the historical usage of these bikes $\{H_{b_1}, H_{b_2}, ..., H_{b_m}\}$ using the usage statistic model (i.e., the rental duration in the basic version of eShare). We then apply future usage prediction model to infer user u's future usage $E(R_u)$. For an arbitrary bike b_i in the station,

if it is selected for user u, then this bike's usage is updated to the sum of its historical usage and future usage, i.e., $R_{b_i} = H_{b_i} + E(R_u)$.

When minimizing the system-wide usage variance, it will be helpful if the usage calibration engine obtains *station-wide* usage statistics. We use an example to illustrate this idea. There are two bikes $\{b_1, b_2\}$ in the station. Bike b_1's historical usage is three hours while bike b_2's historical usage is two hours. When a user u_1 comes to a station to rent a bike, the bikeshare system obtains the user's real-time and historical usage after the user swiped his/her card. The system applies the usage prediction algorithm and obtains u_1's potential usage is one hour. On the other hand, the station-wide average usage is two hour.

Without considering the station-wide statistics, the calibration engine will select bike with minimum usage, i.e., bike b_2, and assign it to user u_1 to minimize system-wide usage variance. Bike b_2's usage is thus updated to $R_{b_2} = H_{b_2} + E_{R_{u_1}} = 2 + 1 = 3$. Since the station-wide average usage is two hour, in statistics, bike b_1's usage is given by $R_{b_1} > 3 + 2 = 5$. The usage variance is thus greater than $(3 - \frac{3+5}{2})^2 + (5 - \frac{3+5}{2})^2 = 2$.

However, if station-wide average usage is considered, the calibration engine will select the bike b_1 for user u_1. The bike usage is updated to $R_{b_1} = 3 + 1 = 4$ and $R_{b_2} > 2 + 2 = 4$, and the usage variance is greater than zero. Comparing these two cases, we find that it is better to consider station-wide usage statistics in bike selection since it helps to achieve smaller usage variance.

4.2.2 Calibration Algorithm. We now introduce how to utilize station-wide usage statistics to let the calibration engine decide the most suitable bike for a user u to minimize the system-wide usage variance. We assume that there are m bikes in the station when user u_1 comes to rent a bike. In the usage calibration design, without losing generality, we consider k potential users, i.e., $u_1, u_2, \ldots, u_k$, where $1 \le k \le m$ and $[u_2, \ldots, u_k]$ are users coming after u_1. Note that the problem formulation with a generic k is helpful for practical settings especially when coming users' reservation information is provided. Let each user u_i's expected usage be $E(R_{u_i})$, where $E(R_{u_1})$ is calculated based on Eq.(1). $E(R_{u_i})$, where $2 \le i \le k$, is estimated based on Eq.(2) when coming users' information is vacant.

Since the variety of bike selection at a station does not change the usage variance of bikes at other stations, the goal of minimizing the system-wide variance is equivalent to

$$\min \sum_{i=1}^{k} \sum_{j=1}^{m} \|(H_{b_j} + E(R_{u_i}))x_{ij} - \overline{R}\|^2, \tag{3}$$

s.t.

$$\begin{aligned} &\sum_{i=1}^{k} x_{ij} \le 1 \quad j = 1, \ldots, m, \\ &\sum_{j=1}^{m} x_{ij} = 1 \quad i = 1, \ldots, k, \\ &\sum_{i=1}^{k} \sum_{j=1}^{m} x_{ij} = k, \qquad x_{ij} = 0 \text{ or } 1, \end{aligned} \tag{4}$$

where $x_{ij} = 1$ if bike b_j for user u_i is selected, 0 if not. In the optimization target, H_{b_j} represents bike b_j's historical usage, $E(R_{u_i})$ represents the expected usage when bike b_j for user u_i is selected, and $H_{b_j} + E(R_{u_i})$ represents the total usage when bike b_j is selected for user u_i. $\overline{R}$ is the system-wide average bike usage and $\overline{R} = \frac{1}{n}(\sum_{k=1}^{n} H_{b_k} + \sum_{i=1}^{k} E_{R_i})$. Then the optimization target represents choosing bikes at station S for coming users to minimize the usage variance. The first set of constraints ensures that no more than one user is assigned to any bike. The second set of constraints ensures that one bike is selected for every user. The third constraint ensures that k of the bikes for all the k users are selected.

