RL-BLH: Learning-Based Battery Control for Cost Savings and Privacy Preservation for Smart Meters

Jinkyu Koo
Purdue University
West Lafayette, IN 47907
Email: kooj@purdue.edu

Xiaojun Lin
Purdue University
West Lafayette, IN 47907
Email: linx@purdue.edu

Saurabh Bagchi
Purdue University
West Lafayette, IN 47907
Email: sbagchi@purdue.edu

***Abstract*—An emerging solution to privacy issues in smart grids is battery-based load hiding (BLH) that uses a rechargeable battery to decouple the meter readings from user activities. However, existing BLH algorithms have two significant limitations: (1) Most of them focus on flattening high-frequency variation of usage profile only, thereby still revealing a low-frequency shape; (2) Otherwise, they assume to know a statistical model of usage pattern. To overcome these limitations, we propose a new BLH algorithm, named RL-BLH. The RL-BLH hides both low-frequency and high-frequency usage patterns by shaping the meter readings to rectangular pulses. The RL-BLH learns a decision policy for choosing pulse magnitudes on the fly without prior knowledge of usage pattern. The decision policy is designed to charge and discharge the battery in the optimal way to maximize cost savings. We also provide heuristics to shorten learning time and improve cost savings.**

I. Introduction

With smart meters that report fine-grained profiles of energy usage, utility providers can conduct sophisticated power grid management like demand prediction and time-of-use pricing. However, this also threatens user privacy, since adversaries may learn a lot about user's behavior pattern, *e.g.*, when you wake up, and when you go out and come back, from hundreds of data points of meter readings even in a day. With nonintrusive appliance load monitoring (NALM) techniques [1], fine-grained meter readings can also be used to analyze what type of appliance is being in use, by detecting load signatures. This can be used for industrial espionage for example. All of these attacks rely on a man-in-the-middle between the smart meter and the (usually trusted) utility. Such man-in-the-middle attacks are discouragingly easy to mount, especially with embedded devices, due to the weakness of the cryptographic mechanisms and the weakness of the passwords or keys in use [2], [3]. Because of this privacy concern, there are several ongoing lawsuits to stop installing smart meters [4], which delays wide and quick deployment of smart grids.

There have been many efforts to address smart meter's privacy issue in the literature, but the most promising line of work has been on battery-based load hiding (BLH) [5]–[11]. The BLH decouples the meter readings from actual usage profile by charging and discharging a battery at the premises of the end consumer in a strategic manner. The design of how to control the battery is at the heart of a BLH system. The most common approach is to make the meter readings remain as constant as possible, thereby flattening the high-frequency components of usage profile [5]–[7]. This method is effective in hiding load signatures, but does not change much the shape of the envelop of usage profile, *i.e.*, the low-frequency components that provide a clue for user's sleep patterns or times of vacancy. To get an intuitive feel for the high-frequency pattern and low-frequency pattern, consider the case of an industrial manufacturing facility where there are several high load instruments with distinctive energy usage profiles. These patterns can be monitored as part of a high-frequency pattern, and, as has been shown before [8], the appliance in use can be identified through analysis of the signature patterns. On the other hand, the long-term, relatively stable patterns in a household or industrial setting, such as the number of employees in each industrial shift or the occupancy level of a shelter, will show up in low-frequency components. We argue, as have many others in academic [9] and practical (legal) settings [12], that it is important to provide privacy protection for both these kinds of usage patterns.

Another common approach to hiding the smart meter information is to model the contol of meter reading as a function of the usage value by the discrete-state Markov decision process (MDP), assuming that the energy usage are quantized to a finite number of discrete values [9]–[11]. This category of methods can in theory deal with low-frequency usage pattern as well together with high-frequency pattern, since they are trying to decorrelate the meter readings and the usage profile at every possible time instance. However, in practice they raise critical issues. First, they require to know the probability distribution of usage profile, which takes a long time to learn accurately in practice. Second, the quantization assumption leads to a trafeoff between performance and complexity. To get better performance, they need fine-granular quantization, but this directly increases the size of state space because the state space size is typically proportional to $O(LN)$, where L is the number of quantization levels in usage and N is the number of time instances. Huge state space involves heavy computation, which is not acceptable in a small embedded device used to control the battery in BLH systems. Indeed, for this reason, the work in [9] considered only four-level quantization, which is quite coarse.

In addition to improving privacy, an important motivation for using a battery is the potential cost savings. Note that smart grid systems change electricity price depending on time. Such a pricing policy is called *time-of-use* (TOU) pricing. Cost savings will be accomplished by charging the battery when the price is low and using the saved energy from the battery when the price is high. However, privacy protection and cost savings are not always compatible with each other, and thus only a few

works have attempted to achieve these two goals at the same time, with their own limitation. For example, [7] has focused on flattening high frequency components of usage profile in a cost effective way, still revealing low-frequency shape. Another work [9] needs to solve the optimal battery control policy for each pricing zone separately, and thus it is difficult to deal with a complex pricing policy of many zones, or of a price rate that changes dynamically at every moment.

In this paper, we address the aforementioned limitations of existing BLH systems, proposing a new battery control algorithm, named RL-BLH, that takes energy expenditure into account as well as user privacy. The RL-BLH shapes the meter readings to rectangular pulses to hide the high-frequency usage pattern. The pulse magnitudes are mainly determined by not the usage but the amount of energy remaining in a battery, thereby significantly reducing correlation between the meter readings and usage patterns in the low-frequency shape. The decision policy for choosing a pulse magnitude also considers at the same time the goal of maximizing savings in the energy cost by charging the battery when the price of power is low and using the stored energy in the battery when the price of power is high. This is an important consideration because the price of power varies during the day, and the variation tends to be increasing due to the higher penetration of renewable, and fluctuating, energy sources [13]. Consider two forms of TOU pricing in use at American utilities. The first is called real-time pricing (RTP) in which electricity rates vary frequently over the course of the day. Rates change over very short intervals, such as an hour, and the customer receives a unique price signal for each interval, reflecting the costs of generating electricity during that time. In New York, large commercial customers face mandatory hourly pricing while two Illinois utilities have begun to implement RTP for residential customers. The second form is time-zone-based TOU which is the more common of the two. This TOU pricing breaks up the day into two or three large intervals and charges a different price for each. Rates can be divided into off-peak prices (generally during the middle of the night to early morning), semi-peak prices (daytime and evening), and peak prices (occurring during periods of highest demand, usually afternoon/early evening); these rates remain fixed day-to-day over the season. The difference between the off-peak and peak prices can be significant, *e.g.*, it is 25% for Pacific Gas & Electric, the largest utility in California.

Although the above decision policy of when to charge or discharge the battery can be formulated as an MDP, it suffers the issues of requiring usage distribution information and high computational complexity. Thus, the key novelty of our work is that the decision policy is learned on the fly by a reinforcement learning technique.[1] As a result, it does not require prior knowledge of usage statistics. Further, it can efficiently deal with continuous state spaces.

