Record: Joint Real-Time Repositioning and Charging for Electric Carsharing with Dynamic Deadlines

Guang Wang[†], Zhou Qin[†], Shuai Wang[★], Huijun Sun[§], Zheng Dong[‡], Desheng Zhang[†]
[†]Rutgers University, [★]Southeast University, [§]Beijing Jiaotong University, [‡]Wayne State University
{guang.wang,zq58}@rutgers.edu,shuaiwang@seu.edu.cn,hjsun1@bjtu.edu.cn,dong@wayne.edu,desheng@cs.rutgers.edu

ABSTRACT

Electric carsharing, i.e., electric vehicle sharing, as an emerging mobility-on-demand service, has been proliferating worldwide recently. Though providing convenient, low-cost, and environmentally-friendly mobility, there are also some potential roadblocks in electric carsharing services due to existing inefficient fleet management strategies, which relocate the vehicles using predefined periodic schedules without self-adapting to the highly dynamic user demand, and many practical factors like time-variant charging pricing also have not been fully considered. To remedy these problems, in this paper, we design Record, an effective fleet management system with joint Repositioning and Charging for electric carsharing based on dynamic deadlines to improve its operating profits and also satisfy users' real-time pickup and return demand. Record considers not only the highly dynamic user demand for *vehicle repositioning* (i.e., where to relocate) but also the time-varying charging pricing for *charging scheduling* (i.e., where to charge). To perform the two tasks efficiently, in Record, we design a dynamic deadline-based distributed deep reinforcement learning algorithm, which generates dynamic deadlines via usage prediction combined with an error compensation mechanism to adaptively search and learn the optimal locations for satisfying highly dynamic and unbalanced user demand in real time. We implement and evaluate the Record system with 10-month real-world electric carsharing data, and the extensive experimental results show that our Record effectively reduces 25.8% of charging costs and reduces 30.2% of vehicle movements by workers, and it also satisfies user demand and achieves a small runtime overhead at the same time.

CCS CONCEPTS

• **Applied computing** → **Transportation**; • **Computing methodologies** → *Planning and scheduling*.

KEYWORDS

Electric carsharing, Fleet management, Dynamic deadline, Deep reinforcement learning

KDD '21, August 14–18, 2021, Virtual Event, Singapore.

ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467112

ACM Reference Format:
Guang Wang[†], Zhou Qin[†], Shuai Wang[★], Huijun Sun[§], Zheng Dong[‡], Desheng Zhang[†]. 2021. Record: Joint Real-Time Repositioning and Charging for Electric Carsharing with Dynamic Deadlines. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3447548.3467112

1 INTRODUCTION

As an innovative app-based shared mobility mode, carsharing (e.g., Zipcar and car2go) has been experiencing rapid growth worldwide due to its convenience and flexibility for use [17], which also has the potential to reduce the use of privately-owned vehicles for traffic congestion alleviation. For example, the carsharing service provider car2go reached three million users within sixteen months, becoming the largest operator in the flexible carsharing sector, and it is expected that the number of carsharing users will reach 36 million by 2025 [1]. In addition, as an additional social benefit, electric carsharing services are the most effective way to introduce the public to electric vehicles, so more and more cities and operators start to promote electric carsharing recently for gasoline consumption and carbon footprint reduction [2].

Even though being more flexible and cost-efficient for people with low annual vehicle usage, the electric carsharing service is also facing many practical challenges during its promotion and fleet management processes, e.g., unbalanced usage patterns and unprofitable business, so potential users may balk from using the service if there are no available vehicles nearby or no parking spots available near their destinations. There are typically two types of tasks for electric carsharing fleet management to satisfy the future user demand: (i) vehicle **Repositioning/Relocation**, i.e., deciding where to relocate vehicles, which means proactively moving unoccupied shared electric vehicles (i.e., EVs) from one service station to another service station by workers, and (ii) vehicle **Charging**, i.e., deciding where to charge shared EVs, which means moving low-battery shared EVs from service stations to charging stations and moving fully-charged EVs from charging stations back to service stations by workers.

Not surprisingly, many existing works have been done to improve the operational efficiency of carsharing by fleet management [7, 14]. However, the majority of these works focused on vehicle repositioning only [7] or charging issue only. Although some recent works [6] have been done to improve the operational efficiency of electric carsharing by considering both vehicle repositioning and charging scheduling, most of them mainly focused on theoretical optimization models and lacking enough data-driven observations. Thus, many important practical factors (e.g., user behavior and preference, time-variant charging pricing, and strict timing requirements of user demand) have not been captured by them. Admittedly, there are lots of works focusing on location-based search for different types of vehicles, e.g., taxi [23], ridesharing [12, 21], bikesharing [10, 13], e-scooter sharing [8], carsharing [14], but most of these existing works set predefined periodic schedules for vehicle repositioning or charging scheduling (e.g., set 10 minutes as a time slot to make the management decisions [12, 14, 21]). It leads to two potential drawbacks: (i) it is challenging to fully satisfy the user demand during intensive usage peaks; (ii) it may cause unnecessary decision making during low demand time periods and result in extra operational overhead.

In this paper, to advance existing works, we design `Record`, a new data-driven fleet management system with joint repositioning and charging scheduling for electric carsharing to improve its operational efficiency while satisfying user real-time pickup and return demand. We first utilize a long-term accumulated electric carsharing dataset to unveil some real-world issues related to user behaviors and complicated charging problems. Based on our data-driven findings, we realize that designing an efficient fleet management strategy considering both vehicle repositioning and charging is also challenging due to *possible conflicting relationships* (e.g., meet the pickup and return demand of users vs. improve the profit of the fleet) and many *confounding factors* (e.g., individual user behaviors like spatiotemporal usage preference, time-variant charging pricing, the availability and reachability of shared EVs). To address these challenges, in `Record`, we design a *Dynamic Deadline-based Distributed Deep Reinforcement Learning* algorithm to learn sophisticated decisions, which has two key advantages for electric carsharing fleet management: (i) the dynamic deadline strategy helps the system to adaptively satisfy the time-varying unbalanced pickup and return demand based on usage prediction and an error compensation mechanism; (ii) the distributed deep reinforcement learning-based decision making enables a long-term benefit of the system and improves scalability at the cost of less coordination between shared EVs, which causes a small runtime overhead for the real-time requirement.

In particular, the key contributions of this paper include:

- We conduct an extensive data-driven investigation by working with an electric carsharing operator, from which we found and address some practical issues of electric carsharing: (i) unique usage patterns and user preference, (ii) time-variant charging pricing, and (iii) the impact factors related to user behaviors, etc. Based on theses observations, we design the first dynamic deadline-based carsharing fleet management system called `Record` to decide where to optimally relocate and charge shared EVs with these factors considered.
- In `Record`, we design a dynamic deadline-based distributed deep reinforcement learning algorithm to learn which service station to relocate and which charging station to charge for each shared EV. It has two major components: (i) A prediction-based dynamic deadline mechanism is utilized to adapt the highly dynamic demand and supply, where the prediction is performed based on the features we capture from our data-driven observations, and an error compensation mechanism is also introduced to make our `Record` more robust to the prediction error. (ii) Based on the dynamic deadline setting, a distributed deep reinforcement learning module is presented, which enables long-term benefits of the system with a small runtime overhead, and it has the potential to make our system more sustainable.
- More importantly, we implement and extensively evaluate our dynamic deadline-based real-time fleet management system `Record` for repositioning and charging of shared EVs with real-world multi-source data from an electric carsharing operator, including 10-month detailed order records from over 12,000 unique users and the metadata of stations. The experimental results show our `Record` effectively reduces the charging cost by 25.8% and reduces 30.2% of vehicle movements by workers without sacrificing user experience.

