Practical method to improve usage efficiency of bike-sharing systems

Chun-Hee Lee | Jeong-Woo Lee | YungJoon Jung

1 INTRODUCTION

Recently, smart cities have been spotlighted all over the world [1]. Smart cities incorporate a wide range of techniques and technologies covering various domains, for example, architectural engineering, sensor technology, transportation technology, information technology, and communication technology. Note that transportation is a primary consideration when designing and building an effective and efficient smart city.

For effective transportation, each individual component of transport systems (e.g., public buses, public rapid rail, taxis, and private automobiles) must be well connected. To satisfy this goal, a bicycle- or bike-sharing system (BSS) can play a significant role because bicycles can be used to travel to a wide variety of places. Compared with other transportation modes, BSSs have numerous benefits, for example, accessibility, prevention of air pollution, health promotion, and cost effectiveness.

However, despite these tangible benefits, there are many problems to be solved. Even though we identify accessibility as an advantage, bikes in a BSS may be placed in a skewed manner because, in most cases, people want to ride a bike at a similar time and place. For example, in the morning, many people may want to use a bike to travel from their area of residence to their place of work or a bus station. Therefore, bikes should be rebalanced properly by relocation managers. That is, relocation managers should fill up empty stations with bikes and take off bikes from full stations. However, employing relocation managers incurs significant cost; thus, we can...
Many approaches have been proposed to solve the bike rebalancing problem in BSSs [2–9]; however, it is difficult to apply these methods to real environments because many realistic issues must be considered and solved, for example, traffic, parking zones, and personally preferable routing. The best way to solve the bike rebalancing problem is by considering all such issues: However, this is an extremely difficult task. Rather than not providing the complete relocation guide with handling all the issues, we propose a method to assist relocation managers, who are staff members responsible for moving bikes from full stations to empty stations. To generate a realistic relocation plan, we analyzed real-world relocation data and rental/return data. We then formalized a realistic relocation plan based on the results.

In addition, most previous studies on BSSs focused on a large city. In contrast, in this study, we focus on a small city, that is, Sejong, Republic of Korea. Although Sejong is small compared with other cities, there are many virtual bike-sharing stations in the city. It is a virtual space to support a BSS without physical stations. As a result, the amount of available rental and return data at the station-level is sparse. Thus, bike demand prediction and the bike rebalancing problem will be more difficult to handle. To construct accurate models in such a case, we employed multiple classifiers and zero models. Here, each station has its own classifier, and we assign the best classifier using cross-validation. In addition, for stations with little rental and return data, we employ a zero model that returns a zero value all the time because in such cases, complex prediction models can be easily overfitted and exhibit poor performance. Note that we also employed classifiers rather than regressors due to this data sparseness.

Our primary contributions are summarized as follows:

- **Real relocation data analysis**: We analyze the real-world relocation data, which are difficult to obtain, by collaborating with Sejong City, Republic of Korea. In addition, we perform extensive analyses using rental and return data, as well as station monitoring data.
- **Bike demand prediction**: To improve the efficiency of relocation, we attempt to construct a good machine learning model. Here, many techniques are employed to realize this model, for example, multiple classifiers, time grouping, categorization, weather analysis, and station correlation.
- **Practical relocation strategy**: We observe the behaviors of relocation managers using real-world relocation data from the Sejong BSS. On the basis of the results, we provide priority score functions to consider practical situations and propose a relocation strategy to help relocation managers relocate bikes effectively and efficiently.
- **Comparison with real relocation jobs**: We compare the results obtained by the proposed to those of real-world relocation managers. The results of this comparison demonstrate that the proposed method is better than the results of the relocation managers in terms of shortages and moving complexity.

The remainder of this paper is organized as follows. In Section 2, we present background information about BSSs. In Sections 3 and 4, we discuss related work and the architecture of our proposed system. We explain bike demand prediction in detail in Section 5, and the relocation method is described in Section 6. Section 7 presents and discusses experimental results. Finally, the paper is concluded in Section 8.

## 2 | BACKGROUND

The number of users in BSSs is increasing globally. Many users prefer to use BSSs than crowded public transportation due to the COVID-19 pandemic. Compared with other modes of transportation, BSSs demonstrate many benefits, which are described as follows.