Our contributions in this paper can be summarized as follows:

1) **Privacy protection**: We hide both the low-frequency and high-frequency components of usage profile by shaping the meter readings to rectangular pulses. Specifically, the flattened meter readings within the pulse width remove high frequency variation of usage profile. At the same time, the magnitude of the pulses is designed to vary depending on the battery level, without being directly related with usage profile. Thus, we also reduce the correlation between the meter readings and usage profile in the low-frequency shape. To the best of our knowledge, this work is the first that can hide both high- and low-frequency components without quantizing energy usage.
2) **Learning-based battery control for cost savings**: We learn the decision policy on the fly to determine the optimal magnitude of the rectangular pulses for given battery level and a time index by a reinforcement learning technique, called Q-learning, without assuming to know the usage pattern. The reinforcement learning approach makes RL-BLH resilient to changes in user behavioral pattern since RL-BLH keeps updating the optimal decision for a state in the run-time. In contrast, prior works [9]–[11] have to recompile the whole decision table after building a new stochastic model for the changed behavioral pattern, which takes a long time and requires heavy computations. The resulting decision policy of RL-BLH achieves the optimal cost savings, thereby providing an economic incentive for adopting a battery that is otherwise used only for privacy protection.
3) **Practical considerations for reducing computing cost**: Since the battery level is a continuous variable, the number of possible states for which control decisions need to be learned could be infinite. We address this issue by approximating the optimal action-value function (defined in Section III-B) with a linear combination of a few features that are represented as a function of state variables. Thanks to this, computational complexity is significantly reduced. Further, in order to overcome the long learning time of reinforcement learning, we propose two heuristic methods: generating synthetic data in the run-time and reuse of data in early phases. The former increases the effective amount of data and the latter utilizes the given data better, thereby reducing the learning time significantly and improving the cost savings as well.

Experimental results show that the RL-BLH is comparable to a high-frequency flattening BLH scheme in hiding load signatures, and outperforms it in hiding the low-frequency shape of usage profile. RL-BLH can achieve over 15% of savings in daily energy expenditure with a 5kWh battery, and the savings grow when the battery capacity increases (see Figures 5 and 9). We can also see from an experiment that our heuristics play a significant role in expediting learning. As a result, RL-BLH with our heuristics finishes learning within 10 days in the situation that takes about 1500 days otherwise (see Figure 6).

The rest of this paper is organized as follows. We introduce our system model in Section II and high-level design objectives in Section III. The reinforcement learning framework to maximize the cost savings is presented in Section IV and the heuristics to reduce the learning time are given in Section V. We summarize the proposed algorithm in Section VI. In Section VII, we evaluate the performance of the proposed algorithm in various angles. We discuss some relevant miscellaneous issues in Section VIII. We briefly review related work in Section IX. The paper is concluded in Section X with some possible future work.

[1]This is why we named our algorithm RL-BLH: the 'RL' stands for reinforcement learning.

II. System Model

We consider a smart meter that measures the energy consumption once in every fixed interval (*e.g.*, one minute), which we call a *measurement interval*. Suppose that a day consists of n_M measurement intervals. We denote by x_n the amount of energy consumed by appliances in the n-th measurement interval, where $n = 1, 2, \ldots, n_M$. We call x_n an *usage profile*. Denote the amount of energy that we draw from the power grid in the n-th measurement interval by y_n, which we refer to as a *meter reading*. The smart meter measures the value of y_n and reports it to a utility company. Without any special protection mechanism, the meter reading must be the same as the usage profile, *i.e.*, $y_n = x_n$ for all n.

In order to decouple these usage profile and meter reading, we put a rechargeable battery at the user-end as shown in Figure 1, and select the value of y_n without considering what the value of x_n will be. The battery plays as a buffer between x_n and y_n. Namely, we charge the battery by y_n, and the amount of energy required by appliances, x_n is provided by the battery instead of the power grid. Note that y_n is a control variable, and x_n is determined by user behaviors.

Assume x_n and y_n are continuous variables such that $0 \leq x_n \leq x_M$ and $0 \leq y_n \leq x_M$, *i.e.*, the usage profile is bounded above by x_M and the meter reading is designed (by RL-BLH) to be so as well. Battery level b_n denotes the amount of energy remaining in a battery at the beginning of the n-th measurement interval. Assuming for simplicity that there is no loss when charging and discharging the battery[2], we have

$$b_n = b_{n-1} + y_{n-1} - x_{n-1}. \tag{1}$$

We assume that the capacity of the battery is b_M, *i.e.*,

$$0 \leq b_n \leq b_M \text{ for all } n. \tag{2}$$

A. How to Achieve Cost Savings

We consider time-of-use (TOU) pricing where the electricity price varies from time to time. We denote by r_n the price rate per unit amount of energy in the n-th measurement interval. Cost savings can be achieved by charging the battery when the rate is low and by using the energy stored in the battery when the rate is high.

To better understand the strategy to achieve cost savings, consider a simple example case where there are two price zones: one is a low-price zone ($r_n = r_L$) and the other is a high-price zone ($r_n = r_H$), where $r_H > r_L$. In this case, if we charge b amount of energy in the low-price zone and use it in the high price zone, we will save $(r_H - r_L)b$. Thus, the maximum possible cost savings per day is $(r_H - r_L)b_M$, which is obtained when we charge the battery from empty to full in the low-price zone and use the battery until empty in the high-price zone. Figure 2 illustrates a notional plot of the variation in the charge level of the battery given the above consideration, where a day is assumed to be divided into two equally sized low-price and high-price zones. In other words, the optimal strategy should result in the battery level that is managed to become full at the end of the low-price zone, and be discharged to empty at the end of the high-price zone.

[2]This assumption can be easily relaxed by multiplying loss coefficients to x_{n-1} and y_{n-1} in (1).

In general, the cost savings can be obtained when the original cost for what a user consumes (*i.e.*, $\sum_{n=1}^{n_M} r_n x_n$) is larger than the bill that the user pays to a utility company (*i.e.*, $\sum_{n=1}^{n_M} r_n y_n$). Namely, the cost savings of a day, denoted by S, can be expressed as

$$S = \sum_{n=1}^{n_M} r_n (x_n - y_n). \tag{3}$$

In the earlier example of two price zones, the cost savings is written as

$$S = r_H \sum_{n \in \text{high-price zone}} (x_n - y_n) - r_L \sum_{n \in \text{low-price zone}} (y_n - x_n). \tag{4}$$

Note here that $\sum_{n \in \text{high-price zone}} (x_n - y_n)$ is the amount of energy that we use from the battery in the high-price zone, and $\sum_{n \in \text{low-price zone}} (y_n - x_n)$ is the amount of energy that we charge to the battery in the low-price zone. If those two quantities are equal to b_M, we can achieve the maximum cost savings, which is $(r_H - r_L)b_M$ as mentioned before.

Although we mainly discussed an example of two price zones here, RL-BLH is not limited to such a pricing policy. Rather, RL-BLH is designed to handle the case where r_n is chaining for each and every n (see Section III-B).

