

Research Article

Recurrent Neural-Based Vehicle Demand Forecasting and Relocation Optimization for Car-Sharing System: A Real Use Case in Thailand

Peerapon Vateekul ¹, Panyawut Sri-iesaranusorn ^{1,2}, Pawit Aiemvaravutigul ¹,
Adsadawut Chanakitkarnchok ¹ and Kultida Rojviboonchai ¹

¹Chulalongkorn University Big Data Analytics and IoT Center (CUBIC), Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand

²Division of Information Science, Nara Institute of Science and Technology, Nara, Japan

Correspondence should be addressed to Kultida Rojviboonchai; kultida.r@chula.ac.th

Received 26 June 2020; Revised 3 January 2021; Accepted 12 January 2021; Published 28 January 2021

Academic Editor: Dongfang Ma

Copyright © 2021 Peerapon Vateekul et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A car-sharing system has been playing an important role as an alternative transport mode in order to avoid traffic congestion and pollution due to a quick growth of usage of private cars. In this paper, we propose a novel vehicle relocation system with a major improvement in threefolds: (i) data preprocessing, (ii) demand forecasting, and (iii) relocation optimization. The data preprocessing is presented in order to automatically remove fake demands caused by search failures and application errors. Then, the real demand is forecasted using a deep learning approach, Bidirectional Gated Recurrent Unit. Finally, the Minimum Cost Maximum Flow algorithm is deployed to maximize forecasted demands, while minimizing the amount of relocations. Furthermore, the system is deployed in the real use case, entitled “CU Toyota Ha.mo,” which is a car-sharing system in Chulalongkorn University. It is based on a web application along with rule-based notification via Line. The experiment was conducted based on the real vehicle usage data in 2019. By comparing in real environment in November of 2019, the results show that our model even outperforms the manual relocation by experienced staff. It achieved a 3% opportunity loss reduction and 3% less relocation trips, reducing human effort by 17 man-hours/week.

1. Introduction

With the growth in population and economy, a car-sharing system becomes an alternative public transportation that alleviates road traffic congestion [1]. Experts in the transportation field predicted that car sharing can significantly increase in the next ten years, especially in Asia-Pacific [2], and can become a possible bridging mode between private cars and traditional public transportation such as bus and train. Furthermore, the number of car-sharing users is predicted to reach 36 million accounts, and the value of the global car-sharing market is forecast to be 11 billion US dollars in 2024 [3].

From the past to the present, there have been two main design choices for a car-sharing system. The first design is a round-trip car-sharing system. In this design, a customer needs to reserve a vehicle in advance and return the vehicle at the station they picked. The other design is a one-way car-sharing system. Multiple vehicles are available for customers at multiple stations. A customer can pick a vehicle from a station and return the vehicle at any station in the operation area. If the one-way car-sharing system operates efficiently, it can be more beneficial for both customers and operators, compared to the round-trip car-sharing system [4]. However, in the one-way car-sharing system, the number of vehicles at each station can become imbalanced in the sense

that too many vehicles are available at the low-demand stations while the high-demand stations contain very few vehicles [5]. Such a situation leads to opportunity loss for the car-sharing system. In order to reduce the opportunity loss, an effective vehicle relocation strategy needs to be considered.

Several strategic solutions have been proposed for vehicle relocation in the one-way car-sharing system, which can be classified into two categories. The first category is the user-based strategy, in which customers/users are incentivized by marketing campaigns and then help relocating the vehicles [6–11]. This strategy easily scales with the number of users and does not need efforts from the vehicle operators. Nevertheless, it is not guaranteed that the users will always follow the vehicle relocation plan as expected. The second category is the operator-based strategy, in which the vehicle operator forecasts the demands at each station and then sends the operation staff to relocate the vehicles according to the demand forecasting results. This strategy ensures that the vehicle relocation tasks are done as needed but requires highly experienced staff to work for the vehicle relocation [8, 12–18]. In [12, 15], solutions based on the mathematical models and rule-based algorithms were proposed. In [13], the battery consumption and recharging issues were considered in order to optimize the relocation operation. Recently, in [14, 19], the deep neural network for demand management in a car-sharing system is introduced. Although all of the aforementioned solutions show improvements in relocation operations based on the realistic data, none of them has been implemented in the real-world car-sharing system.

In this paper, we propose a novel vehicle relocation system with major improvements in data preprocessing, demand forecasting, and relocation optimization. Our proposed system is designed for one-way car-sharing system with an operator-based strategy. Firstly, the data preprocessing automatically removes fake demands caused by search failures and application errors. Then, the real vehicle demand at each station is forecasted using a deep learning approach called Bidirectional GRU. We have described the reason for using BiGRU in Section 4.2. Finally, the Minimum Cost Maximum Flow algorithm is deployed to maximize the vehicle availability according to the forecasted demands, while minimizing the amount of relocations. Furthermore, our proposed system is deployed in the real use case entitled, “CU Toyota Ha:mo,” which is a car-sharing system in Chulalongkorn University, Bangkok, Thailand. Monitoring dashboard and relocation notification for the operation staff are also implemented in our system. In the experiment, the model was trained using the usage log in 2019. The results show that the relocation plan prepared by our model is better than the plan managed by human experts. It can reduce human effort to a great extent. Furthermore, the system has been deployed in the real-world scenario. The results show that our system can successfully operate and replace the existing human process at the operation room.

In summary, our contributions compared to prior attempts are as follows: The main goal of our work is to propose the whole vehicle relation system in the real-world

situation. To provide a suitable relocation plan, it is crucial to have vehicle demands in the future. Thus, ours is the first research work that contains both vehicle demand forecasting and relocation plan optimization in the real-world situation. However, all prior attempts [9, 10, 12–15, 17] are just based on a simulated environment, and none of them proposed both modules as a complete solution. In demand forecasting, we are one of the pioneers who have applied and compared various kinds of deep learning networks. Furthermore, the new loss function has been proposed to take both demand from departure and destination stations into consideration. Also, the vehicle assignment procedure is proposed as a post-processing step of our demand forecasting model. From an intensive experiment, our results show that Bi-GRU outperformed the previous approach, LSTM [14, 19]. For relocation optimization, the optimization approach called “Minimum Cost Maximum Flow” is applied in order to achieve the best relocation plan, while others [11, 16] are based on a simple rule on a simulated environment.

The remainder of this paper is organized as follows: Section 2 discusses the related works. Section 3 explains the overall system and the existing operation of CU Toyota Ha:mo. Our proposed vehicle relocation system including data preprocessing, demand forecasting, and relocation optimization are described in Section 4. The performance evaluation is discussed in Section 5. The conclusion is elaborated in Section 6.

2. Related Works

Vehicle relocation strategies can be classified into two main categories: the user-based strategy and the operator-based strategy [8]. The user-based strategy is based on the users who are available or comfortable to help perform the required relocation tasks. Basically, some incentives such as discounts or free trips are offered to the users. Febbraro et al. [6] proposed a discount for users who volunteer to perform vehicle relocation. Barth et al. [7] introduced two user-based relocation methods called trip joining and trip splitting. When the system realizes that it is becoming imbalanced, it will split users that have more than one passenger to relocate multiple vehicles to the destination station. Conversely, if two users are taking the exact same route at the same time, the system will request the two users to merge into one usage. In addition, Cepolina and Farina [9] investigated a fully user-based vehicle relocation for a car-sharing system in the urban areas. In order to deal with the unbalanced demand, their proposed system notifies the users who are interested in moving the vehicles to the target parking lots. This is not only good for the system management but also benefits the users. Recently, Angelopoulos et al. [10] introduced the user-based strategy for e-motorbike sharing. The idea behind is to combine the public transportation information and graph theory to grant discounts to users. Furthermore, to evaluate the idea, they applied their strategy for a free-floating system in the city of Vitoria-Gasteiz, Spain. These previous researches show that there are various solutions for the user-based strategy. In this paper, our scenario is based on a car-sharing system in a campus of the university. So, the operator-based strategy is simpler and more efficient since it is suitable for a small area (campus).