We find that it is difficult to solve the nonlinear usage calibration problem with the four constraints directly. In the following, we transform the above nonlinear usage calibration problem to a linear programming problem.

Lemma 4.1. ***(Problem Transformation)*** *The nonlinear usage calibration problem in formula (3)-(4) is equivalent to the linear programming problem with the optimization target* $\min \sum_{i=1}^{k} \sum_{j=1}^{m} \|H_{b_j} + E(R_{u_i}) - \overline{R}\|^2 x_{ij}$ *and the same constraints in formula (4).*

We find that this transformed problem is a variation of the classic assignment problem, named k-cardinality assignment problem, in which there are k rows and m columns, but only k rows are assigned to k columns to minimize the sum of corresponding costs. The k-cardinality assignment problem has optimal solutions that can be achieved by the rectangular assignment algorithm with time complexity $O(k^2 \log k)$.

In the usage calibration process, on the one hand, the system-wide usage variance can be further reduced when we have exact reservation information of more coming users (i.e., a larger k). The estimation of the expected usage of coming users contains prediction errors, which may lead to an inefficient calibration to increase the usage variance. When the estimated user information is adopted, we run the usage calibration algorithm with all possible k, where k is an integer and $k \in [1, m]$, to choose the k which achieves the minimal system-wide variance. By solving the usage calibration problem in formula (3)-(4) with the optimal k, the calibration algorithm chooses the bike which is assigned to u_1.

• **Special Case:** Note that the above usage calibration algorithm includes the special case that the algorithm has no knowledge about the coming users, i.e., $k = 1$ in the usage calibration problem formulation.

Lemma 4.2. ***(Lower Bound)*** *When the algorithm has no knowledge about the coming users, the usage calibration algorithm chooses the bike with minimum usage in the station and reaches the lower bound.*

From this lemma, the performance of the usage calibration algorithm will not be worse than the performance of the algorithm which always selects the bike with minimum usage in a station.

4.3 Batch Balancing

The above usage calibration engine minimizes the system-wide bike usage variance. This engine provides us the capability to achieve batch balancing (e.g., balancing the usage of bikes in the same batch to reach the threshold calling for repair or other maintenance

activities at the same time). With the help of the usage calibration engine, we are able to manage the number of wear-out bikes to meet the maintenance capability by simply adjusting bikes' historical usage to either accelerate or postpone bike usage.

Let the batch of controlled bikes be $\{cb_1, cb_2, \ldots, cb_i, \ldots\}$. We update these bikes' usage with $\alpha \times R(cb_i)$, where $R(cb_i)$ is bike cb_i's real usage. We then apply $\alpha R(cb_i)$ as bike cb_i's usage in the usage calibration algorithm. Recall that the calibration model minimizes the system-wide bike usage variance. When other bikes' average usage is $\overline{R}$, the batch of controlled bikes's real usage is $\frac{\overline{R}}{\alpha}$. When $0 < \alpha < 1$, the usage of the batch of controlled bikes are accelerated. These bikes' usage are postponed when $\alpha > 1$. In real-world control applications in bikeshare systems, when the expected usage after control $\{C(cb_1), C(cb_2), \ldots, C(cb_i), \ldots\}$ is provided, then the value of α for bike cb_i is given by $\alpha(cb_i) = \frac{\sum_{i=1}^{n} C(cb_i)}{n \cdot C(cb_i)}$.

5 ADVANCED DESIGN

In the basic version of eShare, we quantify bike usage using rental duration which accounts for riding time. In this section, we provide an advanced design to quantify the bike usage by exploiting the depreciation caused by each rental trip (accounting for terrain features) from user trip data and bike repair data.