III. Key Design Objectives of RL-BLH

Figure 3 shows a typical example of the usage profile. It has been well known that adversaries can identify what appliance is being used by detecting signatures in the usage profile with nonintrusive appliance load monitoring (NALM) techniques [1], [6]. In addition, the low-frequency shape of the usage profile can be used to deduce user's behavioral patterns like when a user goes to bed or when the user goes out for work [9], [12]. Thus, our first design goal is to protect user privacy by hiding both the high and low frequencies usage patterns. Towards this end, we shape the meter readings to rectangular pulses of varying magnitude. The second goal of our design is to maximize the expected savings of a day, *i.e.*, $\mathbb{E}(S)$ so that in addition to privacy protection, customers can benefit from our design economically as well. For this, the rectangular pulse magnitudes are controlled in a way that the battery is charged at the price rate r_n, and the energy stored in the battery is used at the rate $r_{n'}$, where $r_{n'} > r_n$. The following subsections describe our design consideration in more detail.

A. Privacy protection

Prior work in [9] based on a discrete-state MDP approach has indicated that changing the value of y_n in every measurement interval may cause significant correlation between the usage profile and the meter reading, especially between x_{n-1} and y_n. This is because the choice of y_n is inherently constrained by b_n to satisfy the condition in (2), and the battery level b_n, in turn, depends on x_{n-1} from (1). For instance, when $b_n = b_M$, y_n should be zero to avoid overflow in the battery, since x_n can be zero. Thus, the adversaries may be able to exploit this dependence to detect x_{n-1} from y_n and y_{n-1}, with figuring out when the battery is fully charged, *i.e*, knowing the values of b_{n-1} and b_n.

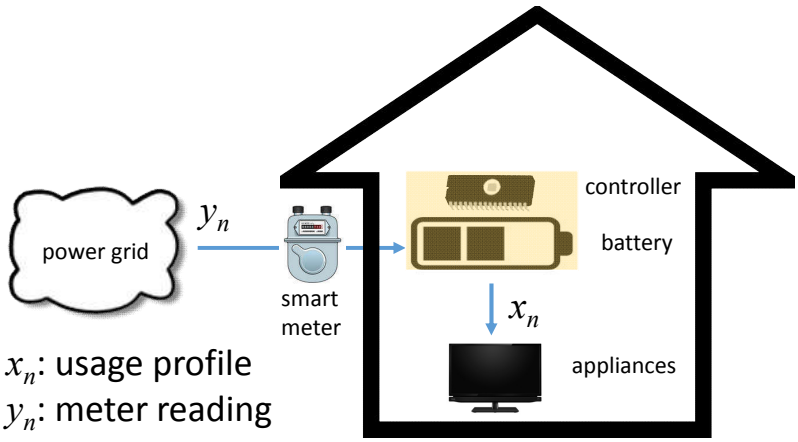


Fig. 1: System model. In the n-th measurement interval, x_n and y_n denotes, respectively, the amount of energy consumed by appliances and the amount of energy that we draw from the power grid. The x_n is supplied by a battery, and the battery is charged by y_n. With the battery acting as a buffer, the value of x_n can be different from the value of y_n.

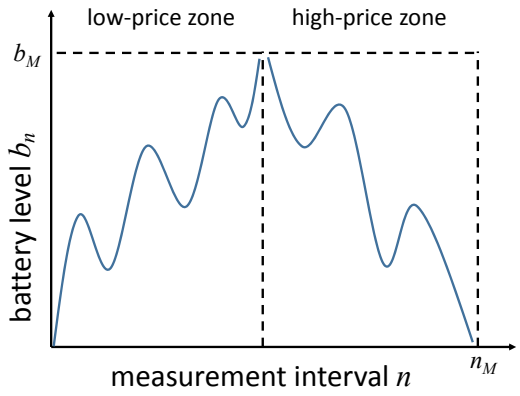


Fig. 2: A typical profile of battery level for a day to achieve the optimal cost savings when there are two price zones. The battery should be fully charged in the low-price zone, and the stored energy in the battery should be fully used in the high-price zone.

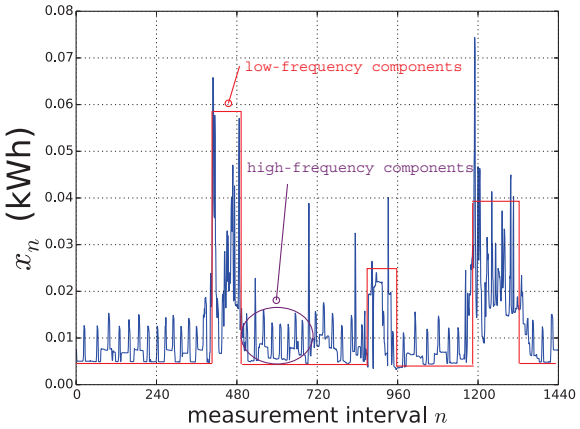


Fig. 3: An example of usage profile. Behavioral patterns can be deduced from the low-frequency components and load signatures can be detected from the high-frequency components.

Our control algorithm below alleviates this problem by changing the value of y_n only once every n_D measurement intervals. This makes the meter readings look like rectangular pulses whose width is n_D measurement intervals. The pulse width, *i.e*, n_D for which the value y_n is held constant is referred to as a *decision interval*. Like flattening high-frequency variation in [5]–[7], keeping y_n constant over n_D measurement intervals reduces the correlation between y_n and x_n around the n-th measurement interval .

We denote by a_k the magnitude of a pulse for the k-th decision interval, which corresponds to measurement intervals $n = (k-1)n_D + 1$ to $n = kn_D$. By design, the value of a_k is allowed to be one out of a_M different choices as follows:

$$a_k = (a-1)x_M/(a_M - 1) \text{ for } a = 1, 2, \ldots, a_M, \quad (5)$$

that is, a_k is one out of evenly spaced a_M different real numbers over an interval $[0, x_M]$. We will still decide the pulse magnitude a_k based on the current battery level $B_k = b_{(k-1)n_D+1}$. However, since

$$B_k = B_{k-1} + \sum_{n=(k-2)n_D+1}^{(k-1)n_D} (a_{k-1} - x_n), \quad (6)$$

we now have more sources of randomness involved in determining the value of B_k, *i.e.*, B_{k-1} and x_n for $n = (k-2)n_D + 1, \ldots, (k-1)n_D$. Thus, from adversary's point of view, it is much harder to know how a_{k-1} is selected.

The value of y_n staying constant as $y_n = a_k$ for the k-th decision interval is effectively similar to flattening high-frequency variation in the meter readings, which was attempted by several existing BLH algorithms. However, our approach can also hide the low-frequency variation in the meter readings

as well, since the choice of a_k varies depending mainly on the battery level B_k, not the shape of the usage profile, *i.e.*, a_k is decided without considering the values of x_n for the k-th decision interval. In other words, the pulse magnitude a_k changes in such a way that it is not directly correlated with the shape of the usage profile. Thus, correlation in the low-frequency shape between the meter readings and usage patterns is significantly reduced.