Contrary to the user-based strategy, the operator-based strategy is based on vehicle relocation performed by the operation staff. Herbawi et al. [12] identified the vehicle relocation as an NP-hard optimization problem and proposed an evolutionary algorithm for a car-sharing system. Their algorithm was evaluated on real-world data published by a car rental and sharing company, “car2go.” They also investigated the system adaptation on the different parameters, such as the maximum allowed duration. Gambella et al. [13] considered the vehicle relocation from the viewpoint of the service provider. The objective is to maximize the profit associated with the trips performed by the users. They introduced a mathematical model for managing staff assignments in a large-scale, car-sharing system by considering battery consumption and recharging processes. Their solutions were developed and tested on a set of realistic data derived from an existing car-sharing system. According to their result, solving the relocation in a car-sharing system is worth and makes the service provider achieve larger profits. By considering the operation staff’s effort and cost, Kek et al. [15] presented a decision support system to determine a set of near-optimal manpower and operating parameters for the vehicle relocation problem. To evaluate and test their system, a simulation was conducted using a data set of commercially operational data from a car-sharing system in Singapore. Regarding the electric vehicle (EV), in [13, 16], the proper solution for EV is examined because the constraints such as charging stations need to be considered. Lu et al. [17] introduced the solution based on matching between the user request and the remaining quantity of electric of EV on simulation. Martin et al. [18] applied the Markov model, a stochastic model which models temporal or sequence data, for relocation tasks in one-way. With the increasing popularity of deep learning, Qu et al. [20] used a simple feed-forward model with historical traffic flow data and contextual factor data for daily long-term traffic flow forecasting. Yu et al. [14] introduced an analytical system that deployed Long Short-Term Memory (LSTM) for forecasting short-term vehicle demand. They also used multiple temporal features, including the time of day, day of week, and weather condition. For evaluation, they used real-world data from a car-sharing system in Chengdu. Then, they demonstrated the reaction of the car-sharing market toward the operating strategies based on the data from two service providers. Ning et al. [19] also used LSTM for demand forecasting, and confirmed the simulation results using a statistical hypothesis test called “Granger causality”. Zhaowei et al. [21] used M-B-LSTM, an end-to-end hybrid deep learning model, for short-term traffic flow forecasting. Ma et al. [22] applied a clustering technique to create many sub-models based on deep neural networks. Then, they [23] proposed the contextual convolutional recurrent neural model to extract inter-and-intra-day traffic patterns for daily traffic flow forecasting. Motivated by the results of these researches, we investigated using a variety of recurrent neural network architecture for a similar purpose. In our scenario, the optimization algorithm must be computationally efficient since the relocation schedule should be run often

(hourly) due to a limited number of sharing cars; therefore, the evolutionary algorithm is not suitable for our scenario because of its high computation cost. Also, more variations of deep learning algorithms, for example, GRU and Bidirectional GRU, are investigated in our work apart from LSTM. We not only just employ those networks but also design our model specifically for the car-sharing system.

Even though the previous studies on the user-based and the operator-based strategies have shown outstanding performance, they were mostly evaluated by feeding real-world data to their solution offline, while results from the actual deployment were not reported. This inspires us to evaluate our solution not only by using realistic data but also by deploying it in the car-sharing system operating in the real world. Furthermore, most prior works focused on traffic forecasting in an open environment, which allows the unlimited number of vehicles in open areas. It is quite different from our vehicle relocation, in which the number of stations and vehicles is fixed.

3. CU Toyota Ha:mo System

Since 2017, Chulalongkorn University and Toyota have collaboratively launched a car-sharing system entitled “CU Toyota Ha:mo” (henceforth referred to as Hamo), as shown in Figure 1. With Hamo’s system, university personnel, such as students, teachers, and staff, can go anywhere in the campus quickly. This system provides them an environment-friendly alternative mode of transportation that complements a public transport within the university. The system performed well in its early stages, but as it is, it cannot avoid opportunity loss problems when it has more users. To deal with this issue, Hamo’s staff employ a manual relocation-based solution by approximating the number of vehicles at each station by their own experience. However, this solution is time-consuming and expensive, while also requiring staff’s experience to be effective.

This section is organized as follows: Section 3.1 describes the dataset and the data collection process, as well as distinguishes between successful and failed usages. Section 3.2 explains Hamo’s existing operation and the problems it currently faces.

3.1. Dataset. Our dataset contains the history of shared vehicle reservations and usages in Hamo’s service. These data were collected through Hamo’s mobile application, where users are required to make reservations before picking up and using the vehicles. As shown in Figure 2, each entry contains information on the user’s trip, for example, the departure station, the destination station, and the reservation duration. These reservation records can be used to predict the demand at a given time in the future. According to the data, we define a *successful usage* as a feasible reservation for which a complete trip can be provided, and a *failed usage* as an unsuccessful reservation where the desired trip cannot be fulfilled. The failed usages can occur due to lack of vehicles at the departure station and/or unavailability of parking lots at the destination station, leading to opportunity losses in the system.



FIGURE 1: An example of a vehicle in CU Toyota Ha:mo system (a) being operated by a user and (b) being parked at a charging station.

UserID	Date	Reservation time	Departure station	Destination station	End usage time	
0	ID#60431	2017-12-01	07:18:06	01 exit cham square	07 chamchuri 5	07:44:52
1	ID#95550	2017-12-01	07:20:35	12 CU terrace	04 engineering	07:44:26
2	ID#17356	2017-12-01	08:14:03	12 CU terrace	04 engineering	08:23:57
3	ID#52570	2017-12-01	08:40:04	03 sala prakeaw	02 economics	08:46:27
4	ID#52570	2017-12-01	08:47:49	02 economics	03 sala prakeaw	08:53:13
5	ID#76189	2017-12-01	09:14:27	12 CU terrace	03 sala prakeaw	09:23:33

(a)

UserID	Date	Reservation time	Departure station	Destination station	Status	
0	ID#90382	2017-12-01	07:06:19	01 exit to cham square	05 arts	1
1	ID#85616	2017-12-01	07:14:40	06 chamchuri 9	06 chamchuri 9	2
2	ID#85616	2017-12-01	07:14:47	06 chamchuri 9	03 sala prakeaw	2
3	ID#85616	2017-12-01	07:15:08	01 exit to cham square	01 exit to cham square	1
4	ID#51386	2017-12-01	07:16:35	01 exit to cham square	01 exit to cham square	1
5	ID#51386	2017-12-01	07:16:36	01 exit to cham square	01 exit to cham square	1

(b)

FIGURE 2: The log of Hamo's existing system. (a) The log of successful usages and (b) the log of failed usages (Status 1 means no available vehicle at the departure station and Status 2 means no available parking lot at the destination station).

To provide Hamo's service for customers, there are 22 stations and 30 vehicles available from 7 a.m. to 7 p.m. on weekdays. The service is closed on weekends and national holidays. Figure 3 shows the service area in the university campus. In our study, we used the data recorded from December of 2017 to October of 2019. According to the data, user behaviors can be analyzed. For example, it can be seen that a lot of usage records were created 30 minutes before 9 a.m., which was the starting time of the first lecture. Moreover, it can be seen that only a few usage records were made during the ongoing time of the lecture.

Figure 4 shows the number of failed usages separated by stations in November of 2018. As can be seen, the top three stations that have the highest number of failed usages are stations 4, 12, and 2, respectively. This is because these stations are located near the shopping mall and/or the faculty with a high number of students. On the other hand, stations 13, 17, and 20 are the top three stations that have the lowest number of failed usages. This is because these stations are located near the faculty with a low number of students and/or difficult to arrive on foot. According to these results, we can know which stations have a high demand, and which

stations have a low demand. Thus, at a given time, we can design the vehicle relocation based on the demand in each station. In our design, such information will be considered together with other important factors such as distances between two stations and staff's workloads.

3.2. Hamo's Existing Operation. Traditionally, Hamo's staff have resolved the opportunity-loss problem by performing the following statistical analysis. First, the operation staff divide the operating time into multiple time slots and then manually assess the demands at each station in each time slot. The ranking of the stations with the highest demand to the lowest demand is presented in a table. Next, vehicles will be assigned to each station ordered by its forecasted demand, so the station with higher demand will be assigned first. Also, we are going to leave at least one parking lot at each station for an incoming vehicle. In contrast, the stations with low demand will be assigned a few vehicles or none. Lastly, based on the table, the operation staff will relocate the appropriate amount of vehicles every two hours.

We investigated that in the data there exist a lot of *fake demands*. This occurred in the data records due to user/system