- **Accessibility**: Public transportation systems, for example, buses and trains, cannot travel to a wide range of locations and are only available at designated times. In contrast, bicycles can be used to travel to a wider range of locations and are not subject to specific schedules.
- **Traffic**: In urban areas, traffic is a major problem when moving around. Taxis or private automobiles can be used to travel to practically any location; however, bike users do not need to worry about traffic.
- **Air pollution**: Currently, environmental pollution is receiving significant attention, and many government policies consider various environmental pollution problems. In terms of air pollution, bikes are a good transportation mode with zero gas emissions.
- **Health and hobby**: Bike riding represents a healthy and enjoyable transportation alternative that can be used by many different people.
- **Price**: Renting a bike is comparatively less expensive than other transportation systems. For example, using taxi services is very expensive. In addition, local governments in Korea are attempting to introduce and support BSSs; thus, the cost of bike riding is being reduced.
BSSs can generally be categorized into two types of systems, that is, the dedicated docking system and the free-floating system. In the dedicated docking system, users are only required to rent or return bicycles to predesignated locations. In the free-floating system, users can return bicycles to any location. In the dedicated docking system, station installation costs are high; thus, the number of stations in the BSS is limited. In the free-floating system, although users can return bicycles to any location, this can inconvenience pedestrians. To solve these problems, the BSS in Sejong City has introduced a virtual BSS (for new bicycles) as a hybrid approach. In this hybrid system, users can pick up and drop off bicycles at many predesignated stations compared with the dedicated docking system, and BSS managers can easily create and remove virtual stations according to bike demand.

As a special self-governing city, Sejong is a planned city in Korea. Although its population is small (approximately 364000 as of July 2021) compared with Seoul City (the capital of Korea), Sejong City is continually developing, and the ratio of young people (less than 40 years) is very high, that is, 52.4%. In contrast, the nationwide ratio of young people is only 42.7%. The local government in Sejong operates its own BSS called Eouling. The Eouling BSS has two types of bikes, that is, old bikes and new bikes. Here, the old bikes have physical docking stations, and the new bikes have virtual docking stations. As this BSS adopts virtual docking stations, it is easy to add or remove virtual stations; therefore, the number of stations for new bikes is much greater than that for old bikes. The number of stations for new bikes is 559, and that for old bikes is only 63. In addition, the number of new bikes is 2461, and that of old bikes is 334.1 Bike users prefer to ride the new bikes; thus, Sejong local government is investing in new bikes. Therefore, in this paper, we focus on new bikes.

3 RELATED WORK

Currently, various advanced technologies and systems are being integrated into smart cities, for example, BSSs. Many countries are becoming increasingly interested in BSSs for several reasons, for example, environmental problems and traffic congestion. Thus, many studies have investigated BSSs. In the follow, we summarize relevant BSS literature into three main categories.

The first category involves the analysis of bike-related data. To efficiently and effectively develop and manage a BSS, comprehensive analysis of the related data is required [10–14]. For example, Caggiani and others [10–14] used data envelopment analysis (DEA) to evaluate the relative efficiency of stations. DEA is a linear programming-based model to determine the relative efficiency of decision-making units [10,15]. Their results are beneficial in terms of planning and managing BSSs. In addition, Yao and others [13] analyzed the network structures of BSSs. Here, they modeled stations as nodes, movements as edges, and the number of movements as edge weight. Then, they observed the network properties and conducted several analyses, for example, out-degree and in-degree, out-strength and in-strength, radiation distance, and community structure.

Xiaolu Zhou analyzed the spatiotemporal biking patterns in BSSs [14]. To realize an effective spatiotemporal pattern analysis, they employed hierarchical clustering. As a dataset, data from Chicago’s Divvy BSS for the years 2013 and 2014 were used. In addition, Li and others [12] considered effective bike usage representation to handle large-scale BSS data. They used discrete wavelet transform to reduce the dimensionality of the bike usage patterns, and the reduced data were clustered via k-means clustering. The final clustering results might help decision makers obtain valuable insight into BSSs.

Although many studies have analyzed BSSs, they did not focus on real-world relocation data collected by relocation managers. Thus, in this study, we attempted to analyze the real-world relocation data collected by relocation managers and reflected the results in a practical BSS system.

The second category involves construction of a prediction model. Various machine learning-based approaches have been proposed to construct effective bike rental and return prediction models [16–24]. For example, previous studies [19,21] have used long short-term memory (LSTM)-based models because LSTM can capture the long-term dependencies between data. In a previous study [19], two layer LSTM models were used to predict bike demands. The output of these models comprised a combination of rentals and returns. In addition, another study [19] used the Citi Bike System Data as evaluation data. These data cover stations in the Citi Bike system, which operates in New York City and Jersey City. Sardinha and others [21] proposed an LSTM-based model to exploit the properties of historical and prospective contexts. Here, the first layer was for the historical context, and the second layer was for the prospective context. They experimentally demonstrated the validity of their system using Lisbon’s public BSS data.