B. Cost savings

We call deciding the value of a in (5), consequently a_k, an *action*. A *decision policy* π defines what action should be taken for each $k = 1, 2, \ldots, k_M$, where $k_M = n_M/n_D$ with an assumption that n_M is a multiple of n_D. Denoting the cost savings for the chosen a in the k-th decision interval by

$$S_k(a) = \sum_{n=(k-1)n_D+1}^{kn_D} r_n(x_n - a), \tag{7}$$

the maximum cost savings of a day is achieved by finding the optimal policy π^* expressed as

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left(\sum_{k=1}^{k_M} S_k(a)\right). \tag{8}$$

Note that $S_k(a)$ should be negative at some moments so that the battery can be charged, and $S_k(a)$ should be positive at the other moments so that the energy in the battery can be used. To achieve the optimal cost savings, we need to well choose the value of a at each moment so that the overall sum of $S_k(a)$ over a day can be positive and maximized.

In the meantime, after the value of a is chosen, the change in the battery level for the k-th decision interval is $\sum_{n=(k-1)n_D+1}^{kn_D}(a - x_n)$, which can range from $-x_M n_D$ to $x_M n_D$. Therefore, when the battery level is too high or too low, an arbitrary choice of action may violate the battery level constraint in (2) for a corresponding decision interval. For this reason, we restrict $a = 1$ (*i.e.*, $a_k = 0$) when $B_k > b_M - x_M n_D$, thereby guaranteeing that the battery does not overflow, regardless of the values of x_n for the k-th decision interval. Similarly, if $B_k < x_M n_D$, we set $a = a_M$ (*i.e.*, $a_k = x_M$), which prevents energy shortage. When $x_M n_D \le B_k \le b_M - x_M n_D$, any action among a_M possible options can be taken to maximize the cost savings.

The optimal policy π^* can be best described by the following recursive equations, called the Bellman equations [14]:

$$Q^*(k, B_k, a) = \int_{-x_M n_D}^{x_M n_D} P_k(z)\left(S_k(a) + V^*(k+1, B_k + z)\right) dz, \tag{9}$$

where

$$V^*(k, B_k) = \begin{cases} \max_a Q^*(k, B_k, a) & \text{if } 1 \le k \le k_M, \\ 0 & \text{if } k = k_M + 1, \end{cases} \tag{10}$$

and $P_k(z)$ is the probability that the change in battery level for the k-th decision interval is z, *i.e.*, $\sum_{n=(k-1)n_D+1}^{kn_D}(a - x_n) = z$. We call $Q^*(\cdot)$ the optimal action-value function and $V^*(\cdot)$ the optimal value function. The optimal value function $V^*(k, B_k)$ describes the optimal cost savings that can be achieved at the state (k, B_k) assuming the following actions are all optimal until the end of a day. The optimal action-value function $Q^*(k, B_k, a)$ describes that the optimal cost savings that result from the choice of a at the given state (k, B_k) is the sum of $\mathbb{E}(S_k(a))$ and the optimal cost savings that can be achieved from the next state followed. Therefore, the maximum cost savings of a day starting at the battery level B_1 can be expressed as $V^*(1, B_1)$, and the optimal action at a state (k, B_k) defined by π^*, which we denote by $\pi^*(k, B_k)$, can be written as

$$\pi^*(k, B_k) = \arg\max_a Q^*(k, B_k, a). \tag{11}$$

While the above MDP formulation is quite standard, it suffers the two issues that we discussed earlier, *i.e.*, it requires usage distribution information, and fine-granular quantization of the energy usage that leads to a huge state space and high computational complexity. Below, we will introduce a reinforcement learning approach that effectively addresses these two practical issues.

TABLE I: Selected features $f_i(k, B_k)$ for approximating $Q(\cdot)$. Here, $\mathfrak{K} = k/k_M$ and $\mathfrak{B} = B_k/b_M$.

i	0	1	2	3	4	5
$f_i(k, B_k)$	1	$\mathfrak{K}$	$\mathfrak{B}$	$\mathfrak{K}\mathfrak{B}$	$\mathfrak{K}^2$	$\mathfrak{B}^2$

IV. Reinforcement Learning to Maximize Cost Savings

Since we do not assume to know the probability distribution of x_n and consequently $P_k(z)$ in (9), direct computation of $Q^*(k, B_k, a)$ is impossible. Therefore, we will try to learn it by a sample mean $Q(k, B_k, a)$ (precisely speaking, a running average across days) in the following way:

$$\begin{aligned} Q(k, B_k, a) \leftarrow &(1-\alpha) Q(k, B_k, a) \\ &+ \alpha\left(S_k(a) + \max_{a'} Q(k+1, B_{k+1}, a')\right), \end{aligned} \tag{12}$$

where α is called a learning rate. By the law of large numbers, $Q(k, B_k, a)$ can converge to $Q^*(k, B_k, a)$ after days for a sufficiently small value of α [15], [16]. Such learning is referred to as the Q-learning in the reinforcement learning literature.

Note in (12) that B_k is a continuous real variable, and thus explicitly representing $Q(\cdot)$ for each possible (k, B_k, a) is infeasible, *i.e.*, the number of the states to learn is infinite. For this reason, we approximate $Q(\cdot)$ with a function estimator $\hat{Q}(\cdot)$ that is a linear combination of representative features [15], [17]. By experiments[3], we have found that $Q(\cdot)$ can be well approximated by the following form:

$$\hat{Q}(k, B_k, a) = \sum_{i=0}^{5} w_i^{(a)} f_i(k, B_k), \tag{13}$$

[3]Given a state (k, B_k), we have tried all the possible linear combinations of up to second-order terms for k/k_M and B_k/b_M, and picked the combination that results in the maximum cost savings.

where $w_i^{(a)}$ for $i = 0, 1, \ldots, 5$ are weights to find for each a, and the features $f_i(k, B_k)$ are selected as shown in Table I. That is, for each action a, we represent $\hat{Q}(\cdot)$ using six features that are combinations of the normalized battery level, B_k/b_M and the normalized decision interval index, k/k_M. Now what we need learn is the weights $w_i^{(a)}$ for each a, instead of the value of $Q(k, B_k, a)$ for each possible tuple of (k, B_k, a).

On the other hand, if we rewrite (12) as

$$Q(k, B_k, a) \leftarrow Q(k, B_k, a) + \alpha \Delta Q_k^{(a)}, \tag{14}$$

where

$$\Delta Q_k^{(a)} = S_k(a) + \max_{a'} Q(k+1, B_{k+1}, a') - Q(k, B_k, a), \tag{15}$$

we see that the $Q(\cdot)$ update rule in (12) is to move to convergence by reducing the magnitude of $\Delta Q_k^{(a)}$. It has been shown in [17] that applying the same underlying idea as in updating $Q(\cdot)$, a linear estimator in the form of (13) can be found by solving the following:

$$\min_{w_i^{(a)}, \forall i} \mathbb{E}\left(\Delta \hat{Q}_k^{(a)}\right)^2, \tag{16}$$

where

$$\Delta \hat{Q}_k^{(a)} = S_k(a) + \max_{a'} \hat{Q}(k+1, B_{k+1}, a') - \hat{Q}(k, B_k, a). \tag{17}$$

That is, $\hat{Q}(k, B_k, a)$ converges to its optimal by minimizing the magnitude of $\Delta \hat{Q}_k^{(a)}$.