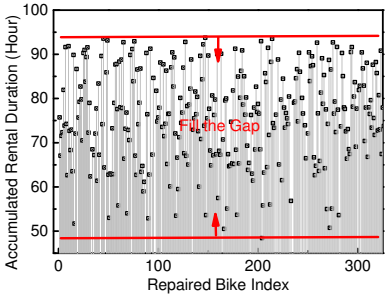

Figure 9: Accumulated Duration of Damaged Bikes

In the basic version of eShare, we use "rental duration" to quantify the bike usage. It works well under the situation that the depreciation caused by rental trips with same rental time are similar or the same. But in some scenarios, even with the same rental time, some rental trips may cause more bike depreciation than others, which leads to a higher repair cost. For example, in Bay Area bikeshare system in San Francisco, the rental trips locate in steep slopes cause more depreciation than those trips locate in flat roads. Under such a scenario, simply using "rental duration" may not accurately decide the status of a bike and the repair frequency. To provide some empirical insights, we use a repair event as an indicator of high usage for a bike. Then, we can explore the rental time distribution of repaired bikes to investigate if accumulated rental duration can be effectively used to indicate usage or not. Figure 9 shows the rental duration of the repaired bikes in NiceRide bikeshare system from 08/03/2015 to 11/01/2015. From this figure, we find that a bike that is repaired due to high usage can have a rental duration as short as 50 hours and as long as 95 hours . Thus, the rental duration alone is not enough to quantify usage.

In the following, we propose an enhanced usage quantification method by leveraging the depreciation caused by each rental trip from the large repaired bike data set. Let the differentiated source-destination pairs in the user trip records be $SD = \{sd_1, sd_2, \ldots, sd_l\}$. For each source-destination pair $sd_j \in SD$, we introduce depreciation rate w_j, where $w_j > 0$. Let the bikes in the repair records be $[rb_1, rb_2, \ldots, rb_N]$. For each repaired bike rb_i, we extract all its rental records from the user trip data within the repair-cycle. For bike rb_i's each differentiated source-destination pair $sd_j^{rb_i}$, we sum up the rental duration under this source-destination pair and obtain $A_j^{rb_i}$, which is accumulated rental duration under trip sd_j. Then, for each repaired bike rb_i, we obtain its adjusted total service duration $\sum_{j=1}^{l} w_j A_j^{rb_i}$.

Since we have a large repaired bike data set, we obtain a large observation of variables about the repaired bikes' total service duration, i.e., $X = [x_1\ x_2\ \ldots\ x_i\ \ldots]^T$, where $x_i = \sum_{j=1}^{l} w_j A_j^{rb_i}$. We assume that the total bike service duration subjects to normal (Gaussian) distribution, i.e., $X \sim N(\mu, \sigma^2)$, where μ is the expected average bike service duration. To reduce the variance of repaired bikes' total service duration, we aim to

$$\min \sum_{i=1}^{N} (\sum_{j=1}^{l} w_j A_j^{rb_i} - X'[i])^2, \tag{5}$$

where $X' = [x'_1\ x'_2\ \ldots\ x'_i\ \ldots]^T$, $X' \sim N(\mu, \sigma'^2)$ and $\sigma' \to 0$.

Lemma 5.1. *The minimum variance of repaired bikes' total service duration will be achieved when*

$$\hat{\mathbf{w}} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{X}'. \tag{6}$$

Eq.(6) provides an optimal solution to the problem in Eq.(5) when the total bike service duration follows normal distribution. From Lemma 5.1, we obtain the depreciation rate w_j for each differentiated trip sd_j. The depreciation rate w_j quantifies each source-destination pair's rental cost and fills the gap among repaired bikes' rental duration. The enhanced usage statistics model thus builds a connection between the usage and rental costs by inferring repaired bikes from the accumulated rental duration. Besides, this depreciation rate provides the operators the basis of a new charging scheme about rental fees with fine-grained depreciation rate.

6 EVALUATION: ID SWAPPING

This section evaluates the performance of eShare with real world data sets from three bikeshare systems. The basic information of these data sets are shown in Table 2. The format of these data sets is introduced in Section 3.2.

6.1 Evaluation Methodology

This section introduces the ID swapping based evaluation methodology which utilizes long-term multi-event data, including user trip, operator rebalancing and bike repair data, to evaluate the performance of eShare. The existing systems' bike usage information, including user ID, bike ID, time and station, can be found directly from the user trip data. Recall that eShare selects a bike from the available bikes in the station for each rent to minimize the system-wide usage variance. We first recover the available bikes (IDs) in a station at anytime. We observe that station status changes if and only if a user borrows/returns a bike or a system operator unloads/loads a bike. In other words, a bike is always in a station in the

Table 2: Basic Information of Collected Data from Three Bikeshare Systems.