Many other approaches [18,22,24] have used traditional machine learning models to predict bike demand. For example, in a previous study [24], five prediction models, that is, CUBIST, regularized random forest, classification and regression tree, K-nearest neighbors, and conditional inference tree models, were evaluated to
predict bike demands. Another study evaluated a log–log regression model for bike prediction. In addition, Liu and Pelechrinis [18] adopted a Skellam regression model to estimate excess demand, for example, how many users attempted to rent a bike from an empty station, and Gast and others [25] attempted to model bike availability in a station using a queuing model.

Even though the above approaches attempted to develop good BSS models, and the models are useful for bike rebalancing and BSS planning, these previous studies did not consider the bike rebalancing problem directly. Thus, in this paper, we focus on the bike rebalancing problem to improve the usage efficiency of BSSs.

The last category involves rebalancing bikes properly to improve user convenience. Many previous studies [2–9,26,27] have investigated the bike rebalancing problem. For example, Dell Amico and others [5] solved the bike rebalancing problem using a stochastic programming model, where the problem was formulated as a two-stage problem. In addition, two types of algorithms have been proposed to address this problem, that is, exact algorithms and heuristic algorithms [5]. In another study [8], an operator-based relocation method and user-based relocation method were evaluated. This user-based relocation method rebalances bikes according to BSS users using an incentive. The cost of this method is low; however, the rebalancing task is inefficient. In addition, the operator-based method hires operators, for example, relocation managers. This method is very efficient, but its cost is high. Cheng and others [3] proposed a user-based relocation method using the incentive concept. They proposed a bidding model-based incentive mechanism to optimize incentive prices. Chiariotti and others [4] developed a dynamic rebalancing method by modeling the survival time of each station and formulating the rebalancing problem as an optimization problem in a network. Here, they attempted to maximize user satisfaction while maintaining low rebalancing costs.

These studies have provided effective methods in terms of modeling and mathematical formulation. However, due to various reasons, these methods face limitations in terms of practical implementation in real-world environments, traffic, parking zones, and personalized routing. Thus, in this paper, we focus on a more practical approach.

4 | PROPOSED METHOD

Figure 1 shows the architecture of the proposed method. The proposed method comprises three main components, that is, the data cleansing, bike demand prediction, and bike relocation strategy components. The data cleansing component cleans the bike-relevant data. Prior to cleansing the data, data were collected using two API servers, that is, the Sejong BSS API server and the Meteorological Administration API Server (https://data.kma.go.kr/). Here, we built our own Sejong BSS API server by collaborating with Sejong officials. From the Sejong BSS API server, we downloaded rental/return and station monitoring data. The station monitoring data were recorded to monitor each station every hour, that is, the number of bikes parked at each station every hour. Note that bike demand is strongly related to weather data; thus, we collected weather data from the Meteorological Administration API Server. After obtaining the relevant data, the data were cleaned using the data cleansing component. This process is described in detail in Sections 5 and 6.

After cleaning the data, we constructed machine learning models to predict bike demand using the bike demand prediction component. Here, the goal of bike demand prediction is to relocate bikes in a balanced manner; thus, we attempt to predict station-level bike demand. In addition, each station has a different rental/return pattern and properties. To exploit this point, we collected weather data from the Meteorological Administration API Server. After obtaining the relevant data, the data were cleaned using the data cleansing component. This process is described in detail in Sections 5 and 6.

Finally, based on the bike demand models, the bike relocation strategy is determined using the bike relocation component. The relocation strategy can have the routing information; however, routing involves overly detailed information when performing the relocation task. Here, to realize beneficial routing, we considered many issues, which are described in detail in Section 6.3. Instead, we provide a realistic approach. Here, in summary, we consider the origin-destination (OD) table rather than all of the relocation information. Then, managers use the OD table as a reference.

5 | BIKE DEMAND PREDICTION

To rebalance bikes effectively, we must predict user bike demands precisely. For example, assume some stations
have a small number of bikes. In this case, a relocation manager can drop off bikes at that location as a relocation job. However, if there is no bike demand at those stations in future, the relocation job represented wasted effort. To avoid such situations, we must predict user bike demands accurately. Thus, we attempt to construct an accurate machine learning model. Many studies have investigated bike demand prediction; however, we cannot apply such methods directly to a small city, for example, Sejong City, because the number of bike rentals/returns at the station-level is very small and sparse. To construct an effective bike demand model in such an environment, we first describe how our data are preprocessed in Section 5.1. Then, the model construction methods are described in Sections 5.2 and 5.3.