Using the stochastic gradient descent, the weights $w_i^{(a)}$ that are the solution to (16) can be learned by the following iterations:

$$w_i^{(a)} \leftarrow w_i^{(a)} + \alpha \Delta \hat{Q}_k^{(a)} f_i(k, B_k), \forall i. \tag{18}$$

Once the learning is complete, the optimal action in (11) can be re-defined as

$$\pi^*(k, B_k) = \arg\max_a \hat{Q}^*(k, B_k, a). \tag{19}$$

V. Means to Expedite Learning

Since our solution approach is intended to find the optimal policy π^* by online learning, how fast we can learn is one of the important aspects to take into account for practicality. Since $\hat{Q}(\cdot)$ is a linear function, one may try a closed-form formula approach like the least square policy iteration (LSPI) [18], instead of the iterative update in (18). However, we have found that the LSPI does not work well in our case, because it produces a matrix, which can be singular with a high chance.[4] For this reason, rather than the LSPI, we come up with two kinds of heuristic methods to boost up the speed of learning, each of which will be explained in the following subsections.

[4]In our case, the LSPI requires to compute the difference in features between two consecutive states (k, B_k) and $(k+1, B_{k+1})$, which is the same or can be very similar across k. This characteristic reduces the LSPI to an under-determined system of linear equations, and thus leads to poor performance.

A. Generating synthetic data on the fly

In the past decade, supervised learning communities have observed that the performance of a classifier can be significantly improved by generating synthetic data and thus increasing the size of a training data set [19]. Inspired by such an idea, we try to use synthetic data in the context of accelerating the speed of learning. In reinforcement learning, convergence to the optimal policy takes time, which is exactly proportional to the time to collect the enough number of training samples (*i.e.*, x_n in our case). Therefore, we can reduce the wall-clock time to convergence by feeding artificially generated data.

The synthetic training data can be generated in many different ways, *e.g.*, through shifting x_n a little in the time domain, or picking a random number based on the statistics for x_n. In this paper, we take a statistical approach. For each measurement interval n, we track the sample distribution of x_n. Every d_G days, we generate t_G days of artificial usage profiles where x_n is randomly sampled according to the statistical characteristic of the n-th measurement interval. Thus, every d_G days, we apply such synthetic usage profiles to additionally train our $\hat{Q}(\cdot)$ function following (18).

Our experiments shows that the synthetic usage profiles can play an important role in reducing the convergence time with the parameter d_G well chosen to make the synthetic data statistically close to real data in early days. Considering computation load, we limit the use of generating synthetic data within the first d_G^M days.

B. Reuse of data

From (17) and (18), we can see that for the very early phase of learning, the weights $w_i^{(a)}$ (and, in turn, the approximator $\hat{Q}(\cdot)$) are not much different from the initial values, which are arbitrarily given. Although this is an inherent nature of Q-learning, we thought that the data x_n within $S_k(a)$ is not fully utilized, because if the initial weights were more meaningful, we could use the data x_n in a more effective way.

In this sense, we try to reuse the usage profile sequence by training our system multiple times using the same data. That way we can fully utilize each and every data sample even in the early phase of learning. Specifically, until the first d_R days, we store the usage profile sequence of a day, and re-train the system t_R times using the sequence of the day. We can see from experiment results that the reuse of data helps reduce the learning time significantly.

VI. Summary of RL-BLH

The aforementioned pieces of our ideas are summarized in Algorithm 1. The Inner Loop of Algorithm 1 corresponds to the core procedure that determines how to choose the pulse magnitude a_k, *i.e.*, the meter reading y_n for the k-th decision interval and how to learn the optimal policy π^* that maximizes the cost savings. Note in lines 5-10 that we use the ϵ-greedy strategy to handle the 'explore vs. exploit' dilemma: instead of the best action at the current moment, we explore the other available options every once in a while [15]. The Outer Loop of Algorithm 1 describes the iteration of the Inner Loop over days. In line 23, the REUSE mode means that we use the usage profile that is pre-collected at the day (refer to Section V-B).

Algorithm 1 RL-BLH

1: Set $w_i^{(a)} = 0, \forall i$
2: // INNER LOOP describes the loop over measurement intervals within a day.
3: **procedure** INNER LOOP
4: **for** each $k = 1, 2, \ldots, k_M$ **do**
5: Choose a number u randomly over $[0, 1]$.
6: **if** $u < \epsilon$ **then**
7: Choose a randomly among the possible at (k, B_k)
8: **else**
9: $a = \arg\max_{a'} \hat{Q}(k, B_k, a')$.
10: **end if**
11: **for** each $n = (k-1)n_D + 1, \ldots, kn_D$ **do**
12: Set $y_n = (a-1)x_M/(a_M - 1)$.
13: **end for**
14: Update $w_i^{(a)}, \forall i$ by (18).
15: **end for**
16: **end procedure**
17: // OUTER LOOP describes the loop over days.
18: **procedure** OUTER LOOP
19: **for** each day d **do**
20: Execute the INNER LOOP
21: **if** $d <= d_R$ **then**
22: **for** $v = 1, 2, \ldots, t_R$ **do**
23: Execute the INNER LOOP in REUSE mode
24: **end for**
25: **end if**
26: **if** d is a multiple of d_G and $d \leq d_G^M$ **then**
27: **for** $v = 1, 2, \ldots, t_G$ **do**
28: Execute the INNER LOOP in SYN mode
29: **end for**
30: **end if**
31: **end for**
32: **end procedure**

The SYN mode in line 28 implies that the usage profile is synthetically generated according to our description in Section V-A.

VII. EXPERIMENTS

A. Metrics and experiment environment

The load signature can be detected by observing the high-frequency variation of the usage profile, especially by watching two successive values [1], [6]. Thus, we have to measure how well we can hinder the adversary from guessing the length-two sequence of the usage profile $X_n = (x_n, x_{n+1})$ by observing the same length sequence of the meter reading $Y_n = (y_n, y_{n+1})$. As other prior works [6], [9]–[11], we quantify this metric using normalized mutual information (MI) (on average) defined as follows:

$$MI = \frac{1}{n_M - 1} \sum_{n=1}^{n_M - 1} \frac{H(X_n) - H(X_n|Y_n)}{H(X_n)}. \quad (20)$$

Here, $H(\mathcal{X}) = -\sum_i P(\mathcal{X} = i) \log_2 P(\mathcal{X} = i)$ denotes the *uncertainty* of $\mathcal{X}$ in bits. If $H(X_n)$ is z bits, it can be roughly understood in such a way that X_n has 2^z possible realizations, each of which has an equal probability $1/2^z$. Thus, the MI quantifies how much uncertainty about X_n is reduced by observing Y_n on average, normalized to the uncertainty of X_n. The smaller MI means the fewer clue about X_n leaked from Y_n. For example, MI$= 0$ implies that Y_n gives no clue about X_n at all.