Data Set Name	Accessed Periods	Data Size	Number of Records	Recording Frequency
Capital_User_Trip	10/01/10 - 12/31/15	1.4GB	11.8×10^6	Event Driven
Capital_Station_Status	08/04/14 - 12/31/15	1.9GB	269×10^6	Every 1min
Capital_Repair_Bike	01/01/11 - 09/30/15	297KB	3.14×10^3	Event Driven
HangZhou_User_Trip	01/01/13 - 12/31/13	4.2GB	103×10^6	Event Driven
HangZhou_Station_Status	01/01/13 - 12/31/13	1.2GB	248×10^3	Every 30min
HangZhou_Rebalancing	01/01/13 - 12/31/13	109MB	8.5×10^6	Event Driven
NiceRide_User_Trip	06/10/10 - 11/01/15	189MB	1.8×10^6	Event Driven
NiceRide_Rebalancing	04/01/12 - 11/01/15	31.9MB	465×10^3	Event Driven
NiceRide_Repair_Bike	08/03/15 - 10/05/16	272KB	3,724	Event Driven

duration after it is returned/loaded by the previous user/operator and before the next borrow/unload activity comes.

Based on this observation, eShare exploits the available bikes (IDs) for the bike selection of each rent. By utilizing both historical usage and real time card swiping information, eShare selects a bike for each rent in the user trip data. If the selected bike ID is different from the bike ID in the rent record in user trip data, we swap these two bikes' IDs and create a new user trip data with the bike (ID) selected by eShare. In other words, the selected bike by eShare will replace the original bike in the rent record. *Note that such a swap will not change the station status and thus will not change either user demand or service level.* Based on the newly-built user trip data, we are able to do statistics on eShare's system-wide bike usage.

•**Compared Algorithms:** To show the effectiveness of usage balancing in sharing platforms, we compare eShare with (i) the ground truth, which is the bike selection strategy extracted from the user trip data set and (ii) the algorithm which always selects the bike with minimum usage in a station, indicated as "Minimum Usage" in the figures.

•**Performance Metric:** We evaluate the bike usage efficiency from both single-bike and system-wide perspectives:

(i) **Service Timespan:** from single-bike perspective, we investigate each bike's service timespan which is defined as the time duration between the first use of a bike and the last use before a repair.

(ii) **Usage Variance:** from system-wide perspective, we investigate usage variance which is given by $\frac{1}{n}\sum_{i=1}^{n}(R_i - \overline{R})^2$, where n is the total number of bikes in the system, R_i is each bike's usage and $\overline{R}$ is the average bike usage.

The bike usage is defined as the total rental duration in service timespan multiplying by the depreciation rate. In the basic version of eShare, the depreciation rate is set to one.

6.2 Main Performance

In this subsection, we evaluate the performance of eShare by investigating bikes' service timespan and usage variance. Figure 10 shows the each bike's usage in two weeks in Capital bikeshare system with different bike selection strategies. In the figures, the bike usage of ground truth varies dramatically. The bike usage in Capital bikeshare system varies from 0 to 2000 minutes. As a comparison, Figure 10 also shows each bike's usage with eShare and "minimum usage" algorithm. From the figures, we can see that all the bikes in the systems are relatively fully utilized and the usage variance with

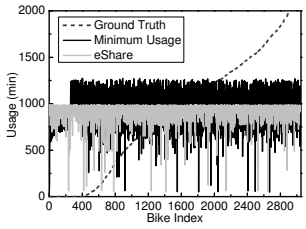

Figure 10: eShare main Performance

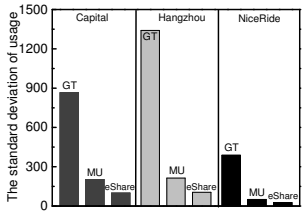

Figure 11: Bike usage Deviation

eShare is significantly reduced. In addition, eShare always achieves better performance than the "minimum usage" algorithm which is a special case of the eShare design. Figure 11 shows the standard deviation of usage with different algorithms in the three bikeshare systems. On average, the standard deviation of usage is reduced by 90% after eShare is applied to existing systems.