5.1 | Data preprocessing

The key data used to predict bike demand are the Rental/Return Data in Figure 2A. These data comprise Bike ID, User ID, Rental Date, Rental station ID, Return Date, Return station ID, Travel Time, and Distance. Note that the collected raw data were dirty; thus, we preprocessed the Rental/Return Data as follows:

- Duplicates: The raw data contain duplicates, where all values are the same as those shown in Figure 3A. Therefore, we removed duplicates.
- NULL value: The raw data included many NULL values (Figure 3B). NULL values should be processed properly according to the given situation. Here, some records did not have rental or return stations. In such cases, we removed the records. In addition, some records had a distance value of 0. In these cases, we removed the corresponding records because this indicates that the users did not ride bikes.
- Test Station: The raw data also contained stations that were not used for the BSS service; that is, they were used for testing. For example, in Figure 3C, TEST1 is a test station. Thus, we removed the records generated from test stations.
- Invalid Time: Some records did not contain valid rental/return times. The rental time must be before the return time; however, some records contained rental times that were later than the return time (Figure 3D). Such records were also removed from the data.
- Overlap Time: Realistically, a bike can only be rented after the bike is returned. However, in some cases, the previous bike rental/return time overlapped with the next rental/return time (Figure 3E). In such cases, we remove the former records.

5.2 | Basic model building

To provide the OD relocation table, we must predict station-level bike demand. Here, if we consider only the rental/return data, it will be difficult to construct an effective machine learning model. Therefore, we used an external data source, that is, Sejong weather data. Typically, bike demand is strongly related to weather data. For example, on rainy days, bike demand tends to be low. The goal of bike demand prediction is to rebalance bikes; thus, we predict the number of returns — the number of rentals rather than only the number of rentals. In the basic model construction process, the machine learning models were developed with the following attributes.
Weather Data: We added weather data, for example, temperature, wind direction, wind speed, rain, and humidity.

Holidays: We added holidays. Here, weekends and public holidays were considered holidays in this study.

Date: We added several date-related attributes. Here, we appended the month, season, and day of the week to the data.

As mentioned previously, station-level rental/return data from Sejong City are very sparse. Thus, for a station with limited rental and return data, we employed a zero model that returns a zero value at all times. In such an environment, complex prediction models can be overfitted easily, which will result in low accuracy.

5.3 Advanced model construction and optimization

We employed many techniques to improve the accuracy of the models, for example, demand categorization, time grouping, historical data, and correlation between stations.

- Bike Demand Categorization: In this study, the goal of bike demand prediction is to rebalance bikes rather than obtaining accurate bike demand. Thus, we do not attempt to construct a model whose output is the exact number of bike demands. Instead, we categorized bike demands by $H$, $Z$, $L$, where $H > 2$, $-2 \leq Z \leq 2$, and $L > -2$. Thus, we constructed a machine learning classifier rather than a machine learning regressor.

- Time Grouping: Relocation managers do not do their relocation job every hour. Therefore, we group 24 h into four time periods $A$, $B$, $C$, and $D$, as shown in Figure 4.

- Historical Data: To exploit the time series of data, we added the $n$ previous rental/return data.

- Patterns of Relevant Stations: Some stations may have similar rental/return patterns. Therefore, we evaluated the correlations between stations. Based on the results, we appended rental/return data from similar stations.

In this study, we evaluated many different machine learning models and selected the best model. Here, the following classifier models were used as candidate models.

- OneVsRestClassifier(XGBClassifier ( ))
- OneVsRestClassifier(GradientBoostingClassifier( ))
- OneVsRestClassifier(MLPClassifier(hidden_layer_sizes = (30, 30, 30)))
- OneVsRestClassifier(SVC(kernel = ’rbf’, probability = True))
- OneVsRestClassifier(RandomForestClassifier( ))

Note that our prediction model is a multiclass classifier with three classes rather than a binary classifier. Several approaches are available to extend a binary classifier to a multiclass classifier. Here, we used the one-vs-the-rest approach [28,29] (i.e., OneVsRestClassifier in the Python sklearn package). In this approach, the binary classifier of each label is constructed against the other classes. This approach is conceptually simple and good for extending the receiver operating characteristic (ROC) curve for binary classifiers into that for multiclass classifiers. Note that the ROC curve was designed for binary classifiers.

6 REAL RELOCATION DATA ANALYSIS AND RELOCATION STRATEGY

To consider the real environment, we analyzed the real-world bike relocation data from Sejong City. Prior to discussing out data analyses, we describe data preprocessing in Section 6.1. The analytical results are then described in Section 6.2, and we discuss our relocation strategy in Section 6.3.