The low-frequency shape of the usage profile can tell the adversary that someone is staying home or active, thereby resulting in a high average in usage for hours and vice versa. In order to measure how well we hide such a low frequency shape, we consider the Pearson correlation coefficient (CC) between x_n and y_n defined as

$$CC = \frac{\sum_{n=1}^{n_M}(x_n - \bar{x}) \sum_{n=1}^{n_M}(y_n - \bar{y})}{\sqrt{\sum_{n=1}^{n_M}(x_n - \bar{x})^2} \sqrt{\sum_{n=1}^{n_M}(y_n - \bar{y})^2}} \quad (21)$$

where $\bar{x}$ and $\bar{y}$ are the sample means of x_n and y_n over a day, respectively. The CC is a measure of the linear dependence between x_n and y_n. If x_n and y_n are changing in the same direction, *e.g.*, y_n jumps up when x_n does so, we will have a large value of CC. Thus, the higher value of CC is, the more similar the low-frequency shapes of the usage profile and the meter reading are.

Lastly, as a metric of cost savings, we consider saving ratio (SR) defined as

$$SR = \mathbb{E}\left(\frac{\sum_{n=1}^{n_M} r_n(x_n - y_n)}{\sum_{n=1}^{n_M} r_n x_n}\right). \quad (22)$$

The SR quantifies the expected ratio of the cost savings to the original cost for what the user actually consumes for a day.

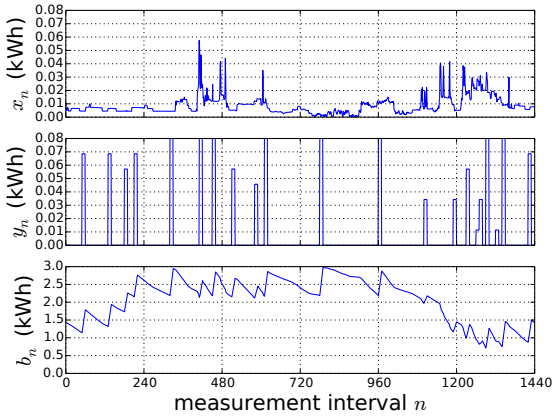
We evaluate RL-BLH using the UMassTraceRepository HomeC model [20], *i.e.*, the usage profile is generated following the statistics of real measurements for the UMassTraceRepository data. The measurement interval is set to one-minute, thereby resulting in $n_M = 1440$. The value of x_M is 0.08 kWh. According to SRP residential time-of-use price plan [21], the electricity rate is set as $r_n = 21.09$ cent per kWh for $n > 1020$, and $r_n = 7.04$ cent per kWh for $n \leq 1020$. With this pricing policy, the UMassTraceRepository model results in the electricity bill that is about 1.65 dollars a day or 50 dollars per month.

We set the hyper-parameters of RL-BLH as $a_M = 8$, $d_G = 10$, $d_G^M = 50$, $t_G = 500$, $d_R = 20$, $t_R = 100$, $\alpha = 0.05$, and $\epsilon = 0.1$. The values of α and ϵ are decreased by a factor of $1/\sqrt{d}$ across days, where d means the number of days.

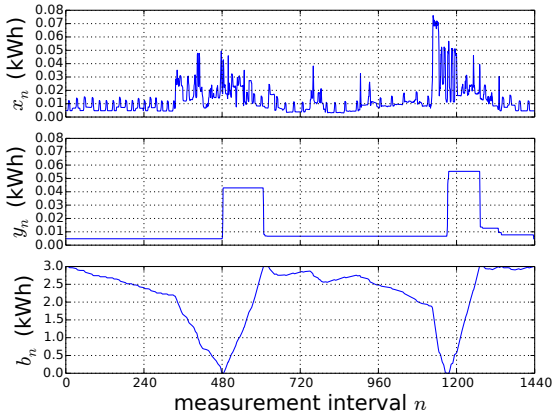
B. Comparison with an existing scheme

In order to show RL-BLH's ability that hides the low-frequency shape of usage profile, we first compare RL-BLH with a representative high-frequency flattening algorithm in [5], which we refer to as a 'low-pass' scheme in this paper.

Figure 4 shows a typical usage profile and meter readings, and corresponding battery level changes. In RL-BLH, the meter readings give almost no idea of what the low-frequency shape of the usage profile looks like. Rectangular pulses of varying magnitude show up aperiodically to charge and discharge the battery as the trend illustrated in Figure 2. In contrast, the low-pass scheme shows a clear correlation in the low-frequency shape between the usage profile and the meter reading. The adversary may figure out that by looking at the bumps in the meter reading around $480 \leq n \leq 600$ and



(a) RL-BLH

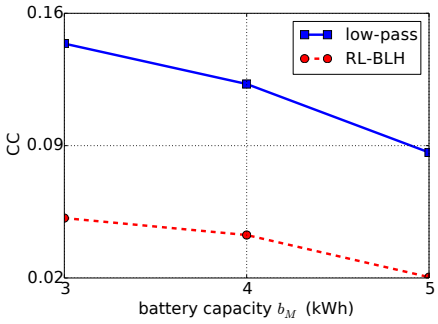


(b) Low-pass

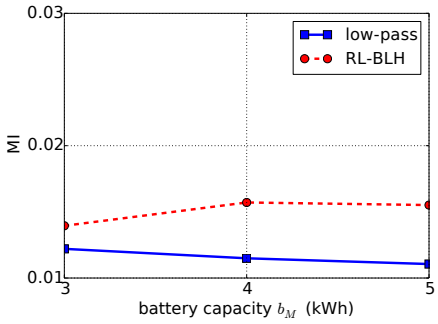
Fig. 4: Typical examples of the RL-BLH and a low-pass schemes for real measurement data when $n_D = 10$ and $b_M = 3$ kWh. The electricity rate here is set as $r_n = 21.09$ cent per kWh for $n > 1020$, and $r_n = 7.04$ cent per kWh for $n \leq 1020$. (4a) The envelope of y_n gives almost no idea of what the envelop of x_n looks like. As intended for cost savings, RL-BLH charges the battery when r_n is low ($n \leq 1020$) and lets the energy in the battery be used when r_n is high ($n > 1020$). (4b) There is a clear correlation in the envelop between x_n and y_n. The adversary may figure out that some activities are happening inside the house by looking at the bumps in the meter reading around $480 \leq n \leq 600$ and $1180 \leq n \leq 1300$.

$1180 \leq n \leq 1300$, some activities are happening inside the house. This kind of information can also be used to figure out when the house is left empty and when users come back home. We can see from Figure 5a that RL-BLH is indeed better in hiding the low-frequency shape than the low-pass scheme by an order of magnitude in the CC.

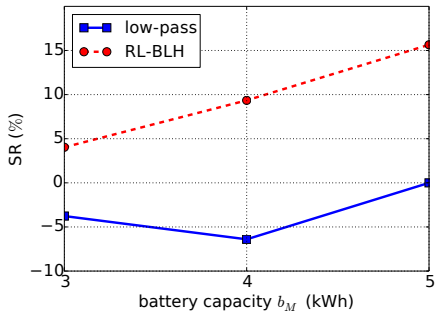
Figure 5b shows that the MI of RL-BLH is slightly higher than that of the low-pass scheme. However, note that at worst, the MI is less than about 0.015, *i.e.*, observing Y_n reduces the uncertainty of X_n by less than 1.5% in RL-BLH, which can be said almost trivial.