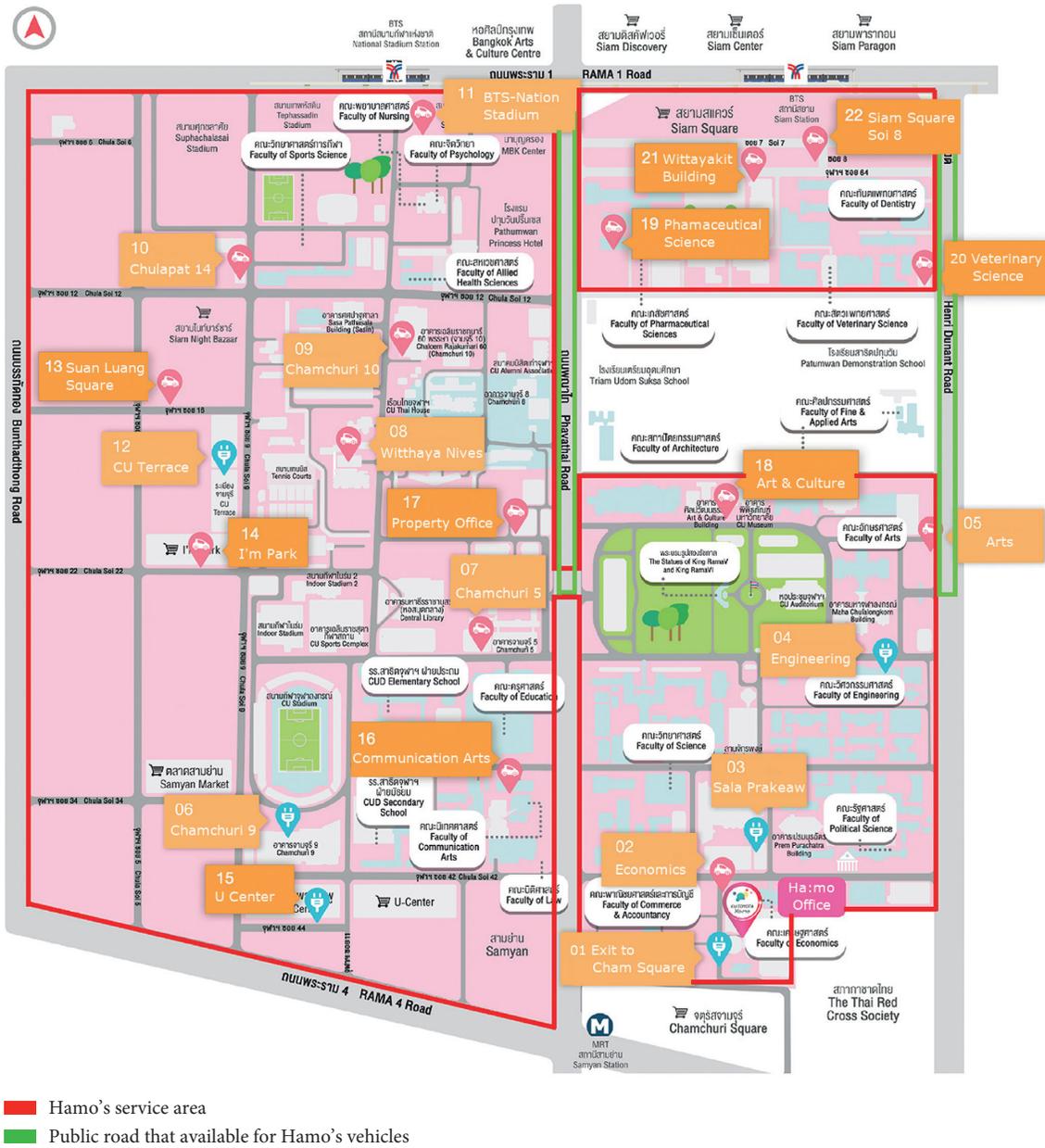


FIGURE 3: Service area in the university campus (source: CU Toyota Ha:momo website available at https://www.cutoyotahamo.com/en/graphic_map-2/).

errors. For example, some users repeat their reservation several times after their first attempt fails. This causes the number of reservation records no longer reflects the number of the actual usage. Another example is when the system creates multiple reservation records from users while they are driving the vehicles. This is a system error and should not happen in the system. However, since these fake demands actually appear in the data, the operation staff manually remove them one by one. This consumes a lot of time and effort of the staff.

4. Our Proposed System

The relocation system has been proposed as illustrated in Figure 5 and includes four modules: (i) automatic data

preprocessing, (ii) vehicle demand forecasting, (iii) relocation optimization, and (iv) web application. Given the history of reservations of the service, the task of our system is to automatically determine optimal vehicle relocation trips for the staff. First, fake demands are removed with an automatic data preprocessing algorithm according to the staff's experience. Second, for each hour in the operating time, we normalize the departure demand fraction and the destination demand fraction, so that the summation of all stations is one at each time step. In our data set, the interval of data is hourly. We aim to forecast one day ahead, so there are 12 forecasted time steps representing 12 operating hours/day. The input window is 30 days prior to the forecasted day (360 time steps), and the output is one time step ahead with the rolling strategy to

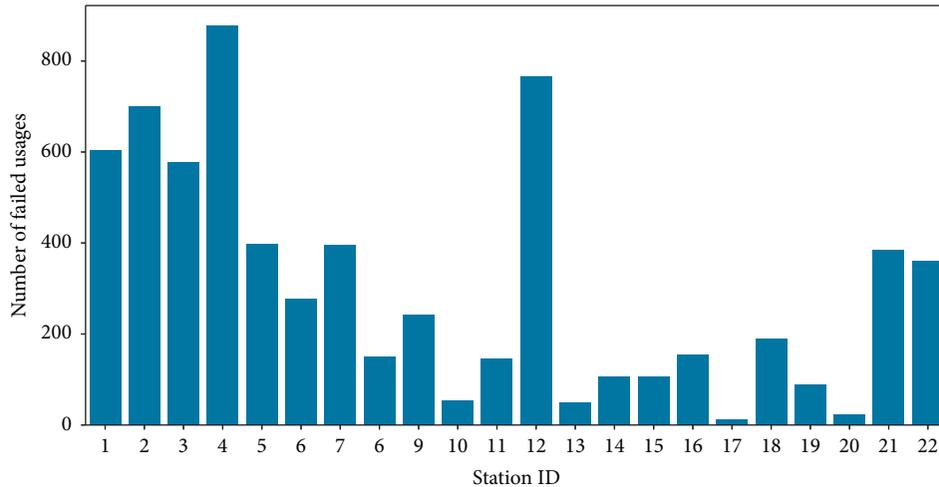


FIGURE 4: Failed usages at each station in November of 2018.

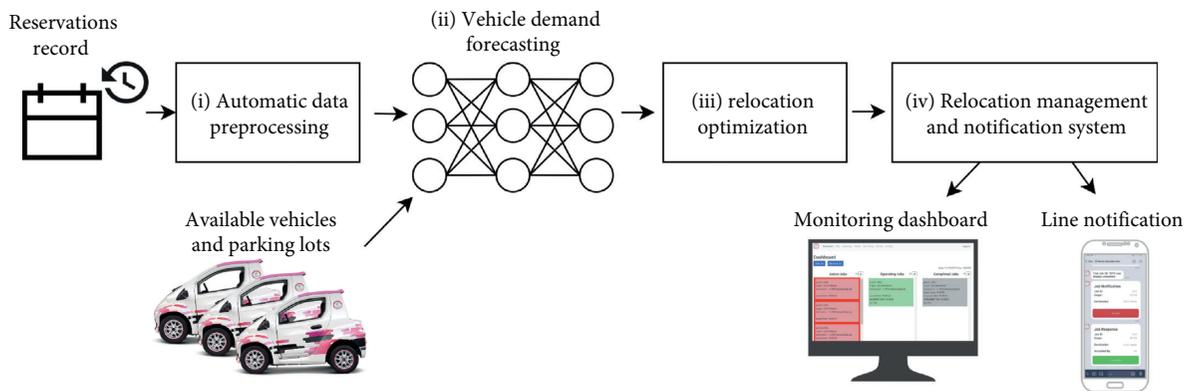


FIGURE 5: Overview of our proposed relocation system with four modules: (i) automatic data preprocessing, (ii) vehicle demand forecasting, (iii) relocation optimization, and (iv) web application.

complete the whole time steps of the forecasted day. The model's architecture is based on a recurrent neural network, which is suitable for dealing with time series data. The rankings are then transformed into suitable amounts of vehicles that each station should have. Third, the Minimum Cost Maximum Flow algorithm is applied on the existing and suitable amount of vehicles at each station, to find the optimal vehicle relocation trips for the staff. Finally, the relocation plans are deployed in our web application and they are automatically notified to all staff.

4.1. Automatic Data Preprocessing. To reduce the staff's workload on manual preprocessing, we implemented the following automatic data preprocessing based on the procedures described by the staff in order to remove fake demands. Our preprocessing is shown in Figure 6. First, we took the reservation data, a mix of real and fake demand, as the input. Then, based on actual usage records, we considered each user's extra reservations during their usage period to be fake demand, and discarded them.

There are two main incorrect demands: (i) abnormal search logs after successful booking and (ii) repeated

reservation attempts. The abnormal search logs can be explained by an example; in the case of a successful usage of a user lasting from 7:00 p.m. to 7:30 p.m., any reservations from that user during that time should be fake demand. For the repeated reservation attempts, only the first reservation made is considered as the real demand, and according to staff's experience, any reservations made within the next 15 minutes are fake demands.

4.2. Vehicle Demand Forecasting. The approach of deep neural networks was selected due to its high performance in various fields such as computer vision, natural language processing, and traffic management [24–26]. In particular, Recurrent Neural Network (RNN) is a type of deep neural network that is suited to variable length time-series problems, allowing the use of past information to predict “future” values. One of the popular RNN models is Gated-Recurrent Unit (GRU), the performance of which can be compared with Long Short-Term Memory (LSTM) while using less number of parameters [27]. Since GRU computes the input only in the forward direction, Bidirectional GRU (BiGRU) is implemented to process the data in both directions. We also compare these three models, with a manual

```

Data : reservation logs
Result: demand logs with fake demand cleaned
sort all reservations by event start time;
group all reservations by user;
for each user do
  // (i) remove abnormal search logs after successful booking
  for each reservation from user do
    if reservation is successful usage then
      While next reservation time < current reservation end time do
        delete next reservation;
      end
    end
  end
  // (ii) remove repeated reservation attempts
  for each reservation from user do
    if current reservation time - previous reservation time <= 15 min. then
      if current and previous reservations have the same or nearby departure and destination stations then
        delete current reservation;
      end
    end
  end
end
end

```

FIGURE 6: Automatic fake demand removal algorithm.

model by experts, and then select the model with minimum root mean square error (RMSE) and the best simulation result as the final model. This section proposes a deep neural network based on the BiGRU structure for demand forecasting.