6.1 Preprocessing

Figure 2C shows the structure of the relocation data. The relocation data comprise Bike ID, Pick-up station, Pick-up Date, Drop-off station, and Drop-off Date. However, in the BSS in Sejong City, the relocation data are stored separately (Figure 5). Figure 5A shows the complete relocation data, and Figure 5B shows the raw relocation data. The first record in Figure 5A (bike1, station1, 2021-10-20 9:00 AM, station3, 2021-10-20 9:30 AM) is actually stored into two records (record1: bike1, station1, 2021-10-20 9:00 AM, NULL, NULL; record2: bike1, NULL, NULL, station3, 2021-10-20 9:30 AM), as shown in Figure 5B. We guess that this is because the complete record of the relocation data is generated after the relocation manager drops off a bike. Therefore, the complete relocation data must be

<table>
<thead>
<tr>
<th>Time Zone</th>
<th>Drop</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>0:00 – 6:00</td>
<td>O</td>
<td>-</td>
</tr>
<tr>
<td>6:00 – 9:00</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>9:00 – 17:00</td>
<td>-</td>
<td>C</td>
</tr>
<tr>
<td>17:00 – 21:00</td>
<td>-</td>
<td>D</td>
</tr>
<tr>
<td>21:00 – 24:00</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
acquired from the separate raw data. If all data were included in the raw relocation data, acquiring the complete relocation data would be an easy task; however, some data were missing in the raw relocation data. Thus, we used the algorithm shown in Figure 6 to acquire the complete relocation data. Here, we first added two attributes (columns), that is, NewDate and Type, to the raw relocation table. These are set in Lines 2 to 8. For the pick-up record, we set “Type” to “P,” and we set “Type” to “D” for the drop-off record. In addition, “NewDate” is set to the corresponding date. Then, we attempted to form pick-up/drop-off pairs in sequence. However, we had to consider potentially missing records, for example, pick-up1/pick-up2/drop-off2. In such cases, we remove unpaired records, for example, pick-up1 (Line 13). In addition, we removed records whose travel time (“drop-off time”–“pick-up time”) was greater than 1 day (line 15).

<table>
<thead>
<tr>
<th>Bike ID</th>
<th>Pick-up Station</th>
<th>Pick-up Date</th>
<th>Drop-off Station</th>
<th>Drop-off Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>b1k1</td>
<td>station1</td>
<td>2021-10-20 9:00 AM</td>
<td>Station3</td>
<td>2021-10-20 9:30 AM</td>
</tr>
<tr>
<td>b1k2</td>
<td>station1</td>
<td>2021-10-20 9:00 AM</td>
<td>station4</td>
<td>2021-10-20 9:50 AM</td>
</tr>
</tbody>
</table>

6.2 Analysis

First, we observed the moving patterns from the real relocation data from Sejong. According to the time zone (A, B, C, or D) and the day of the week, we visualize the moving patterns for relocation jobs in Figure 7, which shows the moving patterns in the 24th and 25th weeks of 2021. On Sunday, relocation managers perform the relocation task in time zone B. However, we could not find any special moving patterns because, in the real environment, bike demand and relocation processes may be dynamic and complex.

We then analyzed the relocation data with rental/return data from June and July of 2021. Here, we observed the relocation data by separating the pick-up data and drop-off data. Figure 8 shows the top 20 stations with the highest number of pick-up records, and Figure 9 shows the top 20 stations with the highest number of drop-off records. In Figure 8, the second column represents the rank by the number of pick-up records, and the third column represents the rank by the number of return records. Note that the rank begins from 0. Intuitively, as the number of return records increases, the number of pick-up records also increases. As shown in Figure 8, the top 20 stations according to the number of

---

**Algorithm: make_complete_relocation_data()**

**Input:** raw relocation data T (table format)

**Output:** complete relocation data

1: add the columns “NewDate” and “Type” into T
2: for each row of T
3: if pick-up date is not NULL
4: set “NewDate” to pick-up date
5: set “Type” to “P”
6: else
7: set “NewDate” to drop-off date
8: set “Type” to “D”
9: T_group = group T according to bike ids
10: T = a
11: for each group g in T_group
12: sort g by “NewDate”
13: remove the elements of g such that they are not paired (P/D)
14: T = T ∪ g
15: remove elements of T such that their travel interval is more than one day
16: return T

---

**FIGURE 5** Example of relocation data: (A) complete relocation data and (B) raw relocation data

**FIGURE 6** Algorithm used to acquire complete relocation data

**FIGURE 7** Real relocation patterns: (A) 24th week and (B) 25th week
pick-up records have a small return rank. In most cases, the return ranks are less than 100, which means that the number of pick-up records is proportional to the number of return records.