(a) Correlation coefficient



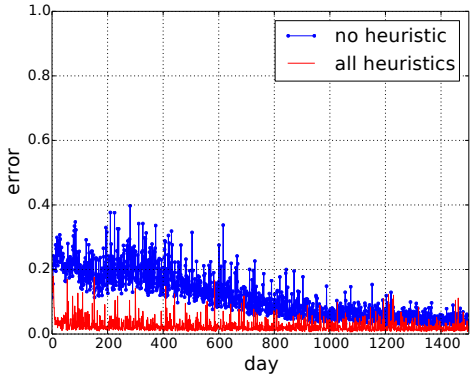
(b) Mutual information



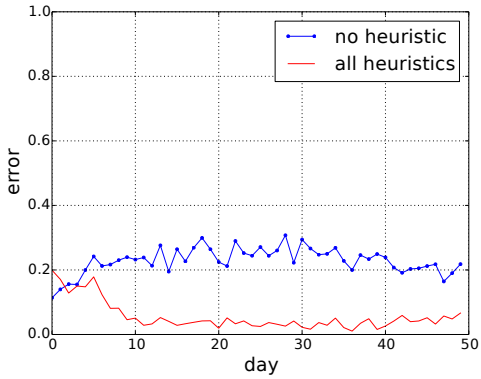
(c) Saving ratio

Fig. 5: Comparison to a low-pass scheme when $n_D = 10$. (5a) RL-BLH is better in hiding the low frequency components than the low-pass scheme by an order of magnitude in the CC. (5b) RL-BLH is comparable to the low-pass scheme in hiding high frequency components. (5c) RL-BLH provides cost savings that increases along with the battery capacity, while the low-pass scheme achieves a random cost savings, which can go negative.

As shown in Figure 4a, RL-BLH charges the battery when the price is low ($n \leq 1020$) and discharges it when the price is high ($n > 1020$), thereby resulting in the cost savings. From Figure 5c, we can see that the RL-BLH can achieve more cost savings by increasing the battery capacity. At $b_M = 5$ (kWh), the SR of RL-BLH is about 15%, which corresponds to 0.25 dollars of cost savings a day or 7.5 dollars a month. As we discussed in Section II-A, the maximum possible cost savings in a day can be expressed as $(r_H - r_L)b_M$, which is 0.7

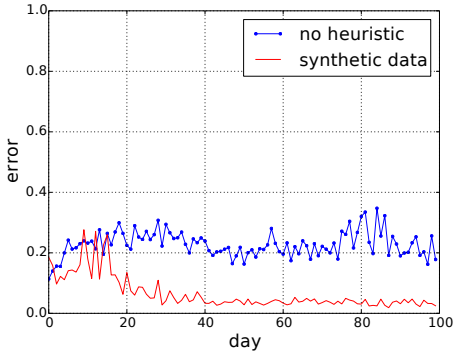


(a) Error plot for the first 1500 days

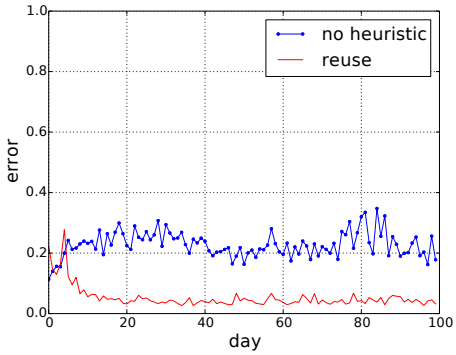


(b) Zoomed-in version of (a)

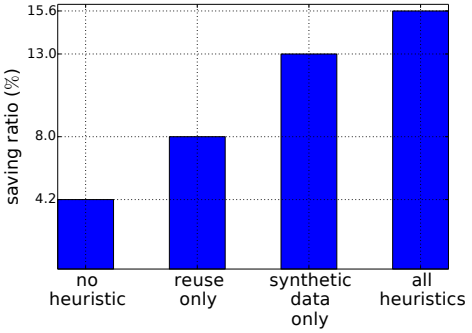
Fig. 6: Effect of all heuristics when $n_D = 15$ and $b_M = 5$ kWh. With our all heuristics, convergence time can be reduced from 1500 days to 10 days.



(a) Synthetic data only



(b) Reuse only



(c) Saving ratio

Fig. 7: Effect of each heuristic when $n_D = 15$ and $b_M = 5$ kWh. Although convergence time and saving ratio can be improved significantly by each heuristic alone, they are improved most when all heuristics are used together at the same time.

dollars in our experiment environment with $b_M = 5$ (kWh). Therefore, what we achieve, although it is optimal for a given situation, is less by 0.45 dollars than the theoretical limit. This is because we are losing the opportunity to achieve additional savings at the cost of privacy protection, *i.e.*, not changing the value of y_n for a decision interval. We will see from Figure 8a that the SR increases when we decrease the length of the decision interval, thereby improving the controllability of a battery level. Meanwhile, it is natural that the cost savings is arbitrary with the low-pass scheme, and it can easily go negative, since there is no consideration for savings in the low-pass scheme.

C. Effects of the heuristics

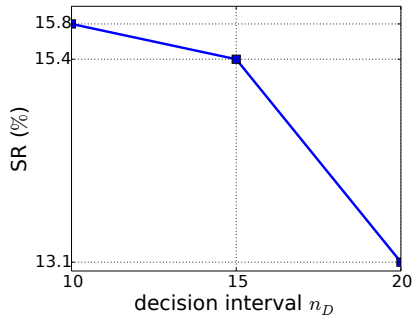
We now show how helpful the heuristics introduced in Section V are to expedite the learning and to improve the cost savings.

To decide the convergence time of RL-BLH, we define the error as the sum of $\Delta\hat{Q}_k^{(a)}$ in (17) over a day, *i.e.*,

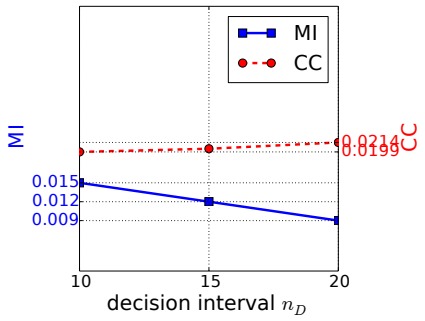
$$error = \sum_{k=1}^{k_M} \Delta\hat{Q}_k^{(a)}, \tag{23}$$

where a is what is chosen for the k-th decision interval by our policy. We can say that the algorithm is converged when the error starts to saturate below.