2 PRELIMINARY AND MOTIVATION

2.1 Electric Carsharing System and Operation

A typical electric carsharing operation paradigm is shown in Fig. 1. There are four main parties in the system, i.e., operator, users, shared EVs, and stations. The operator provides a fleet management system to monitor all real-time status information of users, shared EVs, and stations, and makes decisions. In particular, the real-time location information and order information of users are recorded and uploaded when they use the mobile app. The real-time location and status information of shared EVs are also periodically uploaded to our servers via communication devices. The transaction information is recorded when users return shared EVs.

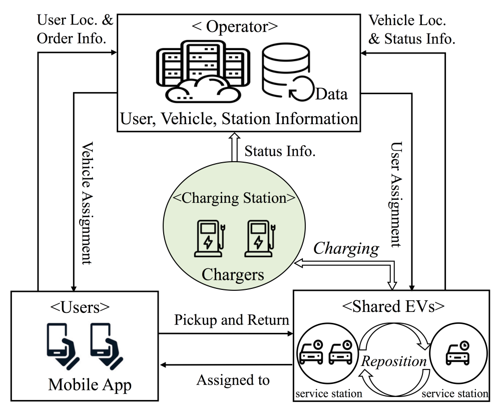

Figure 1: An electric carsharing operation paradigm.

Ideally, if there are an unlimited number of shared EVs and parking spots at each service station, users' pickup and return demand can be satisfied trivially. However, this assumption is normally not realistic. The carsharing operator usually possesses a limited number of shared EVs and parking spots at each service station due to high costs. Thus, some efforts are needed to balance the demand and supply, which includes two tasks: relocating shared EVs between service stations and driving low-battery shared EVs to charge in charging stations and then distributing fully-charged EVs back to service stations for satisfying the future demand. Intuitively, given the unbalanced user demand and supply, **how to effectively decide the optimal locations for shared EVs to relocate and charge** is essential to increase operating profits for the electric carsharing operators. To better understand the research challenges that exist in designing the fleet management system for repositioning and charging, we first perform a comprehensive analysis based on one real-world dataset.

2.2 Data Description

In this paper, we utilize a real-world electric carsharing operation dataset collected from the Chinese city Beijing. The time span of the dataset is from January 2017 to October 2017. The dataset includes three different types of data, i.e., vehicle usage data, vehicle GPS data, and station data (including information of service stations and charging stations). The details of the three types of data are shown as follows.

- **Vehicle Usage Data** includes all users' carsharing usage records. Each usage record consists of 26 fields describing shared EVs, users, and usage-related information, e.g., the order number, the user ID, user age, gender, workplace & occupation, order time, vehicle pickup and return time and station, the vehicle ID, usage time, and payment, etc.
- **Vehicle GPS Data** is collected via the on-board device on each shared EV. Each GPS record includes fields that describe the real-time status of each shared EV, e.g., vehicle ID, timestamp, and longitude & latitude.
- **Station Data** describes the service station and charging station information, e.g., the station IDs, the station names, coordinates (i.e., longitudes and latitudes), the number of parking spots in each service station, and the number of charging points in each charging station.

2.3 Data-Driven Investigation

Based on the multi-source dataset, we perform an extensive data-driven analysis to understand the existing issues in electric carsharing and motivate our design. The details are shown below.

2.3.1 Highly Dynamic Pickup and Return Patterns. Fig. 2 shows the pickup and return patterns of shared EVs in one week, and we found the pickup and return patterns are highly dynamic in different hours of different days. In particular, (i) We found that the pickup peaks and return peaks are at different hours of a day, e.g., 9:00 vs. 18:00, and the highest pickup demand is around 18:00 of Friday and the highest return demand is around 9:00 of Monday. (ii) The usage patterns on weekends are different from that on weekdays, e.g., the pickup peaks of weekends appear around 11:00, and the return peaks are around 20:00.

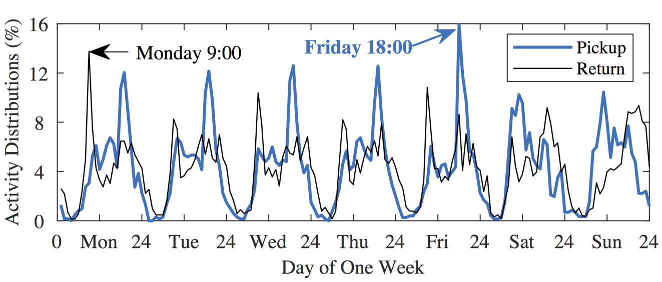

Figure 2: Pickup and return distributions in one week.

2.3.2 Unbalanced Spatiotemporal Usage Patterns. We then further investigate the fine-grained pickup and return distributions of different service stations at different hours of a day. As shown in Fig. 3, the red circles mean there are more pickups than returns in these service stations, and the aquamarine circles mean there are more returns in these service stations. The size of each circle stands for the absolute value of the difference between the number of returns and the number of pickups, i.e., |# of returns − # of pickups|.

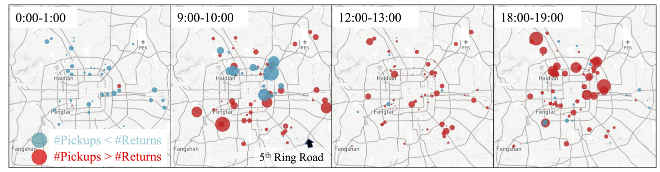

Figure 3: Spatiotemporal unbalanced usage patterns.

We found the pickup and return patterns have significant differences between different hours of a day. For example, there are more vehicle returns in most service stations during late-night hours (e.g., 0:00-1:00), and both the number of pickups and returns are small. In the morning rush hours (e.g., 9:00-10:00), there are very high pickup demand and return demand in different areas, e.g., more pickups in residential areas and more returns in IT industrial areas. The number of pickups increases during the day time, and it peaks in the evening rush hours (e.g., 18:00-19:00). To meet users' highly dynamic pickup and return demand, there should be enough shared EVs for users to pick up and enough parking spots for users to return vehicles in each service station. However, this is not easy to be achieved by users themselves, and it is necessary for the operator to hire specialized workers to relocate shared EVs between service stations and move low-battery vehicles to charge so that they can be utilized by users later.

2.3.3 Time-Variant Charging Pricing. In addition, we found the charging rates in Beijing are divided into three types, i.e., off-peak prices (low rates), flat prices (medium rates), and peak prices (high rates), and the corresponding electricity rates are 1.1946, 1.4950, and 1.8044 CNY/kWh, respectively [3]. The time-variant charging pricing of Beijing is shown in Fig. 4. We found the peak price is 51% higher than the off-peak price, which means the charging costs can potentially reduce 51% if the operator charges shared EVs in off-peak charging pricing hours instead of peak charging pricing hours. Hence, the charging prices should also be considered for the electric carsharing fleet management, but it is rarely considered by existing carsharing works.

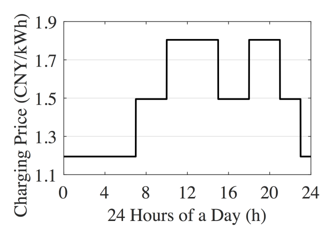

Figure 4: Time-variant charging pricing in Beijing.

Hence, based on our data-driven investigations, we found it is nontrivial to design an efficient electric carsharing fleet management system with both vehicle repositioning and charging consideration due to many practical factors, e.g., highly dynamic and unbalanced usage patterns caused by individual user behaviors and preference, combined with the time-variant charging pricing, etc.