The drop-off data with the number of rental data are shown in Figure 9. Compared with the number of pick-up records, here, the number of the drop-off records is not proportional to the number of rental data. The average return rank in Figure 8 is 44.05, and the average rental rank in Figure 9 is 148.2. We examine this in detail as follows. The rental rank in SJ_00107, SJ_00061, SJ_00367, SJ_00476, SJ_00478, SJ_00072, and SJ_00054 is worse than other stations. Those stations are near residential areas, for example, apartment buildings. We consider that this is because relocation managers consider residential areas more carefully than other areas. For example, if a shortage of bikes occurs in a residential area, residents may call the BSS to communicate complaints. Therefore, relocation managers are likely to relocate bikes proactively in residential areas. Thus, we must exploit this point to rebalance bikes in a practical manner.

In addition, we analyzed the relationship between relocation data and station monitoring data. Here, we evaluated the correlation between these data, and the corresponding scatter plots are shown Figure 10. The pick-up data and station monitoring data exhibit positive correlation in Figure 10A (correlation coefficient: 0.411) because, if there are many bikes parked at a station, the relocation managers can pick up many bikes. As once may expect, the drop-off data and station monitoring data exhibit negative correlation (Figure 10B); however, the correlation coefficient is not strong as –0.127 compared with the previous one. Note that the drop-off data may be affected by many factors, for example, residential areas. We can guess that relocation managers perform the drop-off task more carefully than the pick-up task, and we consider this point in our relocation strategy.

### 6.3 Relocation strategy

Several studies have proposed relocation algorithms that consider routing information. If such methods considered all practically relevant factors, they would be very
effective; however, it is difficult to consider all practical factors. Thus, to develop a practical relocation system, we should consider at least the following factors.

- Traffic conditions: Relocation managers drive their truck, which can load spare bikes, by considering traffic conditions. Even if a route is short and easy, it will be avoided in heavy traffic conditions.
- Road accessibility: Relocation managers consider road accessibility when performing the relocation task. For example, managers may prefer a route with wide roads over a route with narrow roads.
- Parking zone: Relocation managers must park their vehicle in order to drop-off or pick-up bikes. Therefore, parking zones are a major factor when moving between stations. However, it is difficult to know all parking zones because sometimes they park their car in the loosely prohibited area.
- Personally preferable routing: Relocation managers may have their own driving patterns and preferred routing. For example, they may stop at a gas station with less expensive fuel prices.
- Immediate request: Occasionally, users may request immediate pick-up or drop-off. In such cases, relocation managers would have to alter their current route.

It is difficult to make a complete routing plan by considering all of the factors above. Relocation managers would likely not use a routing plan generated by an algorithm that did not consider all such practical factors. Therefore, we attempt to support relocation managers rather than providing a complete relocation plan with routing information. Here, we do not focus on the route itself. Instead, we provide the OD relocation table, which can be used by relocation as a reference when performing the relocation task.

Our relation strategy is shown in Figure 11. The algorithm of Figure 11 uses collect_bikes() and distribute_bikes() in Figures 12 and 13. The collect_bikes() algorithm generates a collection plan using the prediction models described in Section 5. First, the algorithm generates a candidate list from stations with a probabilistically high return demand (i.e., number of returns - number of rentals > 2) in Lines 3 and 4 and stations with a high number of currently parked bikes (Lines 5 and 6). Then, the collection plan is generated by checking the candidates in reverse order (Lines 7–11).

The distribute_bikes() algorithm generates a distribution plan using the prediction models described in Section 5. Similar to collect_bikes(), the algorithm generates a candidate list from the stations with a probabilistically high rental demand (i.e., number of returns - number of rentals < -2) in Lines 3 and 4 and stations with a low number of parked bikes currently in Lines 5 and 6. Next, in contrast to collect_bikes(), the candidate list is sorted in ascending order with the score
function `eval_dist_priority_score()`. Here, `eval_dist_priority_score()` is computed as follows:

\[
\text{score}(S) = \begin{cases} 
\frac{S.\text{num_bikes}}{2}, & \text{if } S \text{ in } AR \\
S.\text{num_bikes}, & \text{otherwise}
\end{cases}
\]

where `S` is a station, `S.\text{num_bikes}` is the number of bikes parked at `S`, and `AR` is a special area, for example, a residential area.

### 6.4 Time complexity of relocation strategy

Here, we describe our analysis of the time complexity of the relocation strategy (Figure 11). Our system comprises bike demand prediction and the relocation process. After constructing the models, we do not need to require a training process to retrain the bike using our relocation strategy, and the testing time of the bike demand prediction model is much less than the training time. Thus, to analyze computational costs, we focus on the relocation strategy. The relocation strategy shown in Figure 11 comprises three main parts, that is, bike collection, bike distribution, and OD creation. The bike collection algorithm (`collect_bikes()`) is described in Figure 12. From Lines 2 to 6, a candidate list is generated at a time complexity of $O(n)$, where $n$ is the number of stations. In Line 7, the algorithm performs a sorting operation whose cost is $O(n \log n)$. In Lines 8 to 11, a collection map is created at a time complexity of $O(n)$. The bike distribution algorithm is shown in Figure 13. Similar to Figure 12, candidate list creation, sorting, and distribution map creation are performed at time costs of $O(n)$, $O(n \log n)$, and $O(n)$, respectively. Finally, we evaluate the OD creation in Lines 3–16 (Figure 11). Note that two loops are present in Lines 3 and 7. Here, the maximum number of entities in each loop is $n$; thus, the time complexity of OD creation is $O(n^2)$.