The architecture of the proposed model is shown in Figure 7. The demand of each station is normalized to be a fraction, where the summation of all stations is one at each time step, separately between departure and destination demands. BiGRU was used to extract features that should capture the correlation between departure and destination demand at each station across time. To improve upon accuracy, temporal features are included in our model. However, the order of time should be treated cyclically, for example, daily, weekly, monthly, etc. Therefore, we add an auxiliary input of cyclical time features, consisting of representations of the hour of day, day of week, and month of year. The extracted features and the cyclical time features are concatenated, and finally used as an input in the next fully connected layer, to separately forecast the destination and departure demand in the future time step. In practice, to predict the next day, our model will predict for the first hour of that day using the last 30 day's data as the input window. Then, to predict the next hour, we discard the oldest data and instead add the previous hour's prediction to the input. This process is repeated until we obtain the full day's predictions. We describe each part of the model as follows:

(i) BiGRU: it is a recurrent neural network that has been proven to be effective and fast as explored in previous

studies, especially in the domains of speech and bioinformatics [28–30]. BiGRU has the advantage over one-directional GRU in that it combines the outputs from a forward GRU with a backward unit, allowing it to capture relationships from both past and future time steps. In our network, we used two layers of BiGRU, each with 32 neural nodes, to extract features from departure and destination demand inputs.

- (ii) Fully connected layer: after obtaining the extracted features, we feed them into fully connected layers to predict the *destination* and departure demand in the target time step. All hidden, fully connected layers in our model have 256 neural nodes each. Each layer's weights are initialized according to He et al. [31] because their strategy is suitable for layers with ReLU activation function, which we describe in a later section.
- (iii) Activation function: rectified linear unit function (ReLU) was selected as an activation function to alleviate the vanishing gradient problem [32, 33]. We used this activation function for all hidden, fully connected layers.
- (iv) Regularization: to increase the numerical stability of our neural network, we perform batch normalization after every fully connected layer [34]. This is done before applying the activation function. After each activation layer, we also employed *dropout*

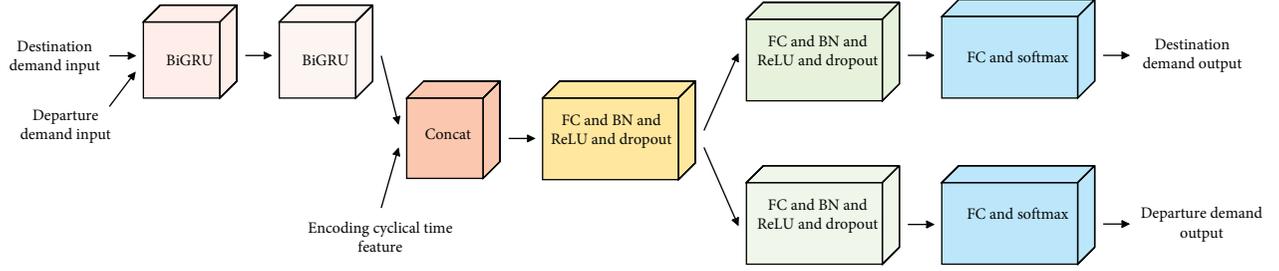


FIGURE 7: An architecture of our vehicle demand forecasting based on BiGRU.

layers to prevent overfitting by forcing the model not to rely on the same patterns all the time [35].

- (v) Encoding cyclical time features: as we expect time in our data to be cyclical most of the time, we need to encode this information into the model's input, so that the model can treat all points in time the same way [36]. For example, we want the model to address the boundary between December and January the same way it does with January and February. By assigning numerical values to each month (0 for January, 1 for February... and 11 for December), the gap between December and January will be different from that of consecutive months. Instead, we encode all cyclical time features with sine and cosine transformations. Each cyclical time feature is calculated as shown in equations (1) and (2), where i is the numerical representation of time and T is the maximum value of that time feature. For example, T for the day of week feature will be 7, and i being 1 represents Monday.

$$\text{cir}_x(i) = \sin\left(\frac{2\pi i}{T}\right), \quad (1)$$

$$\text{cir}_x(i) = \cos\left(\frac{2\pi i}{T}\right). \quad (2)$$

Our model has two outputs, one for the departure demand and another for the destination demand. As both of them are fractions that sum up to one, we use softmax as an activation function for both output layers. From preliminary investigation, we found that using softmax as is gives the model a tendency to assign high fractions to only a few stations; the remaining stations' fractions are not learned properly. To deal with this issue, we used temperature scaling, which adds a hyperparameter to the softmax function that can be calibrated to smooth down this tendency and allow the model to learn fractions of the low-ranked stations as well [37]. The softmax function with temperature scaling is defined as follows:

$$\sigma(s_i) = \frac{\exp((1/T)s_i)}{\sum_{j=1}^N \exp((1/T)s_j)}, \quad (3)$$

where s_i is the predicted demand for station i , N is the total number of stations, and T is the temperature scaling hyperparameter.

To train the model, we used a loss function (L), or objective function, based on categorical cross-entropy.

Normally, categorical cross-entropy is used in classification tasks, which has only one correct label. In this study, however, we applied this loss to make the model's output converge toward the ground truth fractions. Since our model has two outputs, our final loss function is the sum of categorical cross-entropy for both demand predictions. As a result, the loss function is defined as follows:

$$L = -\left(\sum_{i=1}^N t_{i,\text{depart}} \log(s_{i,\text{depart}})\right) + \left(\sum_{i=1}^N t_{i,\text{dest}} \log(s_{i,\text{dest}})\right), \quad (4)$$

where $t_{i,\text{depart}}$ and $t_{i,\text{dest}}$ are the ground truth fractions of departure and destination demand outputs for station i , respectively, $s_{i,\text{depart}}$ and $s_{i,\text{dest}}$ are the predicted demand fractions, and N is the total number of stations.

The predicted demands will be used to rank the stations, and the vehicle assignment table for the target time will be created accordingly. More specifically, the vehicle assignment process has to follow certain rules, such as the minimum requirement that almost all stations must have at least one vehicle and one empty parking lot. Here, we describe our vehicle assignment procedure that follows these rules:

- (1) We assign one vehicle to every station, except some of the stations, because they have only one parking lot each and the nearby areas are not as crowded as the other stations.
- (2) Following the ordering from the model's ranking output, we assign vehicles to the highest ranked station until there is only one remaining parking lot. For example, 3 vehicles are assigned to a station with 4 parking lots if the station is in the high ranking category.
- (3) The process is repeated until there are no more vehicles remaining. The final output is a list that shows the required number of vehicles at each station for the target hour.

In practice, our system removes fake demand and re-trains the model once a week. This generally helps to reduce computational costs and make the system capable of capturing any changes in demand.

4.3. Relocation Optimization. After obtaining the vehicle assignment table, the next step is to produce relocation trips for operation staff according to the number of available

```

Data: G: the network structure
Result: p: list path of minimum cost
k = maximum-flow(G);
p = [];
f = 0;
while f < k do
    t = shortest-path(G);
    m = infinity;
    for each node in t do
        if node.capacity < m then
            m = node.capacity;
    p.push(t);
    f = f + m;
    for each node in t do
        node.capacity = node.capacity - m;
return p;

```

FIGURE 8: Minimum Cost Maximum Flow algorithm.

vehicles currently presented at each station. In order to ensure that the produced trips are efficient in terms of distance, energy consumption, and staff's efforts, we applied the Minimum Cost Maximum Flow algorithm for the relocation optimization. This algorithm is well-known and widely used in cloud resource allocation [38], energy sharing management [39], resource management [40, 41], and network traffic management [42, 43]. The objective is to minimize the cost required to deliver the maximum amount of flows possible in a network. In our experiment, the nodes in the network represent the stations and the edges represent the distances between them.

The Minimum Cost Maximum Flow algorithm is illustrated in Figure 8. The input is the network structure (G), where each node refers to a station with its maximum capacity (the number of vehicles), and the link between nodes refers to a distance between them. First, the algorithm performs the "maximum-flow" to find a feasible flow through a capacity that obtains the maximum flow rate (k). Second, the algorithm performs the "shortest-path" to find the path (t) wherein the cost summation is the minimum. Then, the capacity of each node is subtracted by the minimum capacity (m) of the shortest path (t), while the cumulative flow (f) is added. The algorithm repeats until the cumulative flow is equal to the maximum flow rate (k). Finally, we obtain the list path of minimum cost. In this study, we used the Minimum Cost Maximum Flow algorithm implemented by Aric et al. [44].