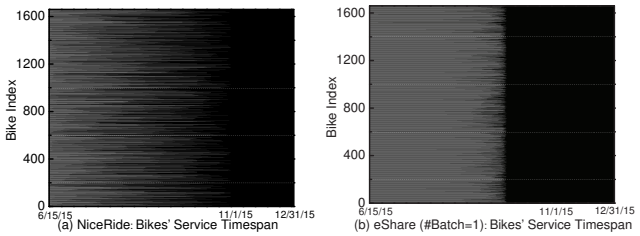

Figure 12: Service Timespan: (a) NiceRide system and (b) eShare

When bike usage is balanced, each bike's service timespan is changed accordingly. Figure 12 plots 1658 bikes' service timespan before (Figure 12(a)) and after (Figure 12(b)) eShare is applied in NiceRide bikeshare system. In the figures, when the color turns to black, it means that the bike's accumulated usage reaches the threshold for routine repair. From Figure 12(b), we find that the end of the service timespan is close when the bikes in the system are balanced used. It is helpful when the end of service time sits in a period when the bikeshare system is idle or in mandatory-decommission/bike-upgrade day.

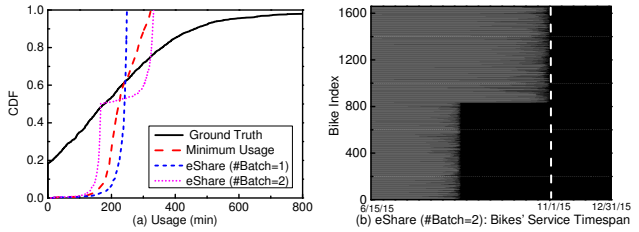

Figure 13: Capability of Batch Balancing:(a) CDFs of Usage and (b) Service Time.

•**Capability of Batch Balancing:** With eShare, we are able to achieve batch balancing. In this experiment, we accelerate the usage of half of the bikes by setting these bikes' usage to half of the real one in bike selection algorithm. Figure 13(a) shows the CDF of two-week bike usage with "Ground Truth", "Minimum Usage", "eShare ($\#Batch = 1$)", and "eShare ($\#Batch = 2$)" in NiceRide bikeshare system. From the figure, we can see that "eShare ($\#Batch = 2$)" put the 18% barely used bikes into use.

On the other hand, there is a usage gap between two portions of bikes. Figure 13(b) shows the corresponding service timespan under the bike usage with "eShare ($\#Batch = 2$)". From the figure, we can see that "eShare ($\#Batch = 2$)" makes portion of bikes' end-of-service-time sit in the idle period (e.g., November) of NiceRide bikeshare system. Figure 13 only shows eShare's capability on batch balancing while a specific control model is required to control the number of repairing bikes to fit with operators' repair capability.

6.3 Design Insight Analysis

In this subsection, we reveal design insights to illustrate why we achieve the above performance.

6.3.1 Performance of Usage Prediction. We perform the evaluation of our method with one-year user trip data from NiceRide bikeshare system which has 483,239 rental records. In this evaluation, we compare the method in eShare with the historical average (HA) method and the "HA" method with start time ("HA w/ ST"). In the "HA" method, we use the average rental duration of a source destination pair's all records to predict a new coming user's rental duration. In the "HA w/ ST" method, we use the average rental duration of a source-destination pair's records with the same hour of the start time.

The metric we adopt to measure the result is error rare which is given by $\frac{1}{N}\sum_{i=1}^{N}\frac{|\widehat{R_i}-R_i|}{R_i}$, where N is the total number of checked rental records, R_i is the ground truth of the rental duration in the i^{th} rental record and $\widehat{R_i}$ is the corresponding prediction value.

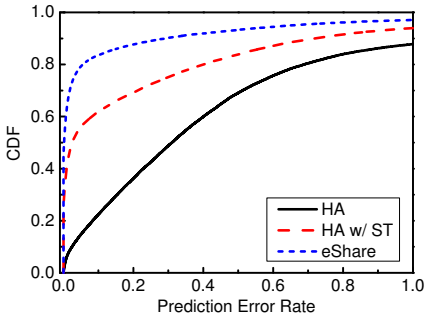

Figure 14: The CDFs of the prediction error rate

Figure 14 plots the CDFs of the prediction error rate of the "HA", "HA w/ ST", and eShare. As we can see, 80% of the predictions with our method have an error rate less than 5.5%. Compared with "HA" and "HA w/ ST", our prediction method significantly reduces the prediction error rate. The rationality behind this improvement is that our method fully exploits fine-grained user rental information from both historical and real-time user trip data in the prediction.