Therefore, the relocation algorithm has an overall time complexity of $O(n) + O(n \log n) + O(n) + O(n \log n) + O(n) + O(n^2) = O(n^2)$, where $n$ is the number of stations. Generally, $n$ does not take a large value; thus, the relocation algorithm can be executed in nearly real time.

### 7 EXPERIMENTS

In this section, we validate the proposed method experimentally. The experimental settings are described in Section 7.1, and the experimental results are discussed in Section 7.2.

To demonstrate the superior performance of the proposed method, we compare our results to those of real relocation managers in Sejong City, Republic of Korea. Here, we used the dataset shown in Figure 14. For model training, we used both the rental/return data and the weather data. For weather data, we used synoptic weather observation data. To download these data, we used the API (http://apis.data.go.kr/1360000/AsosHourlyInfoService/getWthrDataList) from the Korea Meteorological Administration. In these experiments, we only considered stations with at least one rental record during July 2021. In addition, the Sejong BSS has old and new bikes. In these experiments, we focused on only new bikes because they are used more frequently than old bikes.

We divided the data into two categories, that is, `training data` and `testing data`. The training data were used to train the machine learning models, and the testing data were used to evaluate the performance of the model. Occasionally, in such evaluations, data are classified into `training data`, `validation data`, and `testing data`; however, in this study, we used the two identified categories because we selected the best model via using cross-validation (five folds) with the training data.

For comparison, we used the results of real relocation managers in Sejong. As stated previously, the relocation data are incomplete; thus, the data we preprocessed, where some data may have been removed. For fairness, we set the number of relocations in the proposed method to $N$, that is, the number of relocations in the preprocessed relocation data. In our evaluation, we simulated the rental and return of bikes using the preprocessed rental and return data. Note that the number of parked bikes at each station was initially set using the real station monitoring data at 12:00 a.m. Then, the number of parked bikes was evaluated every hour as follows:

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rental/Return</td>
<td>Training Data</td>
</tr>
<tr>
<td></td>
<td>Test Data (Experim)</td>
</tr>
<tr>
<td>Weather Data</td>
<td>Training Data</td>
</tr>
<tr>
<td></td>
<td>Test Data (Experim)</td>
</tr>
<tr>
<td>Relocation Data</td>
<td>Comparison Data (Experim)</td>
</tr>
<tr>
<td>Station Monitoring Data</td>
<td>Comparison Data (Experim)</td>
</tr>
</tbody>
</table>

FIGURE 14 Dataset description
the previous station monitoring number + (the number of returned bikes – the number of rent bikes) + (the number of drop-off records – the number of pickup records).

Each day at 12:00 a.m., we initialized the number of bikes parked at each station using the real station monitoring data.

7.2 Experimental results

First, we discuss the experimental results obtained using the proposed prediction model. Here, we compared the proposed two models, that is, a naive model and a basic model. These models were constructed using the attributes described in Section 5.2 and the bike demand categorization and time grouping factors discussed in Section 5.3. The naive model employed a logistic regression model, and the basic model employed XGBoost, which has demonstrated good performance in the BSS context [20]. We compared the proposed prediction model with the basic and naive models in terms of accuracy and ROC on test data from 1 August to 15 August 2021. To realize a detailed analysis, we selected the top 20 stations. The top 20 stations and the stations using zero models were determined with the number of positive return demands (i.e., the number of cases [time unit: hour] such that the number of returns - the number of rentals > 0) in Living zones 1 and 2 of Sejong City in July, 2021. Figure 15 shows the average accuracy and average area under the curve (AUC) for the top 20 stations. In terms of accuracy and AUC, the proposed method obtained the highest accuracy (68.42%) and the best AUC value (0.8220). The naive approach exhibited the worst performance in terms of accuracy and AUC. Figures 16 and 17 show the ROC curves for the top four stations and the remaining 16 stations, respectively. Recall that the output of the models involves three classes (L, Z, and H); thus, we show three ROC curves according to each class. As shown in these figures, in some cases, the basic and naive models exhibit better ROC curves than the proposed model; however, the average AUC value of the proposed model is the highest.