3 KEY IDEA AND PROBLEM FORMULATION

Definition 3.1. **Deadline**: As shown in Fig. 5, suppose we divide a long time period T into h consecutive intervals, e.g., $\{I_1, I_2 \ldots, I_h\} = \{(d_0, d_1], (d_1, d_2], \ldots, (d_{i-2}, d_{i-1}], (d_{i-1}, d_i], \ldots, (d_{h-1}, d_h]\}$, where d_0 is the start time of the time period and d_i is the deadline for decision completion. The $(i-1)th$ deadline is the time when we need to finish all vehicle repositioning and charging scheduling to satisfy the user demand by the $(i)th$ deadline. In other word, the vehicle repositioning and charging scheduling performed between the $(i-2)th$ and $(i-1)th$ deadlines will satisfy the user demand arising during the period between the $(i-1)th$ and $(i)th$ deadlines. The dynamic deadline means d_i is not predefined and unfixed, which needs to be learned from the real-world demand and supply and usages. The dynamic deadline generation method will be introduced in Sec. 4.

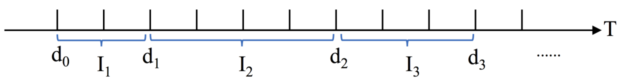

Figure 5: An illustration of the dynamic deadline.

In the electric carsharing services, user experience is impacted by the availability of shared EVs and parking spots in the service stations. Intuitively, users suffer from bad user experience if (i) the service station is empty when they want to pick up shared EVs, or (ii) the service station is full when they want to return a shared EV. Especially, those situations always occur during the rush hours at some busy service stations if the shared EVs are not managed effectively. Thus, to guarantee good user experience, these two dynamic usage behaviors, i.e., pickup and return, need to be taken into consideration. On the one hand, how many shared EVs are picked up and returned at each service station determines which service station is jammed or starved, thus impacts how to perform vehicle repositioning and charging scheduling; on the other hand, how to conduct the vehicle movement impacts how many available shared EVs and parking spots in each service station, which impacts the future pickups and returns of users.

Therefore, how to make decisions by considering both future pickup and return demand for good user experience is a key task of fleet management, which inspires us to (i) characterize the user pickup & return demand using dynamic service deadlines; (ii) design a practical and efficient decision making strategy to optimize the operating profit while all the deadlines can be met.

3.1 Key Idea of `Record`

In this paper, we design `Record`, a new data-driven fleet management system for electric carsharing to improve the shared EV fleet's **operating profits** (which are highly related to the revenue of the fleet from serving users, costs for charging, and the payment to workers for moving the shared EVs) without sacrificing **user experience** (e.g., pickup and return demand) based on joint repositioning and charging scheduling with dynamic deadline. In addition, `Record` considers different complicated real-world factors.

Due to the sequential decision characteristics of fleet management, we formulate the problem as a Markov Decision Process (MDP), and then we present a dynamic deadline-based distributed deep reinforcement learning method to achieve our goals, which not only makes the system adaptively satisfy the time-varying unbalanced pickup and return demand based on usage prediction, but also achieves long-term benefits with a small runtime overhead for the real-time requirement.

3.2 Problem Formulation

Formally, we model the electric carsharing fleet management problem as an MDP $\mathcal{G}$ for N agents, which is defined by a five-tuple $\mathcal{G} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$, where $\mathcal{S}$ is the set of states; $\mathcal{A}$ is the action space; $\mathcal{R}$ is the reward function; $\mathcal{P}$ is transition probability functions; and γ is a discount factor. In an MDP, an agent behaves in an environment according to a policy that specifies how the agent selects actions at each state of the MDP. The detailed formulation of the MDP $\mathcal{G}$ in our problem is shown below.

Agent: We consider each unoccupied shared EV (i.e., does not rent by users) as an agent, and only the unoccupied shared EVs can be scheduled by our system. Although the number of total agents in the fleet is always N, the number of agents in each time interval N_t is changing over time.

State $\mathcal{S}$: The state of a shared EV is defined as a two-dimensional vector indicating its spatiotemporal status. Suppose there are a set of service stations $\{SS\}$ and a set of charging stations $\{CS\}$, so each unoccupied shared EV may be in one of the service stations or charging stations. We define a local-view state of a shared EV, $s_{t,lo} = [t, l] \in \mathcal{S}_{lo}$, where t is the time index and $t \in ((d-1)th, dth]$, i.e., the time slot between the $(d-1)th$ and the dth deadline.), and $l \in \{SS\} \cup \{CS\}$ is the location index (i.e., which service station or charging station the shared EV is in). In this case, the finite local state space $\mathcal{S}_{lo}$ is a Cartesian product of the set of deadlines and the set of service stations + charging stations, i.e., $\mathcal{S}_{lo} = \{D\} \times (\{SS\} \cup \{CS\})$ and the number of states is $|\mathcal{S}_{lo}| = |D| \times (|SS| + |CS|)$. In addition to the local-view state, we also define a global state $s_{t,go}$ to capture the system status includes the number of shared EVs and parking spots availability at each service station and their real-time predicted usages in the next interval, and it also includes the number of unavailable shared EVs at each station in the current and next time interval. The global-view state $s_{t,go}$ will update in each time slot. Finally, the state of each available shared EV k during the time slot t can be represented as $s_t(k) = [s_{t,lo}(k), s_{t,go}(k)] \in \mathcal{S}(k)$.

Action $\mathcal{A}$: The action space of a shared EV k, $\mathcal{A}(k)$ specifies where it should go by the next deadline. We define six types of actions for the shared EV scheduling. (i) $\mathcal{A}_S$: Staying at the current service station; (ii) $\mathcal{A}_R$: Relocating to another service station to satisfy user demand in that station or make parking spots for vehicles that will be returned to this service station; (iii) $\mathcal{A}_C$: scheduling to Charge in a charging station; (iv) $\mathcal{A}_K$: Keeping charging at the charging station; (v) $\mathcal{A}_B$: moving Back to a service station; (vi) $\mathcal{A}_P$: Picked up by a user.

The action to take is decided by two factors: (i) *Availability* of shared EVs or parking spots for users in each service station by deadlines. (ii) *Reachability* to the charging stations or other service stations of shared EVs when performing repositioning or charging scheduling. That means for each service station $s_i \in \{SS\}$: (1) # of pickups $\leq$ # of available shared EVs; (2) # of shared EVs (available or unavailable) + # of returns $\leq$ # of parking spots. Besides, for each shared EV k, it becomes unavailable if its battery level decreases to below a threshold value η (e.g., 30%), which means the low-battery shared EVs should be scheduled/moved to charge in order to satisfy future demand. Since there are enough public charging stations in Beijing, and workers always check where there are available charging points, so we envision that the number of charging points is sufficient for shared EVs, and shared EVs can always be charged in the nearest available charging stations.

Reward $\mathcal{R}$: Reward usually determines the optimization goal and reflects the immediate sense of the action in a specific state. A typical measurement is to estimate the difference of the accumulated reward between with and without an action. We define three types of immediate rewards here, i.e., positive reward, zero reward, and negative reward, which capture the money transaction.