6.3.2 Performance of Usage Statistics. This section evaluates the performance of the enhanced usage model. In NiceRide bikeshare

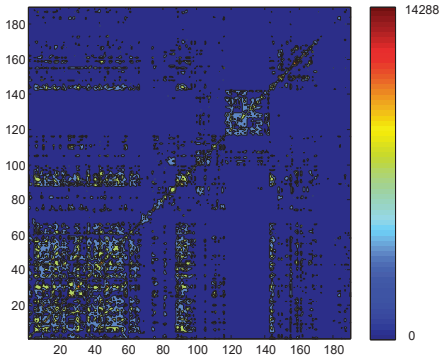

Figure 15: The four-year system-wide trip distribution with all origin-destination pairs in NiceRide bike sharing system.

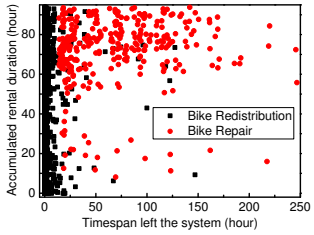

Figure 16: Timespan of Redis. and Repair

system, there are 190 stations and there are 190^2 possible origin-destination pairs. Figure 15 shows four-year system-wide trip distribution with all origin-destination pairs. There are 1,469,882 rental trips in total while the busiest origin-destination has 14,288 trips. In Figure 15, a origin-destination pair has more trips when its color is closer to red. From the figure, we can clearly see that there are many partial red dots in the diagonal. It means that there are a large number of trips which have the same origin and destination. That's the major reason why we do not adopt "mileage" as the measure of usage in this paper. Since the bikes in current bikeshare systems are not equipped with GPS, simply utilizing station distance to measure bike usage will introduce great errors in usage statistics because of the zero distance of the trips with same origin-destination.

We find that a large portion of areas are marked "blue" which means that these origin-destination pairs have almost no traffic. According to our statistics, 3,291 origin-destination pairs have more than one trip in two weeks. That's because that people normally take bike as a short-distance transport tool and only ride bikes to nearby stations.

In the following, we compute the depreciation rate $\hat{\mathbf{w}}$ for these 3,291 origin-destination pairs which requires at least 3,291 bike repair records. We manually record 3,724 bike repair logs with bike id and repair date from 08/03/15 to 10/05/16. Since the usage statistics model will be more accurate when we have more bike repair records, we use the collect repair logs as ground truth and build a model to infer repaired bikes from operator rebalancing data automatically.

For each bike unloading behavior (by operators), we compute this bike's accumulated usage from the user trip data as well as the timespan the bike left the system to classify this unloading behavior is for bike redistribution or for repair. Figure 16 plots the timespan of a bike left the system and the bike's accumulated rental duration when this bike is unload from the system. From the figure, we can see that the bike unloading for repair has long departure timespan and accumulated rental duration.

Table 3: Performance of Classification

Algorithm	TP Rate	FP Rate	Precision	Recall
KNN	0.987	0.137	0.987	0.987
Naive Bayes	0.977	0.346	0.975	0.977
Decision Trees	0.990	0.046	0.991	0.990
RandomForest	0.989	0.065	0.990	0.989
AdaBoost	0.991	0.022	0.992	0.991

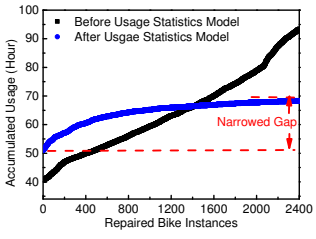

Figure 17: Model Validation

We apply classification algorithms to the data set. Table 3 reports the true positive (TP) rate, false positive (FP) rate, precision, and recall with KNN, Naive Bayes, RandomForest and AdaBoost by 10-fold cross-validation. From the experiment results, we can see that the repaired bike can be accurately predicted from user trip and operator rebalancing data. AdaBoost achieves the best performance since the data set has few outliers which can be found from Figure 16. We apply AdaBoost to four-year user trip and operator rebalancing data where we recover 12,662 bike repair records. Based on these bike repair records, we apply our usage statistics model. Its performance is shown in Figure 17 where we find that the gap of repair bikes' accumulated usage is narrowed. This is because that the usage statistics model leverages the depreciation (caused by terrain features) which reduces the variance of bikes' accumulated usage time.