Our relocation optimization can be represented by a network, as depicted in Figure 9. In our scenario, this optimization algorithm is customized to maximize the number of relocation vehicles, while minimizing the total distance to move those vehicles; therefore, the operation plan is efficient with less staff's efforts. The source node (S) is the node from which the flow starts, while the target node (T) is the node where all flows are terminated. In this case, the flow represents the total amount of vehicles in the service. As can be seen, there are two groups of nodes between the source and

the target nodes, which are departure stations and destination stations. Every departure station node has edges outbound to every destination station node. Each edge represents the path from the departure station to the destination station. These edges have costs corresponding to the distance between the departure station and the destination station, and have capacities corresponding to the available vehicles at each departure station. Meanwhile, the edges connecting the source node to departure stations have zero cost and capacity, which is equal to the number of vehicles available at the current time. Similarly, the edges from the destination stations to the target node have zero cost and capacity, which is equal to the number of vehicles available at the prediction time.

By running the algorithm on this network, we can obtain the optimized relocation trips for operation staff. The flows from each departure station to each destination station indicate the vehicle relocation trips that have to be done between these two stations. By optimizing the distance that the vehicle needs to travel, we can ensure the minimum energy and time consumption for our vehicle relocation tasks.

4.4. Relocation Management and Notification System. In order to completely deploy our vehicle relocation system in the real-world use case, it is necessary to implement an application that can automatically notify the operation staff whenever a relocation trip is needed. Figure 10 shows the monitoring dashboard and Line notification that we implemented for our system. As shown in the monitoring dashboard, the vehicle relocation trips issued by our model are classified into three types: active trips, operating trips, and completed trips. The active trips are the trips issued by the model but not yet accepted by any staff. The operating trips are the trips accepted by the operation staff and still in the relocation process. The completed trips are the trips already done by the staff. The staff leader usually monitors the dashboard and also has an authority to add or remove the trips manually. This is allowed in order to increase the flexibility of the relocation management. Once a new vehicle relocation trip is issued as an active trip, all of the operating staff will be notified by Line application, as shown in Figure 10(b). Then, the staff who are available can press "Accept" to accept this new trip. When the staff finish moving the vehicle to the destination station, the staff will press "Complete" to inform the system that this trip has been completely done.

For more clarification, Figure 11 depicts the sequence diagram of our vehicle relocation process. When the demand forecasting model pushes a new optimized vehicle relocation trip to the system, the system will notify all operation staff via Line application. Only the first operation staff who accepts the trip will work for this relocation trip. Other operation staff who press accept later will receive failed responses and the details showing the name of the operation staff who is working for this relocation trip. After the operation staff finishes working for the relocation trip, the operation staff has to report to the system using the Line application.

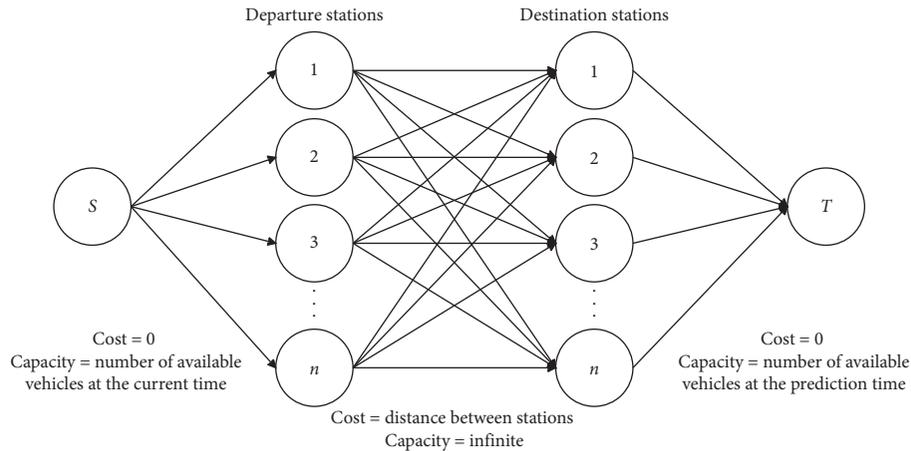


FIGURE 9: The network when the Minimum Cost Maximum Flow algorithm is applied to our relocation optimization.

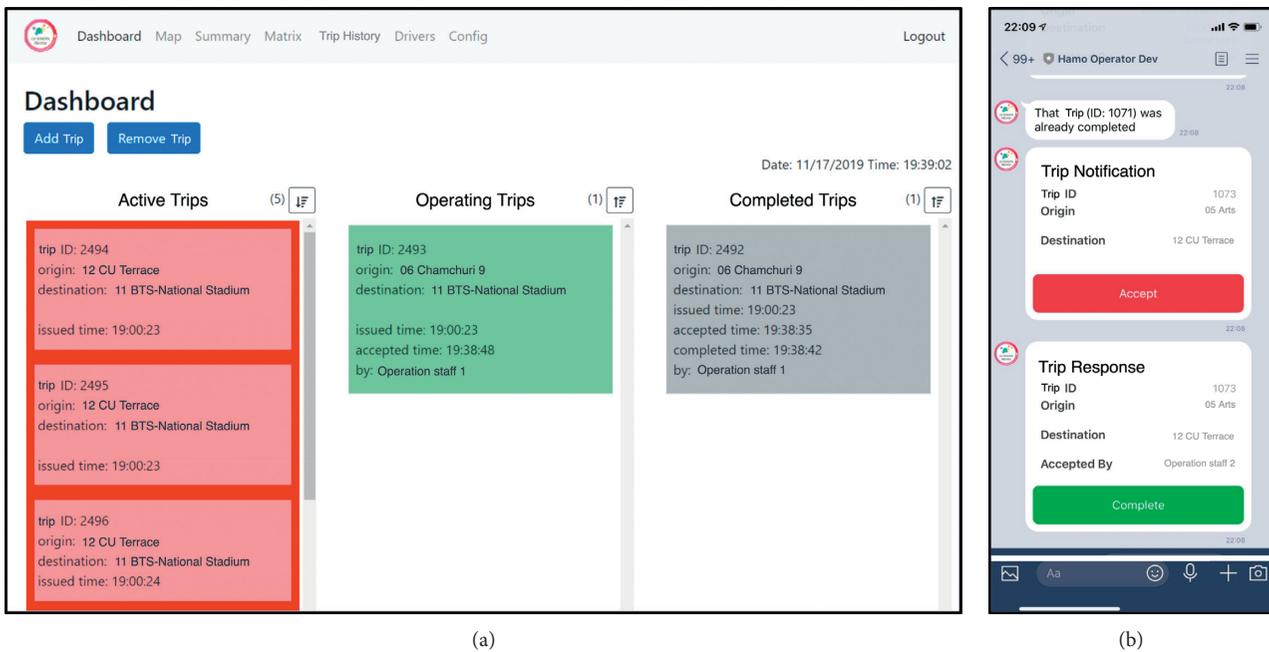


FIGURE 10: (a) Monitoring dashboard and (b) line notification.

Figure 12 illustrates our deployment diagram. Our computation server runs three processes including the shared database, the model service, and the backend of the core service. Another server, named CU-HAMO.COM, serves the administrative dashboard website, which is the frontend of the core service. We use Cloudflare’s service to certify our domain name “cu-hamo.com” and to use the HTTPS protocol. The HTTPS protocol is required by Line service when communicating with Line application servers.

5. Performance Evaluation

In this section, there are three experiments reported. First, we aim to show that our data preprocessing can really remove fake demands compared to experienced staff.

Second, many variations of the relocation models have been compared. Finally, the winner model was compared to experienced staff in the real environment showing that it can really improve in all measures.

5.1. *Experimental Setup.* In this study, the model is trained based on a history of usage logs. To make the experiment more realistic, a scenario based on the real data was simulated during September to November of 2019 in order to examine the performance of our system. The simulation environment is setup as follows:

- (i) The number of vehicles and the station are 30 and 22, respectively
- (ii) The service time in one day starts from 7 a.m. and ends at 7 p.m

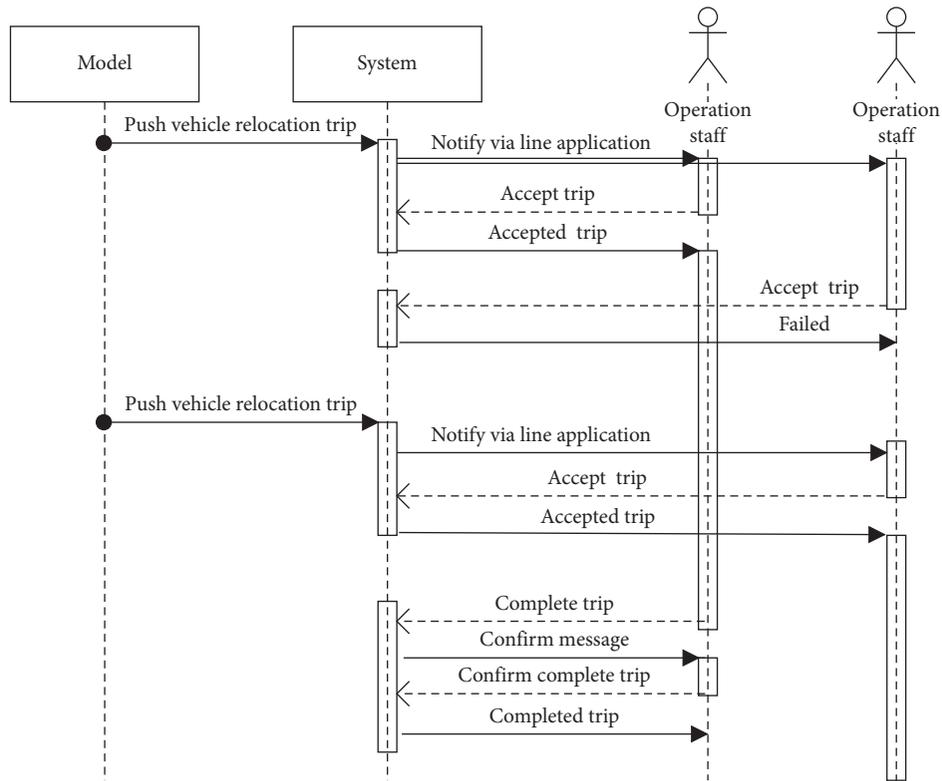


FIGURE 11: Sequence diagram of our vehicle relocation process.