Specifically, (i) if a shared EV is picked up by a user during certain interval (i.e., $\mathcal{A}_P$), it will have a positive reward, which is equivalent to the money paid by the user; (ii) if a shared EV stays at the current service station (i.e., $\mathcal{A}_S$), it will not be used by users (i.e., no revenue) and also have no charging and moving costs, so the immediate reward is zero; (iii) if a shared EV is relocated from one service station to another service station (i.e., $\mathcal{A}_R$), scheduled from one service station to a charging station (i.e., $\mathcal{A}_C$), or moved backed to a service station from a charging station (i.e., $\mathcal{A}_B$), it will have a negative reward due to the labor cost. (iv) if a shared EV is charging in a charging station (i.e., $\mathcal{A}_K$), it will also have a negative reward due to the charging cost, which is explicitly related to the charging time and charging prices. Implicitly, the charging cost is also related to the previous usages and repositionings since these activities will directly cause the energy consumption of shared EVs. Hence, we define the reward function as follows

$$R_{\mathcal{U}} - C_C - C_{\mathcal{M}} = \sum_{i=1}^{m} R_u^{(i)} - \sum_{j=1}^{n} \left(\lambda \cdot T_c^{(j)}\right) - \phi \times z, \tag{1}$$

where $R_{\mathcal{U}}$ is the total revenue from serving users; C_C is the total charging cost of the electric carsharing fleet; $C_{\mathcal{M}}$ is the total labor cost for moving shared EVs, e.g., relocating shared EVs from one service station to another service station and moving shared EVs to charging stations and moving back to services stations; $R_u^{(i)}$ is the revenue from serving ith electric carsharing order; m is the total number of served orders; n is the total number of charges of the fleet. $T_c^{(j)}$ is a three-dimensional vector $T_c^{(j)} = \left[T_p^{(j)}, T_f^{(j)}, T_o^{(j)}\right]$ describing the charging time of jth charging event, where $T_p^{(j)}$, $T_f^{(j)}$, and $T_o^{(j)}$ denote the time in peak, flat, and off-peak charging pricing hours of the jth charging event, respectively. Similarly, we also describe the time-varying charging pricing as a three-dimensional vector $\lambda = \left[\lambda_p, \lambda_f, \lambda_o\right]$, where λ_p, λ_f, λ_o denote the charging prices during peak, flat, and off-peak hours, respectively (as shown in Fig. 4). ϕ is the payment to workers for each movement of shared EVs (**Note**: one worker can move only one vehicle at a time), and we intuitively envision it is a constant value for simplification; and z is the total number of vehicle movements by workers, so reducing the number of vehicle movements by workers will potentially increase the profit of the electric carsharing fleet.

Hence, if we can guarantee all current demand to be satisfied, then the total revenue from serving users $R_{\mathcal{U}}$ should be the same. Then we can improve the profit of the operator by reducing its charging cost C_C for the electric carsharing fleet and labor cost for moving shared EVs $C_{\mathcal{M}}$.

Probability function $\mathcal{P}$ defines the transition probability between states by taking action $\mathcal{S}\times\mathcal{A}\times\mathcal{S} \rightarrow [0, 1]$, e.g., $p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ denotes the probability of transition to the next state $\mathbf{s}_{t+1}$ given the action $\mathbf{a}_t$ in the current state $\mathbf{s}_t$. Our goal is to find a function that maps a state to the best action that each shared EV can take.

Discount factor γ essentially determines how much the reinforcement learning agents care about rewards in the distant future relative to those in the immediate future. The value of γ is typically selected from $[0, 1)$, so the final expected reward in the infinite horizon will be convergent and bounded to a finite number. If $\gamma = 0$, the agent will be completely myopic and only learn about actions that produce an immediate reward without considering future rewards.

4 DYNAMIC DEADLINE-BASED DISTRIBUTED DRL

In this section, we present the algorithm design in Record.

4.1 Dynamic Deadline Based on Prediction

As indicated in Fig. 2 and Fig. 3, the number of pickups and returns are highly dynamic in both spatial and temporal dimensions, so it may be challenging to satisfy the pickup demand and return demand in some usage peak hours and it may also cause high operational costs during low demand hours with only predefined periodic schedules. Hence, in this paper, we develop a dynamic deadline strategy to address this issue. An illustration of the dynamic deadline setting is shown in Fig. 6. Intuitively, if dense deadlines are set up during pickup & return peak hours and sparse deadlines in other hours, the system efficiency can be effectively improved by relocating or charging vehicles based on the deadline distribution.

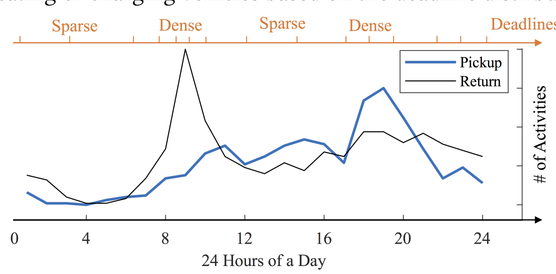

Figure 6: The key idea of the dynamic deadline setting.

Definition 4.1. We divide a long time period T into a set of small time slots, e.g., we set the time slot length as 5 minutes to capture the more fine-grained pickup and return patterns. We then define the **net flow** of a service station in a time slot as the number of returns (i.e., inflow) minus the number of pickups (i.e., outflow) in this time slot.

For each time slot t_i, we calculate the net flow f_{t_i} of a service station in this time slot based on the real-world order records, i.e., the net flow value will be deducted by one if there is a pickup activity and the net flow value will be added by one if there is a return activity.

Fig. 7(a) shows an example of the calculation of the *net flow* of a service station. In this example, there are 10 small time slots, and the value in each time slot denotes the net flow in this time slot, e.g., the net flow of the service station is 1 in t_1 and -2 in t_4, which means there are one more returns than pickups in t_1 and there are two more pickups than returns in t_4. Based on the net flow f_{t_i} of the service station in each time slot, we then calculate the accumulated net flow in multiple time slots, which is defined as follows:

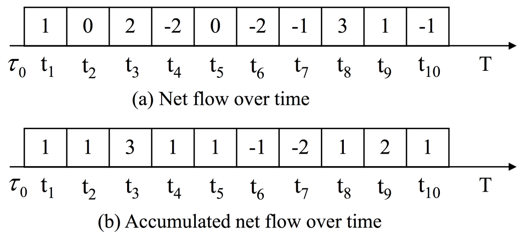

Figure 7: The (accumulated) net flow of a service station.

Definition 4.2. Given a series of time slots $\{t_1, t_2, \ldots, t_m\}$, the **accumulated net flow** by t_i of a service station is defined as the sum of all net flows of previous time slots, i.e., $F_{t_i} = \sum_{j=1}^{i} f_{t_i}$.

Fig. 7(b) shows the corresponding accumulated net flow of Fig. 7(a). Each value F_{t_i} in time slot t_i is the sum of the net flow from time slot t_1 to t_i. For example, the accumulated net flow in t_3 of the service station is $F_{t_3} = f_{t_1} + f_{t_2} + f_{t_3} = 3$.