7 RELATED WORK

The sharing platforms, e.g., Uber [13, 23], Airbnb [12, 20] and Bikeshare, have been on an exponential growth curve over recent years. In sharing platforms, it is crucial to investigate the usage of shared resources to provide an efficient sharing service. To the best of knowledge, most of existing research focuses on participation analysis [13, 23], incentive mechanism and benefits analysis [8, 12, 20] in the sharing economy. Little research, if any, has been conducted to study the fine-grained usage status among shared resources with large-scale data collected from these platforms.

Existing works on bike sharing systems focus on (i) demand analysis [1, 2, 9–11, 14, 15, 24–27], (ii) service analysis [3, 18, 19, 21], and (iii) redistribution scheduling [6, 16, 17]. Compared with previous studies on demand analysis, service analysis, and redistribution scheduling in bikeshare systems, this paper studies a completely new class of problem, i.e., usage balancing for efficient sharing.

8 CONCLUSION

In this paper, we design eShare to achieve usage efficient sharing. We provide a usage statistics model which quantifies (i) each bike's usage status and residual service time and (ii) each rental trip's depreciation rate. These information provides system operator valuable insights on managing bike resources and improving the quality of sharing service. We also propose a fine-grained usage prediction model for a specified user by leveraging his/her historical and real-time bike usage information collected by existing bikeshare systems. In addition, an optimal usage calibration algorithm is proposed to choose the most suitable bike to optimize the system-wide bike usage. The performance of eShare is evaluated through the data sets collected from three representative bikeshare systems. The evaluation results show that our eShare design helps existing bikeshare systems fully utilize shared bikes, improve service quality and reduce maintenance cost.

ACKNOWLEDGMENTS

The work is supported by China National Key R&D Program 2017YFB1003000, US NSF Grants CNS-1446640 and DGIST Research and Development Program funded by MSIP.

REFERENCES

[1] A. Bargar, A. Gupta, S. Gupta, and D. Ma. 2014. Interactive visual analytics for multi-city bikeshare data analysis. In *Proceedings of the 3rd Urbcomp*.

[2] P. Borgnat, C. Robardet, P. Abry, P. Flandrin, J. Rouquier, and N. Tremblay. 2013. A dynamical network view of Lyon's Velo'v shared bicycle system. *Dynamics on and of Complex Networks* 2 (2013), 267–284.

[3] Longbiao Chen, Daqing Zhang, Gang Pan, Xiaojuan Ma, Dingqi Yang, Kostadin Kushlev, Wangsheng Zhang, and Shijian Li. 2015. Bike Sharing Station Placement Leveraging Heterogeneous Urban Open Data. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*. 571–575.

[4] Longbiao Chen, Daqing Zhang, Leye Wang, Dingqi Yang, Xiaojuan Ma, Shijian Li, Zhaohui Wu, Gang Pan, Thi-Mai-Trang Nguyen, and Jérémie Jakubowicz. 2016. Dynamic Cluster-based Over-demand Prediction in Bike Sharing Systems. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*. 841–852.

[5] A. Cohen, D. Simons, M. Martignoni, J. Olson, and C. Holben. 2014. The Bikeshare Planning Guide. *Institute for Transportation and Development Policy (ITDP)* (2014).

[6] G. Dantzig and J. Ramser. 1959. The Truck Dispatching Problem. *Management Science* 6, 1 (1959), 80–91.

[7] P DeMaio. 2009. Bike-sharing: History, Impacts, Models of Provision, and Future. *Journal of Public Transportation* 12, 4 (2009).

[8] T. R. Dillahunt, X. Wang, E. Wheeler, H. F. Cheng, B. Hecht, and H. Zhu. 2017. The Sharing Economy in Computing: A Systematic Literature Review. In *Proceedings of the ACM on Human-Computer Interaction*.

[9] Côme Etienne and Oukhellou Latifa. 2014. Model-Based Count Series Clustering for Bike Sharing System Usage Mining: A Case Study with the VÉLib' System of Paris. *ACM Trans. Intell. Syst. Technol.* 5, 3 (2014), 39:1–39:21.