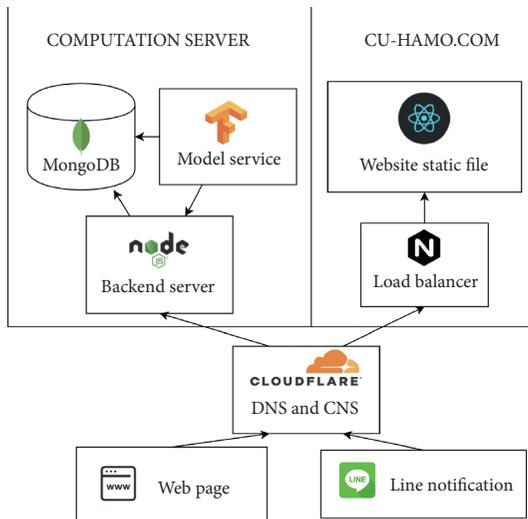


FIGURE 12: Deployment diagram.

- (iii) Each usage in the service history is processed chronologically
- (iv) Each user books a vehicle for no longer than 15 minutes
- (v) The system runs every hour, and its output will be used to relocate vehicles

In this setup, there are two main performance measures: the opportunity loss (booking search failures) and the number of relocation trips (staff efforts). The first metric was

used to measure how opportunity loss reduces, which is our main goal, while the second metric was selected to measure staff's workload in our system.

To make the results more reliable, the cross-validation in time series is employed by using the rolling basis method since there is an order of sequences in the vehicle usage. Thus, no future data are allowed to be used in the training data. As shown in Figure 13, the whole data form the usage log in 2019. There is a threefold cross-validation in time series. In the first iteration, the data in September were treated as test data for the model trained by the data from January until August. After that (in the second iteration), the data in September were integrated into the training set and the retrained model again, and then tested on the data in October. Finally (the last iteration), this procedure was also applied to the data in November as the test set.

5.2. Results of Data Preprocessing on Fake Demand Usages.

Since the model is trained based on vehicle usage logs, it is crucial to remove fake demands like search fails and multiple search logs from the same user. In this section, we compared unprocessed logs in September and October of 2019 to cleaned usage logs by (i) manual data preprocessing by experienced staff and (ii) our automatic data preprocessing. Table 1 shows that there are a lot of fake demands caused by no vehicle available and no parking lots available with more than 50% false demands compared to the cleaned data by experts. Also, our data preprocessing can really reduce those fake demands by more than 50% on average, which is comparable to how experts clean the data. The elimination of

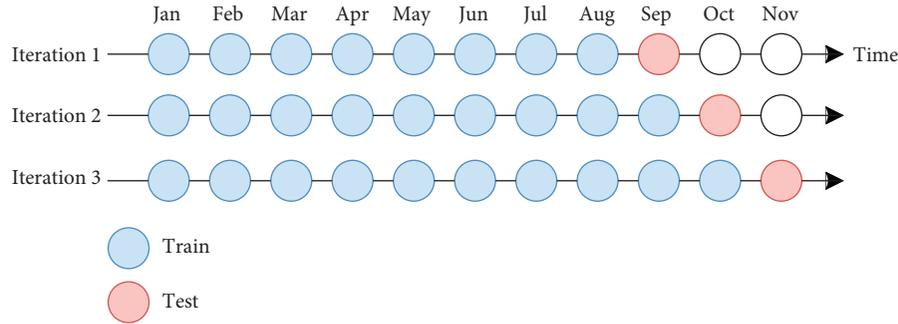


FIGURE 13: The threefold cross validation in time series of the vehicle usage in 2019, where the data in September, October, and November are treated as test sets.

TABLE 1: A comparison of usage logs among (1) unprocessed data, (2) cleaned data by experts, and (3) cleaned data by our algorithm.

Experiment	Unprocessed	Cleaned by experts	Cleaned by system
September 2019			
Failed usage (no vehicle case)	592	150 (-442)	297 (-295)
Failed usage (no station case)	545	174 (-371)	263 (-282)
Total failed usage	1,137	324 (-813)	560 (-577)
October 2019			
Failed usage (no vehicle case)	715	130 (-585)	322 (-393)
Failed usage (no station case)	492	149 (-343)	227 (-265)
Total failed usage	1,207	279 (-928)	549 (-658)

The number in the parentheses refers to the number of fake usages that can be reduced.

staff can benefit in many aspects: reduce human efforts, data cleansing consistency, and more accurate model due to better quality of the training data.

In conclusion, data preprocessing is really important and required in any real deployment scenario. Without removing fake demands, the forecasting results can be overestimated. Also, other relocation systems may encounter the same issue as in Hamo’s use case, so that they can directly adapt our data cleansing strategy to their systems.

5.3. Results of Relocation Algorithms. This experiment aims to compare various relocation algorithms based on two measures. The first measure is the Root Mean Square Error (RMSE), which is the evaluation of model precision. The second measure is the opportunity loss, which is the amount of search failures resulting in less services. The third measure is the number of relocation trips, which is the amount of efforts by staff to relocate vehicles in order to have vehicles available in the stations with more demands.

There are four methods in the comparison: three models are based on different forecasting techniques (BiGRU, GRU, and LSTM) and the last one is a relocation by experts. The experiment was conducted on the real usage data of three months (September, October, and November). Note that the results of November are based on data after we deployed our system. Table 2 shows the model precision of three models through RMSE. The results show that an averaging RMSE of BiGRU is slightly higher than that of LSTM and GRU. Figure 14 shows the result of each month separately, while Figure 15 shows the overall result of three months. In Figure 14(a), the results show that BiGRU is the winner in

TABLE 2: A comparison of RMSE among (1) LSTM, (2) GRU, and (3) BiGRU.

Model	September	October	November	Average
LSTM	0.1677	0.1762	0.1680	0.1706
GRU	0.1676	0.1761	0.1678	0.1705
BiGRU	0.1676	0.1751	0.1679	0.1702

terms of the reduction of opportunity losses unanimously in all three months. The number of total reduction losses is 6,778 or 4.10% in comparison to the manual relocation by experts (7,068 losses). In Figure 14(b), the results show that LSTM is the winner with total relocation trips of 7,102 rounds; however, BiGRU also provides a comparable result with total relocation trips of 7,142 rounds or 5.43% reduction of relocation efforts in comparison to the manual relocation by experts (7,552 rounds). It can be concluded that BiGRU-based relocation is the winner in both opportunity losses and relocation costs. In addition, all variations of our automatic relocation algorithms unanimously outperform a process managed by experts in both opportunity losses and less relocation efforts.

5.4. Results of the Real Deployment. Apart from a contribution in terms of algorithm advancement, we also implement a web application with automatic rule-based notification via Line to staff. Our model was trained based on the historical data from December 2017 to October 2019. Then, the deployment in the real-world scenario was conducted in November 2019. The first half of November (1st–11th) was controlled manually by experts from the

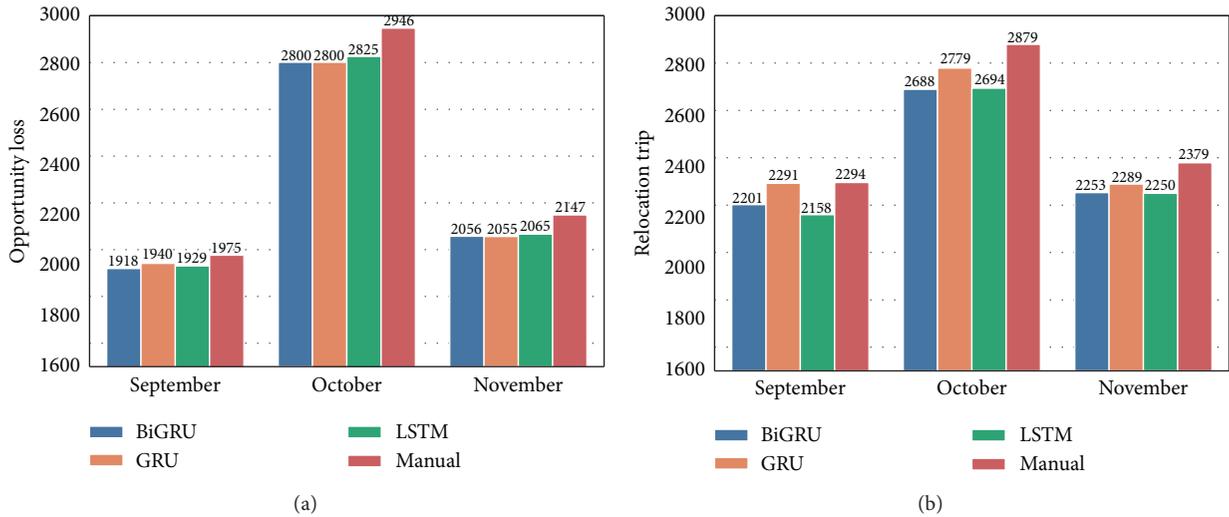


FIGURE 14: A comparison of four relocation algorithms: three are automatic models (BiGRU, GRU, and LSTM) and the last one is a manual model by experts in terms of (a) opportunity losses and (b) relocation trips.