Suppose there are N shared EVs $\{EV_1, EV_2, \cdots, EV_N\}$ in the electric carsharing fleet, and n service stations $\{s_1, s_2, \cdots, s_n\}$ are deployed across the city to park these EVs. The capacity of each service station is $\{c(1), c(2), \cdots, c(n)\}$, and there are $\{v(1), v(2), \cdots, v(n)\}$ available shared EVs at each service station in the initial state (e.g., τ_0 in Fig. 7). If the accumulated net flow of service station s_i over the time period T is $F(i) = \{F_{t_1}(i), F_{t_2}(i), \ldots, F_{t_m}(i)\}$, and there are $k(i) = \{k_{t_1}(i), k_{t_2}(i), \ldots, k_{t_m}(i)\}$ shared EVs that have the battery level lower than a threshold resulting in unavailable at this service station over time, then we can find that there are $F_{t_j}(i)+v(i)-k_{t_j}(i)$ available shared EVs at the service station s_i in the jth time slot. If the predicted number of pickups and returns at s_i in the $(j+1)th$ time slot is $\widehat{p}_{t_{j+1}}(i)$ and $\widehat{r}_{t_{j+1}}(i)$, and the estimated number of low-battery shared EVs is $\widehat{k}_{t_{j+1}}(i)$, so the predicted net flow of s_i in $(j+1)th$ time slot is $\widehat{f}_{t_{j+1}}(i) = \widehat{p}_{t_{j+1}}(i) - \widehat{r}_{t_{j+1}}(i)$ (more detailed prediction process will be elaborated in Sec. 4.2). Hence, the extra shared EV demand $\widehat{vd}_{t_{j+1}}(i)$ and parking spot demand $\widehat{pd}_{t_{j+1}}(i)$ of s_i in $(j+1)th$ time slot are

$$\widehat{vd}_{t_{j+1}}(i) = max\{0, -\widehat{f}_{t_{j+1}}(i) - (F_{t_j}(i) + v(i) - k_{t_j}(i))\}, \quad (2)$$

$$\widehat{pd}_{t_{j+1}}(i) = max\{0, \widehat{f}_{t_{j+1}}(i) + (F_{t_j}(i) + v(i)) - c(i)\}. \quad (3)$$

Definition 4.3. We define the **deficiency count** of the service station s_i in the $(j+1)th$ time slot as $dc_{t_{j+1}}(i) = max\{\widehat{vd}_{t_{j+1}}(i), \widehat{pd}_{t_{j+1}}(i)\}$.

For the n service stations and m time slots, we can obtain a $n \times m$ dimensional *station deficiency matrix* $D_{n\times m}$. Each row of the matrix stands for the *deficiency count* of a service station over time. However, if we make a scheduling decision in every time slot, it may cause very high overhead for the fleet management system. Hence, in this paper, we try to set a sequence of dynamic scheduling deadlines, which combine several 5-minute small time slots into an *interval* based on the *deficiency count* of all service stations.

Suppose we finally combine the m time slots into h intervals, e.g., $\{I_1, I_2, \ldots, I_i, \ldots, I_h\} = \{(\tau_0, d_1], (d_1, d_2], \ldots, (d_{i-1}, d_i], \ldots, (d_{h-1}, d_h]\}$, where τ_0 is the start time, each interval could consist of different number of small time slots, and d_i is the ith deadline for shared EV decision making. The basic idea of the dynamic deadline is that the *deficiency count* between two consecutive deadlines should not be too small or too large for all service stations (i.e., under a threshold Ψ), which means that there should be a certain amount of demand for extra shared EVs or parking spots in some service stations before making a management decision and the workers can finish movements in time, and we should satisfy enough supply for shared EVs and parking spots of the next interval I_{j+1} in all service stations by the deadline d_j.

To achieve the above objective, we first combine the maximum entry of each column of $D_{n\times m}$ to obtain a *max deficiency vector* $D_m = [maxD_{:,1}, maxD_{:,2}, \ldots, maxD_{:,m}]$. We first empirically select an initial threshold Ψ based on observations of D_m, then the threshold Ψ will also dynamically update with the operation of the fleet management system.

4.2 Feature Extraction for Prediction

As shown in the above dynamic deadline generation procedure, accurate *net flow* predictions $\widehat{f_t}(i)$ are very important as they directly impact the dynamic deadline generation and the future search process. Hence, in this paper, we conducted a comprehensive feature extraction process to uncover the factors that may impact users' electric carsharing usage based on our data-driven observations (part of them are reported in Sec. 2.3). We extracted five categories of basic features (i.e., temporal features, spatial features, historical usage features, user demographic features, and contextual features) that are highly related to users' usage behaviors to predict the *net flow* of each service station in a small time slot more accurately, which include 12 features in total.

After identifying the related features, we then develop an XGBoost [4]-based model to predict the *net flow* of each service station in each time slot based on the long-term real-world electric carsharing operation data. XGBoost uses a gradient boosting framework and is one of the most effective machine learning models for prediction. Besides, the base model of XGBoost is a decision tree, so it has the potential to show better performance against overfitting and it normally shows the best performance for the problems with small-to-medium structured/tabular data. The predicted *net flow* of the developed model can be represented as

$$\hat{f}_i = \sum_{k=1}^{K} h_k(\mathbf{x}_i), h_k \in \mathcal{H}, \quad (4)$$

where K is the number of trees; $\mathbf{x}_i$ is the i^{th} input, including the five categories of extracted features (12 in total); $\hat{f}_i$ is the corresponding predicted output, which is learned by a tree ensemble model with a collection $\mathcal{H}$ of K functions h_k. Then the objective function at training round t iteration can be denoted as

$$J^{(t)} = \sum_{i=1}^{n}(l(f_i, \hat{f}_i)) + \sum_{k=1}^{t}\Omega(f_k), \quad (5)$$

where $l(.)$ is the loss function (e.g., Square loss); Ω is the regularization term (e.g., $L2$ norm), which measures the model complexity.

4.3 Prediction Error Compensation Mechanism

In practice, it is possible to make inaccurate predictions due to some complicated and unexpected usage behaviors even using advanced prediction methods combined with good feature extraction. Hence, in this paper, we further design an Error Compensation (EC) mechanism to mitigate the influence of inaccurate predictions.

Since the objective of our work is to improve the operational efficiency of the electric carsharing fleet without sacrificing user experience, our system should guarantee users' pickup and return demand by each deadline. However, if the absolute value of the predicted *net flow* is much smaller than the absolute value of the true value in some service stations, it may cause some users' demand cannot be satisfied. Hence, we add a positive prediction compensation term ϵ, which is an adjustable hyperparameter, in Eq. 2 and Eq. 3 to obtain the following Eq. 6 and Eq. 7 to make the system more robust to prediction errors. The ϵ in Eq. 6 is utilized to compensate for the error that the predicted pickup demand smaller than the real pickup demand. The ϵ in Eq. 7 is utilized to compensate for the error that the predicted parking spot demand smaller than the real parking spot demand.

$$\widehat{vd}'_{t_{j+1}}(i) = max\{0, -\widehat{f}_{t_{j+1}}(i) - (F_{t_j}(i) + v(i) - k_{t_j}(i)) + \epsilon\} \quad (6)$$

$$\widehat{pd}'_{t_{j+1}}(i) = max\{0, \widehat{f}_{t_{j+1}}(i) + (F_{t_j}(i) + v(i)) - c(i) + \epsilon\} \quad (7)$$

Based on the new extra demand of shared EVs $\widehat{vd}'_{t_{j+1}}(i)$ and demand of parking spots $\widehat{pd}'_{t_{j+1}}(i)$, we obtain the new station deficiency matrix $D'_{n\times m}$ and max deficiency vector $D_m{}'$ for dynamic deadlines generation as shown in Sec. 4.1.

4.4 DRL-based Decision Making

After determining the deadlines, we focus on the problem of making decisions for unoccupied shared EVs. The goal of the real-time repositioning and charging for fleet management is to improve the *profit* of the operator without sacrificing user experience, which means that there should always be available shared EVs in a service station when users pick shared EVs up, and there should always be unoccupied parking spots when users return shared EVs. Here, we consider the *revenue* from serving users and *costs* for charging and repositioning as shown in Eq. 1.