[10] Jon Froehlich, Joachim Neumann, and Nuria Oliver. 2009. Sensing and Predicting the Pulse of the City Through Shared Bicycling. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*. 1420–1426.

[11] Andreas Kaltenbrunner, Rodrigo Meza, Jens Grivolla, Joan Codina, and Rafael Banchs. 2010. Urban Cycles and Mobility Patterns: Exploring and Predicting Trends in a Bicycle-based Public Transport System. *Pervasive Mob. Comput.* 6, 4 (2010), 455–466.

[12] Qing Ke. 2017. Service Providers of the Sharing Economy: Who Joins and Who Benefits? *Proceedings of the ACM on Human-Computer Interaction* 1 (2017), 57:1–57:17.

[13] Farshad Kooti, Mihajlo Grbovic, Luca Maria Aiello, Nemanja Djuric, Vladan Radosavljevic, and Kristina Lerman. 2017. Analyzing Uber's Ride-sharing Economy. In *Proceedings of the 26th International Conference on World Wide Web Companion (WWW)*.

[14] N. Lathia, S. Ahmed, and L. Capra. 2012. Measuring the impact of opening the London shared bicycle scheme to casual users. *Transportation Research Part C: Emerging Technologies* 22 (2012), 88–102.

[15] Yexin Li, Yu Zheng, Huichu Zhang, and Lei Chen. 2015. Traffic Prediction in a Bike-sharing System. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 33:1–33:10.

[16] Junming Liu, Leilei Sun, Weiwei Chen, and Hui Xiong. 2016. Rebalancing Bike Sharing Systems: A Multi-source Data Smart Optimization. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1005–1014.

[17] Benchimol M., Benchimol P., Chappert B., Taille A. D. L., Laroche F., Meunier F., and Robinet L. 2011. Balancing a dynamic public bike-sharing system. *RAIRO – Operations Research* 45, 1 (2011), 37–61.

[18] Rahul Nair and Elise Miller-Hooks. 2011. Fleet Management for Vehicle Sharing Operations. *Transportation Science* 45, 4 (2011), 524–540.

[19] R. Naira, R. Hampshire E. Miller-Hooks, and A. Busic. 2013. Large-Scale Vehicle Sharing Systems: Analysis of Velib. *International Journal of Sustainable Transportation* 7, 1 (2013), 85–106.

[20] Giovanni Quattrone, Davide Proserpio, Daniele Quercia, Licia Capra, and Mirco Musolesi. 2016. Who Benefits from the "Sharing" Economy of Airbnb. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*.

[21] Tal Raviv, Tzur Michal, and I. Forma. 2013. Static Repositioning in a Bike-Sharing System: Models and Solution Approaches. *EURO Journal on Transportation and Logistics* 2, 3 (2013), 187–229.

[22] Susan A. Shaheen, Elliot W. Martin, Nelson D. Chan, Adam P. Cohen, and Mike Pogodzinski. 2014. Public Bikesharing In North America During a period Of Rapid Expansion: Understanding Business Models, Industry Trends And User Impacts. *Mineta Transportation Institute* (2014).

[23] Jacob Thebault-Spieker, Loren Terveen, and Brent Hecht. 2017. Toward a Geographic Understanding of the Sharing Economy: Systemic Biases in UberX and TaskRabbit. *ACM Trans. Comput.-Hum. Interact.* 24, 3 (2017), 21:1–21:40.

[24] P. Vogel and D. Mattfeld. 2011. Strategic and operational planning of bike-sharing systems by data mining- a case study. In *Proceedings of the 2nd International Conference on Computational Logistics*. 127–141.

[25] S. Wang, T. He, D. Zhang, Y. Shu, Y. Liu, Y. Gu, C. Liu, H. Lee, and S. Son. 2018. BRAVO: Improving the Rebalancing Operation in Bike Sharing with Rebalancing Range Prediction. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*.

[26] Zidong Yang, Ji Hu, Yuanchao Shu, Peng Cheng, Jiming Chen, and Thomas Moscibroda. 2016. Mobility Modeling and Prediction in Bike-Sharing Systems. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. 165–178.

[27] Ji Won Yoon, Fabio Pinelli, and Francesco Calabrese. 2012. Cityride: A Predictive Bike Sharing Journey Advisor. In *Proceedings of the 13th International Conference on Mobile Data Management*. 306–311.