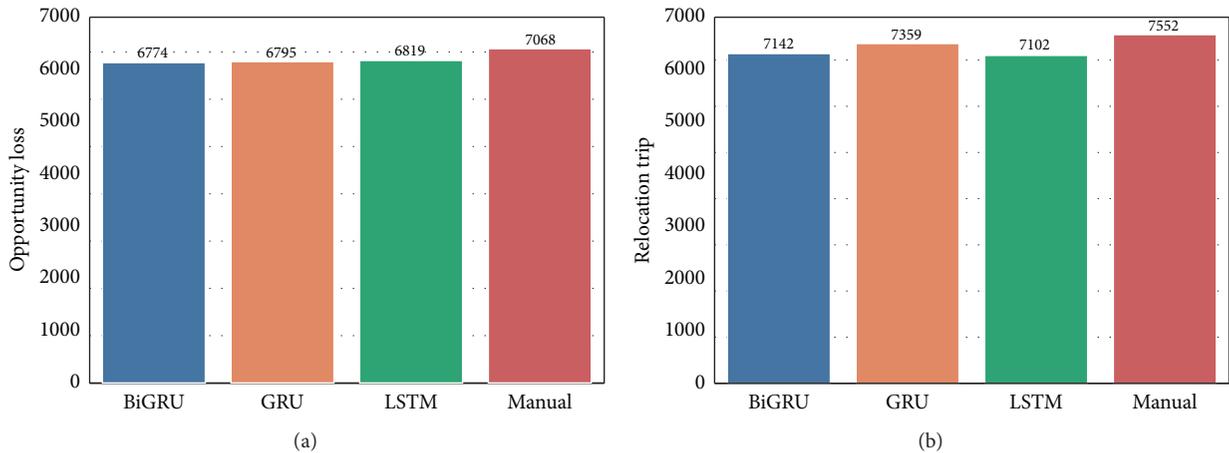


FIGURE 15: A comparison of four relocation algorithms on overall result of all three months in terms of (a) opportunity losses and (b) relocation trips.

operation room, while the other half (12th–26th) was operated automatically by our model (without any effort by the experts). There are three measures including opportunity losses (search failures), relocation trips (costs), and vehicle usages (higher means more services). The first two measures are compared to the last measure (vehicle usage) in order to show a ratio between losses over gains.

In Figure 16(a), the results show that there are 89 (63%) usages and 53 (37%) losses over the total demand of 142 during the relocation period by experts, and there are 86 (66%) usages and 45 (34%) losses over the total demand of 131 during the relocation period by our application. This shows that the system can provide higher usages (+3%) with lower losses (−3%) when compared with experts.

In Figure 16(b), the results demonstrate that there are 64 (42%) relocation trips (costs) over the total demand of 142 during the relocation period by experts, and there are 54 (39%) relocation trips (costs) over the total demands of 131 during the

relocation period by our application. This illustrates that the system also reduces the amount of relocation efforts by 3%. Moreover, there was a report from the staff that the amount of their work can be reduced for about 17 man-hours/week based on the saving hours per week manually spent by Hamo staff on this task before having our system.

In conclusion, our application cannot only increase the number of usages, but it also reduces opportunity losses and relocation efforts. This is one of the biggest contributions of our paper since this result is based on the real use case, while most prior works were evaluated in a simulated environment. Although our algorithms are designed specifically for the Hamo car-sharing system, they can be applied to other relocation systems with large scale users, vehicles, and stations. Demand forecasting and relocation optimization are the common modules in every relocation system. Thus, those systems with large-scale setups can benefit from our proposed methods.

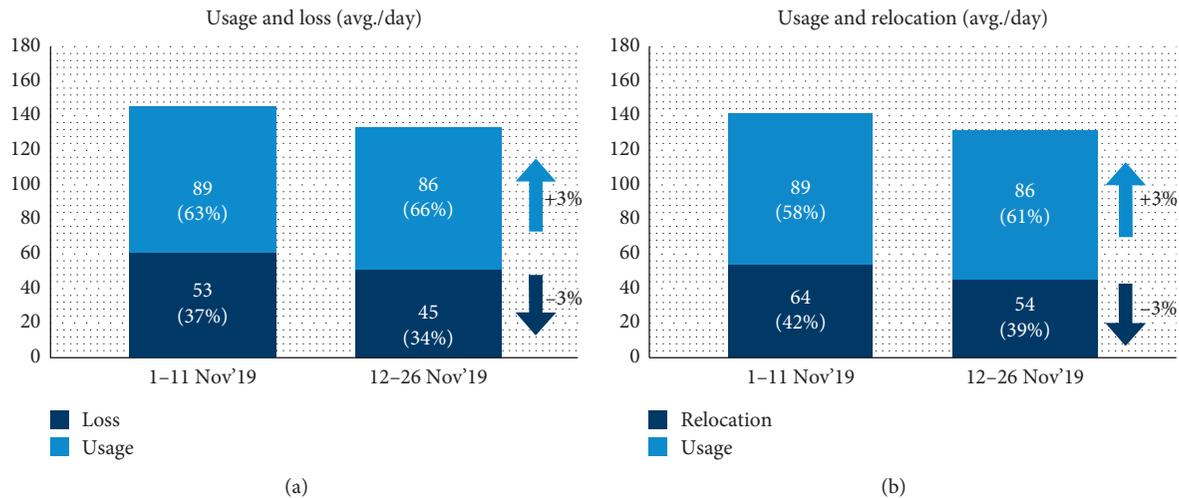


FIGURE 16: A comparison in November of 2019 between the relocation by experts during the first half and by our application during the second half: (a) usages over opportunity losses (failed usages) and (b) usages over relocation trips (staff costs).

From our experience in this real use case, it is concluded that there should be a minimum number of vehicles in each station in order to guarantee the service in all stations. If there is no limit on the number of relocated vehicles, it can cause a shortage of available vehicles in some stations. To address this concern, we have already implemented the condition of a minimum number of vehicles at each station in our system.

6. Conclusion

In this paper, we aim to propose an automatic vehicle relocation platform for a car-sharing scenario. The contributions are shown in terms of the deep-learning-based algorithm and the real production system. For the algorithm, the data preparation is proposed to remove fake demands, then the recurrent neural networks are applied to forecast the demands, and finally the Minimum Cost Maximum Flow algorithm (Min-Cost Max-Flow) is employed to optimize the relocation trips. The experiment was conducted in a real scenario, entitled “CU Toyota Ha:mo,” at Chulalongkorn University, Thailand in 2019. The results show that our data preparation can reduce more than 50% of fake demands, which is comparable to manual processing by human experts. Also, the best relocation algorithm is Bidirectional GRU along with Min-Cost Max-Flow; it outperforms the operation by human experts with lower losses for 4.10% and less number of relocation trips for 5.43%. The real web application has fully replaced a system operated by humans. By comparison to the previous human-operated system, the result is so remarkable that it increases usage by 3% (from 86% to 89%), while reducing both opportunity losses and staff efforts by 3% (from 37% to 34%) and 3% (from 42% to 39%). There was a report from the staff that the amount of their work can be reduced by about 17 man-hours/week.

As in the case of our relocation solution for the Hamo case study, the whole system can be integrated into other car-sharing systems where the relocation is performed by the

operator. Furthermore, each of our proposed modules can be applied separately to solve the issue and increase the performance of other systems. First, the data preprocessing module can be used to deal with the fake demand issue. Second, the vehicle demand forecasting module can be helpful for organizing our supplies in advance. Finally, the relocation optimization module can maximize the profit while minimizing the operation costs.

In the future, we plan to extend our solution into larger areas, such as districts or cities. If the car-sharing system covers an entire area of the city, it may be infeasible to re-balance the vehicles only by the staff. For such large areas, there should be an alternative strategy, such as user cooperation-based relocations.

Data Availability

Access to data is restricted due to commercial confidentiality.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by CU Toyota:Hamo. The authors would also like to thank Attawit Chaiyaraj and Amornpong Trakarnkulphun, Department of Computer Engineering, Chulalongkorn University, for their valuable comments. This research was funded under the project “CU Toyota:Hamo” in 2019.