The goal of the agent is to learn a policy $\pi(a_t|s_t)$ so as to maximize the expected cumulative future rewards $G_t = \sum_{i=0}^{T} \gamma^i R_{t+i+1}$ in an episode T starting from time t. To solve an MDP, a common objective is to learn the value functions, including the state-value function $V_\pi(s)$ and the state-action value function (i.e., Q-function) $Q_\pi(s,a)$, where $V_\pi(s) = \mathbb{E}_\pi[G_t|s_t = s] = \mathbb{E}_\pi\left[\sum_{i=0}^{T} \gamma^i R_{t+i+1}|s_t = s\right]$ and $Q_\pi(s,a) = \mathbb{E}_\pi\left[\sum_{i=0}^{T} \gamma^i R_{t+i+1}|s_t = s, a_t = a\right]$, which measures the expected discounted sum of rewards obtained from state s by taking action a at time t and following policy π.

In this paper, we design a dynamic deadline-based deep reinforcement learning (DRL) method (i.e., Dynamic Deadline-based Distributed Deep Q-Network (D4QN)) to learn the optimal actions for individual shared EVs, i.e., all available shared EVs sequentially learn which action to take between two consecutive deadlines. The advantage of our D4QN lies in its computational efficiency since we streamline our D4QN training and each individual shared EV independently learns its own optimal policy, which ensures scalability at the cost of less coordination between EVs. This is challenging to be achieved by standard DQN or a multi-agent formulation [12, 14].

We define the optimal Q-function for k^{th} shared EV as the maximum expected return achievable by any policy π_t, which is

$$Q^*(s,a) = \max_\pi \mathbb{E}\left[\sum_{i=0}^{T} \gamma^i R^{(k)}_{t+i+1}|s_t{}^{(k)} = s, a_t{}^{(k)} = a, \pi_t\right], \quad (8)$$

which satisfies the Bellman Equation:

$$Q^*(s,a) = \mathbb{E}_{s'}\left[R_{t+1} + \gamma \max_{a'} Q^*(s',a')|s_t{}^{(k)} = s, a_t{}^{(k)} = a\right]. \quad (9)$$

For the electric carsharing fleet management problem, it is challenging to obtain the Q-function as a table containing values due to the large-scale states and actions. Hence, we use deep neural networks to approximate the Q-value function with parameters θ, which makes $Q(s,a;\theta) \approx Q^*(s,a)$. This can be achieved by updating θ_i at each iteration i to minimize the following loss function:

$$\mathcal{L}_i(\theta_i) = \left(R_{t+1} + \gamma \max_{a'} Q(s',a';\theta_i^-) - Q(s,a;\theta_i)\right)^2, \quad (10)$$

which is the squared difference between the target Q values $R_{t+1} + \gamma \max_{a'} Q(s',a';\theta_i^-)$ and the approximate Q values $Q(s,a;\theta_i)$. To make the network updates more stable, we utilize the *experience replay* technique [16], which copies θ to another neural network θ_i^- so that we can fix the Q-value targets temporarily. This forms an input dataset that is stable enough for training. θ_i^- are the parameters from the previous iteration, which are fixed and not updated for learning θ_i. Then we utilize stochastic gradient descent (SGD) with respect to the actual network parameters to minimize this loss. Finally, we obtain the action for each unoccupied shared EV to take.

5 EVALUATION

5.1 Evaluation Data

We utilize 10-month electric carsharing usage data generated by 12,375 unique users in the Chinese city Beijing for evaluation. More than 86,700 usage records are generated by the electric carsharing fleet during this period. In addition to the vehicle usage data, the evaluation dataset also includes the vehicle GPS data, and metadata of 185 service stations and 226 charging stations. The detailed data information has been introduced in Sec. 2.2.

5.2 Evaluation Results

5.2.1 Prediction Performance. We compare the XGBoost-based method with other state-of-the-art prediction approaches, including Historical Average (HA), Auto-Regressive and Moving Average (ARMA), Deep Neural Network (DNN), and Random Forest (RF). As we found the decision making is directly related to the *net flow* of service stations, so we predict the *net flow* (i.e., # of Returns- # of Pickup) of each service station instead of predicting the number of pickups and returns separately.

We adopt the Mean Absolute Error (MAE) to compare the prediction performance of different methods, which is computed as $MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y^{(i)} - \hat{y}^{(i)}\right|$, where $\hat{y}$ is the predicted value, y is the true value, and n is the total number of predictions.

Fig. 8 gives evidence of the advantages of the XGBoost-based approach since its MAE is as low as 0.73, which indicates it achieves very high accuracy for the *net flow* prediction of most service stations in all the time.

We found some other methods like RF also achieves good performance. One possible reason for the high prediction accuracy is our data-driven feature extraction as we extract 5 categories of features that are highly related to users' pickup and return behaviors based on our data-driven observations.

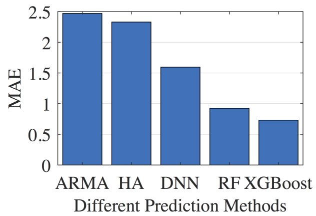

Figure 8: MAE of different prediction methods.

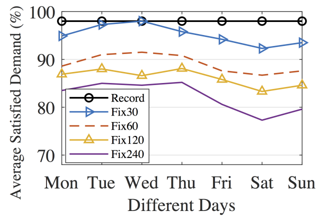

Figure 9: Average % of satisfied user demand.

5.2.2 Dynamic Deadline vs. Fixed Deadline. In this paper, we designed a dynamic deadline mechanism to achieve a win-win performance, i.e., satisfactory *user experience* and low *overhead*. Hence, we compare Record (i.e., dynamic deadline + D4QN) with strategies of different fixed deadlines + D4QN, and we set four typical fixed deadlines for comparisons, i.e., setting a deadline for every 30 minutes (i.e., D4QN + Fix30 or Fix30 for short), 60 minutes (i.e., Fix60), 120 minutes (i.e., Fix120), and 240 minutes (i.e., Fix240).

Fig. 9 shows the average percentage of satisfied user demand (pickup and return) of Record and its variants with different fixed deadlines, which implicitly reflects the user experience, i.e., high satisfied user demand means more satisfaction since more users can have access to the services when they want to pick up or return shared EVs. From Fig. 9, we found our dynamic deadline mechanism can satisfy all user demand on different days. However, with the duration between two consecutive deadlines becomes larger, more users may not be able to have access to available shared EVs or parking spots, resulting in poor user experience.

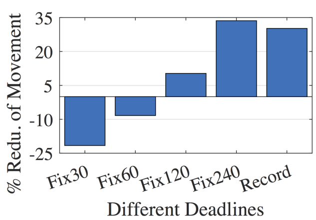

Figure 10: Percentage reduction of movements.

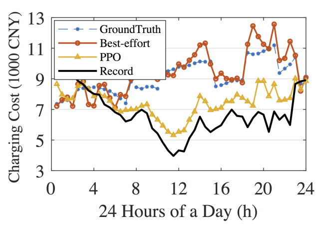

Figure 11: Monthly charging cost distributions.

Fig. 10 shows the percentage reduction of vehicle movement by extra workers of Record and its variants with different fixed deadlines, which is calculated by (Current number of movements by workers - number of movements with different deadline settings)/Current number of movements. From Fig. 10, we found Record reduces 30.2% of movements by workers. With more sparse deadlines, the number of movements by workers becomes less, e.g., Fix240 causes fewer movements than Record, but more demand cannot be satisfied by Fix240 as shown in Fig. 9. With more dense deadlines, the overhead becomes higher due to frequent movements for relocating or charging shared EVs. Since the labor cost is positively correlated with the number of movements, the reduction of movements potentially indicates labor cost reduction and profit increase for the shared EV fleet.