We also conduct experiments to evaluate the relocation strategy over a 1-week period (1 August to 7 August 2021; Sunday to Saturday). Here, we set the relocation schedule to time zones A and C on Monday to Saturday and time zone B on Sunday. We performed the relocation task in Living zones 1 and 2 because bike usage in these zones is much higher than that in other zones. In addition, we set the relocation workloads in Zones 1 and 2 at a proportion of 6:4.

Figure 18 shows the experimental results in terms of the number of shortages. The number of shortages is defined by the number of stations where there are no available bikes. On Sunday, the relocation was performed in only time zone B; therefore, at approximately 9:00 a.m., the number of shortages with the proposed method
FIGURE 17  Receiver operating characteristic (ROC) curves for the remaining 16 stations
drops sharply. We found that the proposed method is better than the real approach. In addition, as shown in Figure 18 (from B to F), the number of shortages with the proposed method is better than that of the real approach. We also found that the numbers of shortages at 6:00 a.m. and 5:00 p.m. were lower than at other times because the relocation of our approach is performed then. In the following, we discuss the number of shortages identified at 11:00 p.m. The gap in terms of shortages between the proposed and real methods on Sunday to Saturday was 8, 31, 25, 35, 10, and 5, respectively. Large gaps were observed from Monday to Thursday. On Saturday (Figure 18G), the number of shortages with the proposed method was similar to that of the real results because, on Saturday, the number of relocations was very small compared to the other days, and the relocation job was not performed effectively.

We also evaluated the proposed method in terms of moving complexity. Although the number of bikes to be relocated is the same, the difficulty of the relocation tasks may differ. For example, if we move all bikes from one location to another simultaneously, the job difficulty will be low. Here, we define moving complexity by (moving volume/total number of bikes to be relocated). As the moving complexity increases, the difficulty of the job will also increase. If the moving complexity is 1, the job difficulty is the highest. On the average, the proposed approach shows about 5% lower moving complexity than the real approach, as shown in Figure 19.

In summary, we have demonstrated that the number of shortages can be reduced by the proposed method compared to the real method, and the relocation plan generated by the proposed method is easier than that generated by real relocation managers in terms of task difficulty (i.e., moving complexity).

8 | CONCLUSION

In this paper, we have proposed a practical method to solve the bike rebalancing problem, with a focus on the Sejong BSS. Sejong, Republic of Korea is a small city; thus, predicting bike demand and forming a relocation plan are not easy tasks. First, we attempted to construct an effective station-level model using various techniques. To improve model accuracy, we investigated various machine learning models and selected the best model. In addition, to exploit real relocation patterns, we analyzed real relocation data from Sejong and then we developed an effective relocation strategy. Finally, we have demonstrated that the proposed method outperformed real relocation managers in terms of both bike shortages and task difficulty.

ACKNOWLEDGEMENTS

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2018-0-00225, Development of City Interior Digital Twin Technology to establish Scientific Policy).

CONFLICT OF INTEREST

The authors declare that there are no conflicts of interest.
REFERENCES


AUTHOR BIOGRAPHIES

Chun-Hee Lee is a senior researcher at the Electronics Telecommunications Research Institute. He received a PhD degree in computer science from the Korea Advanced Institute of Science and Technology in 2010. His research
interests include large-scale data processing, modeling and simulation, and knowledge graph completion.

**Jeong-Woo Lee** received a BS degree in information and telecommunications engineering from Jeonbuk National University, Jeonju, Republic of Korea, in 1996, and MS and PhD degrees in information and communications engineering from Gwangju Institute of Science and Technology, Gwangju, Republic of Korea, in 1997 and 2003, respectively. From 2003 to 2005, he worked at the LG Electronics Institute of Technology, Seoul, Republic of Korea. Since 2005, he has been with Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea, where he is now a Principal Member of the Engineering Staff. His primary research interests include digital video coding algorithms, GPU-based coding algorithms, large-scale data analysis and simulation, and ML-based estimation algorithms.

**YungJoon Jung** received a BS degree in physics from Hankuk University of Foreign Studies, Korea, in 1997, a MS degree in computer science from the same university in 1999, and a PhD degree in computer science from Chungnam National University in 2016. Since 2001, he has been with ETRI, Korea, as a Principal Researcher. His research interests include embedded operating system, real time distributed computing, power management systems, digital twin data analytics, and software simulation.

**How to cite this article:** C.-H. Lee, J.-W. Lee, and Y. Jung, *Practical method to improve usage efficiency of bike-sharing systems*, ETRI Journal 44 (2022), 244–259. [https://doi.org/10.4218/etrij.2021-0408](https://doi.org/10.4218/etrij.2021-0408)