References

- [1] M. J. Beckmann, “Traffic congestion and what to do about it,” *Transportmetrica B: Transport Dynamics*, vol. 1, no. 1, pp. 103–109, 2013.

- [2] F. Zhou, Z. Zheng, J. Whitehead, R. Perrons, L. Page, and S. Washington, "Projected prevalence of car-sharing in four Asian-Pacific countries in 2030: what the experts think," *Transportation Research Part C: Emerging Technologies*, vol. 84, pp. 158–177, 2017.
- [3] S. Frost, *Future of carsharing market to 2025 - technology advancements, market consolidation and government initiatives to influence market growth over the next decade*, Vol. 8, Frost and Sullivan, San Antonio, TX, USA, 2016.
- [4] S. A. Shaheen, N. D. Chan, and H. Micheaux, "One-way carsharing's evolution and operator perspectives from the Americas," *Transportation*, vol. 42, no. 3, pp. 519–536, 2015.
- [5] S. Illgen and M. Höck, "Literature review of the vehicle relocation problem in one-way car sharing networks," *Transportation Research Part B: Methodological*, vol. 120, pp. 193–204, 2019.
- [6] A. Di Febbraro, N. Sacco, and M. Saeednia, "One-way car-sharing: solving the relocation problem," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2319, no. 12, pp. 113–120, 2012.
- [7] M. Barth, M. Todd, and L. Xue, "User-based vehicle relocation techniques for multiple-station shared-use vehicle systems," in *Proceedings of the Transportation Research Board 80th Annual Meeting*, Washington, DC, USA, January 2004.
- [8] E. Cepolina, A. Farina, and A. Pratelli, "Car-sharing relocation strategies: a state of the art," *WIT Transactions on State-of-the-Art in Science and Engineering*, vol. 109–120, 2014.
- [9] E. M. Cepolina and A. Farina, "A new shared vehicle system for urban areas," *Transportation Research Part C: Emerging Technologies*, vol. 21, no. 1, pp. 230–243, 2012.
- [10] A. Angelopoulos, D. Gavalas, C. Konstantopoulos, D. Kyriadis, and G. Pantziou, "Incentivized vehicle relocation in vehicle sharing systems," *Transportation Research Part C: Emerging Technologies*, vol. 97, pp. 175–193, 2018.
- [11] M. V. Behzad, M. Iraj, M. Nezam, and K. Esmaili, "Balancing public bicycle sharing system using inventory critical levels in queuing network," *Computers & Industrial Engineering*, vol. 141, p. 106277, 2020.
- [12] W. Herbawi, M. Knoll, M. Kaiser, and W. Gruel, "An evolutionary algorithm for the vehicle relocation problem in free floating car sharing," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2873–2879, Vancouver, BC, USA, December 2016.
- [13] C. Gambella, E. Malaguti, F. Masini, and D. Vigo, "Optimizing relocation operations in electric car-sharing," *Omega*, vol. 81, pp. 234–235, 2017 12.
- [14] D. Yu, Z. Li, Q. Zhong, Y. Ai, and W. Chen, "Demand management of station-based car sharing system based on deep learning forecasting," *Journal of Advanced Transportation*, vol. 2020, pp. 1–15, 2020 02.
- [15] A. G. H. Kek, R. L. Cheu, Q. Meng, and C. H. Fung, "A decision support system for vehicle relocation operations in carsharing systems," *Transportation Research Part E: Logistics and Transportation Review*, vol. 45, no. 1, pp. 149–158, 2009 01.
- [16] D. Zhao, X. Li, and J. Cui, "A simulation-based optimization model for infrastructure planning for electric autonomous vehicle sharing," *Computer-Aided Civil and Infrastructure Engineering*, 2019.
- [17] X. Lu, Q. Zhang, Z. Peng, Z. Shao, H. Song, and W. Wang, "Charging and relocating optimization for electric vehicle car-sharing: an event-based strategy improvement approach," *Energy*, vol. 207, p. 118285, 2020.
- [18] R. Martin, K. Mor, B. Burak, and G. Nikolas, "Dynamic prediction-based relocation policies in one-way station-based carsharing systems with complete journey reservations," *Transportation Research Part B: Methodological*, vol. 130, pp. 82–104, 2019.
- [19] W. Ning, G. Jiahui, L. Xiang, and F. Tong, "A service demand forecasting model for one-way electric car-sharing systems combining long short-term memory networks with Granger causality test," *Journal of Cleaner Production*, vol. 224, p. 118812, 2020.
- [20] L. Qu, W. Li, W. Li, D. Ma, and Y. Wang, "Daily long-term traffic flow forecasting based on a deep neural network," *Expert Systems with Applications*, vol. 121, pp. 304–312, 2019.
- [21] Q. Zhaowei, L. Haitao, L. Zhihui, and Z. Tao, "Short-term traffic flow forecasting method with M-B-LSTM hybrid network," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2020.
- [22] D. Ma, B. Sheng, S. Jin, X. Ma, and P. Gao, "Short-term traffic flow forecasting by selecting appropriate predictions based on pattern matching," *IEEE Access*, vol. 6, pp. 75629–75638, 2018.
- [23] D. Ma, X. Song, and P. Li, "Daily traffic flow forecasting through a contextual convolutional recurrent neural network modeling inter-and intra-day traffic patterns," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.
- [24] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: a brief review," *Computational Intelligence and Neuroscience*, vol. 2018, pp. 1–13, Article ID 7068349, 2018.
- [25] J. Bai and Y. Chen, "A deep neural network based on classification of traffic volume for short-term forecasting," *Mathematical Problems in Engineering*, vol. 2019, pp. 1–10, Article ID 6318094, 2019.
- [26] G. Lingani, D. B. Rawat, and M. Garuba, "Smart traffic management system using deep learning for smart city applications," in *Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, March 2019.
- [27] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proceedings of the NIPS 2014 Workshop on Deep Learning*, Montreal, Canada, December 2014.
- [28] S. Yu, J. Wang, J. Liu, R. Sun, S. Kuang, and L. Sun, "Rapid prediction of respiratory motion based on bidirectional gated recurrent unit network," *IEEE Access*, vol. 8, pp. 49424–49435, 2020.
- [29] H. M. Lynn, S. B. Pan, and P. Kim, "A deep bidirectional GRU network model for biometric electrocardiogram classification based on recurrent neural networks," *IEEE Access*, vol. 7, pp. 145395–145405, 2019 09.
- [30] V. Vukotić, C. Raymond, and G. Gravier, *A Step beyond Local Observations with a Dialog Aware Bidirectional GRU Network for Spoken Language Understanding*, pp. 3241–3244, Interspeech, San Francisco, CA, USA, 2016.
- [31] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: surpassing human-level performance on ImageNet classification," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034, Santiago, CL, USA, December 2015.
- [32] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of Trends in Practice and Research for Deep Learning," <https://arxiv.org/abs/1811.03378>.
- [33] G. Dahl, T. Sainath, and G. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout,"

- IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 26, pp. 8609–8613, 2013.
- [34] S. Ioffe and C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift,” *Proceedings of the Machine Learning*, vol. 37, pp. 448–456, 2015.
- [35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, p. 1929, 2014–1958.
- [36] G. Petneházi, “Recurrent neural networks for time series forecasting,” 2019, <https://arxiv.org/abs/1901.00069>.
- [37] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in *Proceedings of the ICML*, Haifa, Israel, June 2017.
- [38] M. Hadji and D. Zeghlache, “Minimum cost maximum flow algorithm for dynamic resource allocation in clouds,” in *Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing*, pp. 876–882, Honolulu, HI, USA, June 2012.
- [39] D. Benda, X. Chu, S. Sun, T. Q. S. Quek, and A. Buckley, “Renewable energy sharing among base stations as a min-cost-max-flow optimization problem,” *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 1, pp. 67–78, 2019.
- [40] D. Gulotta, D. Kane, and A. Spann, “Application of min-cost flow to airline accessibility services,” *UMAP Journal*, vol. 27, pp. 367–385, 2006.
- [41] P. S. Pattnayak, S. S. Patra, and J. Pradhan, “Optimizing the network flow in cloud supply chain management,” *International Journal of U- and E- Service, Science and Technology*, vol. 10, no. 5, pp. 43–54, 2017.
- [42] T. H. Szymanski, “Max-flow min-cost routing in a future-internet with improved QoS guarantees,” *IEEE Transactions on Communications*, vol. 61, no. 4, pp. 1485–1497, 2013.
- [43] C. Ababei and R. Kavasseri, “Speeding-up network reconfiguration by minimum cost maximum flow based branch exchanges,” in *Proceedings of the IEEE PES Transmission and Distribution Conference and Exposition: Smart Solutions for a Changing World*, vol. 1–7, New Orleans, LA, USA, April 2010.
- [44] A. A. Hagberg, D. A. Schult, and P. J. Swart, “Exploring network structure, dynamics, and function using NetworkX,” in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pp. 11–15, Pasadena, CA, USA, August 2008.