5.2.3 Performance of Different Fleet Management Strategies. To show the effectiveness of our Record, we compare it with (i) Ground Truth (GT), which is extracted from our real-world data; (ii) Best-effort [11], which means once there is vehicle demand or parking demand in a service station, the vehicle repositioning will be performed to satisfy the demand. Once the battery level of a shared EV is below the threshold, it will be scheduled to charge for satisfying future user demand. In this case, the Best-effort can also satisfy user demand; (iii) PPO in [14], which is also a state-of-the-art electric carsharing repositioning algorithm based on DRL.

Since one of the most important factors for electric carsharing fleet management is the operation profit/cost, so we show the charging cost distribution at different times of a day under different management strategies in Fig. 11. We found our Record can reduce charging costs during most time of the day since it can reduce some unnecessary repositioning and leave some shared EVs to charge in the late-night time, during which the charging price is low.

In total, our Record can reduce about 25.8% of charging costs for the electric carsharing fleet. Due to the frequent repositioning and charging activities of Best-effort, it causes higher charging costs in some high charging pricing durations, e.g., 13:00-15:00 and 19:00-20:00, which leads to a 3.9% of charging cost increase. Even though PPO is also based on DRL, it sets periodic deadlines for repositioning, so it also causes higher charging costs.

6 DISCUSSION

6.1 Practical Impacts

- **Electric Carsharing Management.** User experience and profitability of operators are two key factors that impact the electric carsharing promotion. From this work, we found our dynamic deadline-based fleet management system Record can increase the operating profits for electric carsharing operators and satisfy highly dynamic user pickup and return demand, so it has the potential to be reapplied to other cities and enlarge the electric carsharing fleets. In addition, our system can be also easily adopted by other electric carsharing operators due to its generalizability.
- **Fleet Management of Electric Bikesharing, E-Scooter Sharing, and Electric Ridesharing:** Even though the paper focuses on the electric carsharing, we believe our joint real-time repositioning and charging scheduling and the dynamic deadline-based DRL method have the potential to be reapplied to other types of EV fleet management, e.g., electric bikesharing, e-scooter sharing and electric ridesharing. The difference is that one worker can move multiple e-bikes and e-scooters to charge or to other places, and the ridesharing drivers will perform the repositioning and charging by themselves instead of extra workers.

6.2 Milestones Reached

In this project, we have worked with an electric carsharing operator by accessing its data for our data-driven investigation and the dynamic deadline DRL-based fleet management system design. We have presented our design to the fleet management team and obtained its feedback. We have verified our design based on large-scale data from a big city in China to show its potential.

7 RELATED WORK

In this section, we summarize two types of related works, i.e., vehicle repositioning and EV charging recommendation.

7.1 Vehicle Repositioning

Owing to the availability of the rich vehicle location information and operation log data, there is a surge number of work on addressing the unbalanced demand and supply problem by vehicle repositioning for different mobility modes, e.g., taxi [23], ridesharing [12, 21], bikesharing [10, 13], e-scooter sharing [8], and carsharing [9]. Lin et al. [12] designed a management system for ridesharing platforms to maximize the gross merchandise volume of the platforms by relocating available vehicles to the locations with a larger demand-supply gap than the current one. Liu et al. [13] formulated the station-based bikesharing repositioning problem as mixed integer nonlinear programming to minimize the total travel distance.

Different from existing vehicle repositioning works, in this paper, we study an innovative transportation modality, i.e., *electric carsharing*, which has essential differences with existing mobility modes from both the spatial and temporal dimensions. In addition, different fleets have different management modes, so they have different repositioning mechanisms. Moreover, existing vehicle repositioning works rarely consider complicated charging issues and potential impacts of dynamic deadlines.

7.2 EV Charging Recommendation

In the recent decade, there is also an increase of works on EV charging recommendation [16, 18–20], which is also related to our work. Wang et al. [16] designed a charging recommendation system to learn the charging policy for reducing the range anxiety of e-taxi drivers. Dong et al. [5] developed a real-time EV charging recommendation framework for e-taxi fleets, which informs each e-taxi driver at runtime when and where to charge. Wang et al. [19] presented a data-driven fairness-aware charging recommendation method to reduce charging idle time of e-taxi fleets.

Different from these works, our paper focuses on a new type of EV, i.e., electric carsharing. Shared EVs need to be relocated between different service stations to satisfy the highly dynamic and unbalanced spatiotemporal usage distributions. The interactions between service stations and charging stations also make it challenging to manage electric carsharing fleet efficiently compared to only charge other types of EVs. More importantly, we designed a dynamic deadline-based fleet management strategy, which has not been adopted by existing works.

8 CONCLUSION

In this paper, we design a new effective data-driven fleet management system called `Record` jointly considering repositioning and charging location search for electric carsharing with dynamic deadlines to improve overall operating profits without sacrificing user experience. In `Record`, we designed a dynamic deadline-based distributed deep reinforcement learning algorithm D4QN to adaptively satisfy the time-varying unbalanced pickup and return demand. The dynamic deadlines are learned through usage prediction combined with error compensation mechanism. Extensive experimental results show that our designed `Record` effectively reduces 25.8% of charging costs and reduces 30.2% of vehicle movements by workers for an electric carsharing fleet. `Record` also satisfies users' real-time demand and achieves a small runtime overhead.

ACKNOWLEDGMENTS

The authors would like to thank anonymous reviewers for their valuable comments and suggestions. This work is partially supported by NSF 1849238, 1932223, 1952096, 1951890, and 2003874.

REFERENCES

[1] 2016. Future of Carsharing Market to 2025. https://store.frost.com/future-of-carsharing-market-to-2025.html.

[2] Hossein Abouee Mehrizi, Opher Baron, Oded Berman, and David Chen. 2018. Adoption of electric vehicles in car sharing market. *Available at SSRN* (2018).

[3] BAIC BJEV. [n.d.]. Beijing public charging points will implement peak and off-peak electricity prices. http://www.hmc-hongdu.com/news/20.html.

[4] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 785–794.

[5] Zheng Dong, Cong Liu, Yanhua Li, Jie Bao, Yu Gu, and Tian He. 2017. REC: Predictable Charging Scheduling for Electric Taxi Fleets. In *Real-Time Systems Symposium (RTSS), 2017 IEEE*. IEEE, 287–296.

[6] Carl Axel Folkestad, Nora Hansen, Kjetil Fagerholt, Henrik Andersson, and Giovanni Pantuso. 2020. Optimal charging and repositioning of electric vehicles in a free-floating carsharing system. *Computers & Operations Research* 113 (2020).

[7] Claudio Gambella, Enrico Malaguti, Filippo Masini, and Daniele Vigo. 2018. Optimizing relocation operations in electric car-sharing. *Omega* 81 (2018), 234.

[8] Suining He and Kang G Shin. 2020. Dynamic Flow Distribution Prediction for Urban Dockless E-Scooter Sharing Reconfiguration. In *WWW 2020*. 133–143.

[9] Kai Huang, Goncalo Homem de Almeida Correia, and Kun An. 2018. Solving the station-based one-way carsharing network planning problem with relocations and non-linear demand. *Transportation Research Part C* 90 (2018), 1–17.

[10] Pierre Hulot, Daniel Aloise, and Sanjay Dominik Jena. 2018. Towards station-level demand prediction for effective rebalancing in bike-sharing systems. In *ACM SIGKDD*. 378–386.

[11] Ziyang Li, Yiming Zhang, Yunxiang Zhao, Yuxing Peng, and Dongsheng Li. 2016. Best Effort Task Scheduling for Data Parallel Jobs. In *Proceedings of the 2016 ACM SIGCOMM Conference*. 555–556.

[12] Kaixiang Lin, Renyu Zhao, Zhe Xu, and Jiayu Zhou. 2018. Efficient large-scale fleet management via multi-agent deep reinforcement learning. In *24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1774–1783.

[13] Junming Liu, Leilei Sun, Weiwei Chen, and Hui Xiong. 2016. Rebalancing bike sharing systems: A multi-source data smart optimization. In *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1005–1014.

[14] Man Luo, Wenzhe Zhang, Tianyou Song, Kun Li, Hongming Zhu, Bowen Du, and Hongkai Wen. 2020. Rebalancing Expanding EV Sharing Systems with Deep Reinforcement Learning. In *IJCAI*. 1338–1344.

[15] Tianqihoubao. 2020. Historical Weather in Shenzhen. http://www.tianqihou\bao.com/lishi/shenzhen/month/201910.html.

[16] Enshu Wang, Rong Ding, Zhaoxing Yang, Haiming Jin, Chenglin Miao, Lu Su, Fan Zhang, Chunming Qiao, and Xinbing Wang. 2020. Joint Charging and Relocation Recommendation for E-Taxi Drivers via Multi-Agent Mean Field Hierarchical Reinforcement Learning. *IEEE Transactions on Mobile Computing* (2020), 1–17.

[17] Guang Wang, Harsh Rajkumar Vaish, Huijun Sun, Jianjun Wu, Shuai Wang, and Desheng Zhang. 2020. Understanding User Behavior in Car Sharing Services Through The Lens of Mobility: Mixing Qualitative and Quantitative Studies. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 4 (2020), 1–30.

[18] Guang Wang, Xiaoyang Xie, Fan Zhang, Yunhuai Liu, and Desheng Zhang. 2018. bCharge: Data-Driven Real-Time Charging Scheduling for Large-Scale Electric Bus Fleets. In *2018 IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 45–55.

[19] Guang Wang, Yongfeng Zhang, Zhihan Fang, Shuai Wang, Fan Zhang, and Desheng Zhang. 2020. FairCharge: A data-driven fairness-aware charging recommendation system for large-scale electric taxi fleets. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–25.

[20] Guang Wang, Shuxin Zhong, Shuai Wang, Fei Miao, Zheng Dong, and Desheng Zhang. 2021. Data-Driven Fairness-Aware Vehicle Displacement for Large-Scale Electric Taxi Fleets. In *Data Engineering (ICDE), 2021 IEEE 37st International Conference on*. IEEE, 1376–1387.

[21] Zhe Xu, Chang Men, Peng Li, Bicheng Jin, Ge Li, Yue Yang, Chunyang Liu, Ben Wang, and Xiaohu Qie. 2020. When Recommender Systems Meet Fleet Management: Practical Study in Online Driver Repositioning System. In *WWW*.

[22] Jing Yuan, Yu Zheng, and Xing Xie. 2012. Discovering regions of different functions in a city using human mobility and POIs. In *the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 186–194.

[23] Xun Zhou, Huigui Rong, Chang Yang, Qun Zhang, Amin Vahedian Khezerlou, Hui Zheng, M Zubair Shafiq, and Alex X Liu. 2018. Optimizing Taxi Driver Profit Efficiency: A Spatial Network-based Markov Decision Process Approach. *IEEE Transactions on Big Data* (2018), 145–158.

A APPENDIX FOR REPRODUCIBILITY

A.1 Extracted Features

Owing to the large-scale promotion and long-term operation of the electric carsharing fleet, we have accumulated plenty of data to understand the usage patterns of electric carsharing services, which provides us a good opportunity to predict the *net flow* (i.e., user demand). The details of the five categories of extracted features based on our data-driven investigation are shown below:

Temporal Features $\mathcal{F}_{\mathcal{T}}$: From Fig. 2 and Fig. 3, we found the number of pickups and returns are closely related to the time factor. Hence, we extract three temporal features: the time of a day (e.g., we divide one day into a set of time slots $\mathscr{F}_{tod}$, and each time is set to be 5 minutes for a fine-grained prediction), the day of a week $\mathscr{F}_{dow}$, and holiday $\mathscr{F}_{hol}$.

Spatial Features $\mathcal{F}_{\mathcal{S}}$: We found users have different purposes for using shared EVs, which result in different spatial usage patterns, e.g., people who use carsharing for commuting may pick shared EVs in residential areas and then return them in the industrial areas. Hence, we divide the city into seven categories of functional areas based on the method in [22] (i.e., residence, entertainment, business, industry, education, scenery spot, and suburb) to capture the spatial patterns, which forms the functional area feature $\mathscr{F}_{area}$.

Historical Usage Features $\mathcal{F}_{\mathcal{H}}$: Since the electric carsharing usages show a weekly pattern, hence, we utilize our long-term carsharing operation data to capture the historical usage patterns. We extract the *net flow* of each service station in the same time slot of three previous consecutive weeks (i.e., $\mathscr{F}_{his1}$, $\mathscr{F}_{his2}$, $\mathscr{F}_{his3}$) as the historical usage features.

User Demographic Features $\mathcal{F}_{\mathcal{D}}$: We found that the users who use shared EVs in different regions have different demographic features (e.g., more young male users use shared EVs in areas with many IT companies, and more young female users use shared EVs in areas with many financial companies), so they may have different usage patterns due to their job characteristics. Therefore, the user demographic features are also important for the *net flow* prediction. Finally, we extract the percentage of male users and female users as the gender feature $\mathscr{F}_{gender}$ of each service station, and we utilize the users' average age as the age feature $\mathscr{F}_{age}$ of each service station.

Contextual Features $\mathcal{F}_{\mathcal{C}}$: We also found the contextual features like weather conditions have a great impact on users' electric carsharing usage behaviors. Hence, we collect meteorology data from the website [15] and extract features for the net flow prediction. We identify three contextual features: weather $\mathscr{F}_{wea}$, temperature $\mathscr{F}_{tem}$, and wind speed $\mathscr{F}_{wind}$. Among these features, the weather feature is divided into three categories: sunny (or cloudy), rainy, and snowy. The temperature feature has also three types of values: cold (lower than 15 °C), mild (15 -30 °C), hot (over 30 °C). The wind speed is divided into two categories according to the Beaufort number: light (≤ 3) and heavy (> 3).

A.2 Implementation Environment and Parameters Setting

In this project, we are working with an electric carsharing operator in Beijing, who collects the real-world operation data and provides us the data to improve its operational efficiency. Due to the large size of our electric carsharing GPS and order data, it requires significant efforts for efficient management, querying, and processing. Hence, we performed a detailed cleaning process to filter out the error, duplicate, and incomplete order and vehicle GPS data on a high-performance cluster with Spark and Hadoop, which was equipped with 80 TB memory and 20 nodes.

We verify our `Record` including train and test our prediction models and scheduler on a desktop with 32GB memory, 1TB HDD storage, Intel Xeon CPU E5-1660 v3, and a Tesla K40c, installed with the latest Windows 10 and Python coding environment.

For the D4QN-based decision making strategies, we have the following parameter setting: the same three hidden layer Q-network with 128, 64, and 32 nodes from the first to last hidden layer; the activation functions of all hidden units are ReLu, and output layers of the Q-networks use Softmax activation functions. All the experiments are repeated 10 times to ensure the robustness of the results. The batch size of all deep learning networks is set to be 2000, and we utilize AdamOptimizer with a learning rate of 0.001. For the discount factor, we select $\gamma = 0.99$, so the state value is computed within a decaying future horizon.