We can see from Figure 6 that without the heuristics, the convergence takes about 1500 days, which is not practical at all. Meanwhile, applying our heuristics, the algorithm converges within 10 days. Figure 7 shows that the reuse heuristic only can reduce the convergence time to within 10 days, but the SR becomes the largest when all heuristics are used. Thus, we can think that our heuristics are effective in reducing the

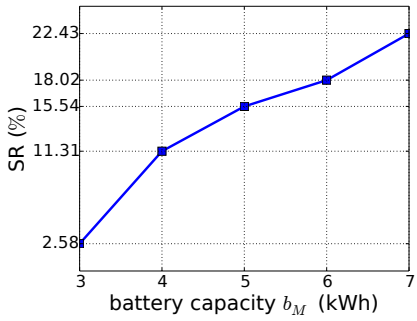


(a) Saving ratio

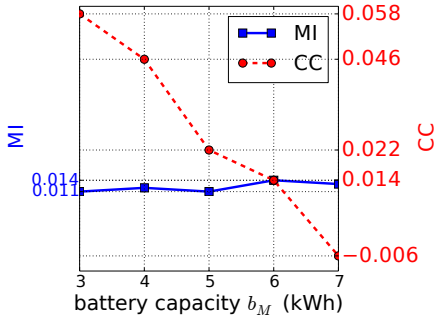


(b) Privacy leakage

Fig. 8: Privacy and cost savings according to n_D when $b_M = 5$ kWh. (8a) When the decision interval n_D increases, the battery level becomes less controllable, and thus cost savings is reduced. (8b) High frequency components can be hidden better with a large value of n_D, while hiding low frequency components is not much affected by n_D.



(a) Saving ratio



(b) Privacy leakage

Fig. 9: Privacy and cost savings according to b_M when $n_D = 15$. (9a) Cost savings increases along with b_M. (9b) Low frequency components can be hidden better with a large value of b_M, while hiding high frequency components is not much affected by b_M.

convergence time as well as improving the cost savings aspect.

D. Effects of the decision interval.

Figure 8 shows how RL-BLH acts when the the decision interval n_D varies. We can see from the figure that the SR goes down when n_D increases, since the larger n_D means the fewer chances to drive the battery level in the direction that we want, *i.e*, a large value of n_D leads to low controllability. The MI also moves in the reverse direction with n_D. This is because the larger n_D is more favorable to hide the high-frequency variation, as the low-pass scheme is slightly better than RL-BLH in Figure 5b. On the other hand, the CC does not seem to change significantly according to n_D. Thus, we can conclude that the decision interval n_D plays a role as a knob to control the tradeoff between privacy protection and cost savings in RL-BLH.

E. Effects of the battery capacity.

Figure 9 shows the performance metric variation according to the battery capacity b_M. The cost savings is proportional to the amount energy that is charged at the low price and used at the high price. Thus, the SR increases when b_M goes up, as we can imagine. We have observed that for a larger b_M, RL-BLH gets more chances to choose $a_k = x_M$, the maximum possible value, when decided to charge. We suspect that this is because the algorithm tries to charge more energy in the battery to achieve the maximal cost savings, and thus causes a_k to have a less correlation with the usage profile. For this reason, the CC shows the inverse relationship with b_M. In the meantime, the MI does not show any linear relationship with b_M. Since the variation is trivial in scale, we think the values are in error range of experiments. Therefore, we can conclude that the larger value of battery capacity b_M is more favorable to RL-BLH for both privacy protection and cost savings. Considering that the battery price is proportional to its capacity, users may decide an appropriate capacity of the battery that meets their requirement for privacy protection and cost savings.

VIII. Discussion

Comparison to [9]: As we mentioned in the Introduction, Markov decision process (MDP) based BLH schemes can also hide both the high-frequency and low-frequency components of usage profile simultaneously. One of such methods [9] can even achieve cost savings as well. Thus, RL-BLH and [9] have similar objectives, *i.e.*, both privacy protection and cost savings. However, [9] used a dynamic programming approach to control a battery under the assumption that the energy usage is quantized to discrete values. Its computation complexity and memory requirement for state space increase quickly along

with the number of quantization levels and the number of time instances. Indeed, [9] stated that the number of state space entries is proportional to $O(LN)$ in a basic version and $O(L^2N + LN^2)$ in an advanced version, where L is the number of quantization levels in usage and N is the number of time instances. In our experimental environment, this corresponds to about 16680960 state entries with $L = 8$ for the advanced version, although $L = 8$ only represents a coarse quantization. For each state entry, [9] calculates the optimal decision. In contrast, what RL-BLH needs to learn is the weights $w_i^{(a)}$ for $i = 0, 1, \ldots, 5$ and $a = 1, 2, \ldots, a_M$. In the same environmental setup, RL-BLH has to deal with only 40 unknowns (with $a_M = 8$), even if it does not need quantization. Thus, RL-BLH can be said to have a huge advantage against [9] in both computation complexity and memory requirement.

Another benefit of RL-BLH is that it can handle the change in user behavioral pattern smoothly, since it keeps updating the weights at every time instance. On the other hand, [9] needs to re-calculate the whole decision table after re-learning the new probability model for usage pattern. In view of the computational complexity in [9], such update would be difficult to be executed at the consumer end, since the BLH controller is expected to be a small embedded system.

Usage patterns changing: The decisions will become sub-optimal if the underlying data differs from future data. However, this issue is common for all MDP approaches due to the inherent difficulty in obtaining the true underlying distribution based on limited data. RL-BLH keeps updating the weights as new data becomes available to alleviate this issue, and hence can be useful even if the user behavioral pattern changes.

Unusual low usage: If there is a day when energy usage is unusually low and deviating from a daily pattern, *e.g.*, no ones are home, the cost savings could go negative in that day. This is because the charged energy at the battery is not fully used at the high-price periods. However, in such a case, the following day can use the energy saved at the battery without charging it again, thereby resulting in higher cost savings than that of a typical day. The SR that we have shown in experiments is an average over days, taking such a case into account.

Battery cost: In our experimental environment, we have shown that a battery of 5kWh can achieve 7.5 dollars of cost savings from 50 dollars bill per month. One may argue that this is relatively small compared to the battery cost. Indeed, the initial cost for the battery ranges from $150 to $200 per kWh currently (although GM expects its battery cell cost hitting $100 per kWh in 2022) [22]. However, note that our first objective is to protect user privacy. What we are doing is to exploit the battery that is anyhow required for privacy protection, and to provide an economical benefit simultaneously in order to encourage privacy-conscious customers to buy our solution. This would be similar to hybrid car's marketing point: environment-conscious consumers buy a hybrid car and save some fuel-cost, although it requires a considerable initial cost due to the battery.

IX. Related Work

There has been extensive research to address the privacy issue in smart grids. One common approach was to modify the meter readings directly in such a way that the gathered data contains some level of uncertainty for sensitive information about individuals. Distortion [23], obfuscation [24], anonymization [25], and aggregation [26] fall into this category of methods. However, modifying the meter readings could lead to inaccurate billing and uncertainty in grid controls (*e.g.*, demand prediction). In addition, this approach requires the existing smart meters to be replaced, which may not be a viable option to utility providers and customers.

Recently, the main stream of research to address the privacy problem has been the battery-based load hiding (BLH) approach. The idea of the BLH is to employ a rechargeable battery at user-ends and feed appliances from the battery so that the meter readings are less correlated with the actual usage. Kalogridis *et. al.* [5] pioneered such a method. They used a battery to flatten high-frequency variation of the load profile in a best-effort manner. Similar ideas were proposed in [6] and [7] where privacy leakage was studied in a more quantitative way. Zhao *et. al.* [8] devises a BLH method that adds a noise to usage profile to assure differential privacy. This group of approaches is effective in hiding load signatures that indicate which appliance is being used. However, there was no consideration for the low-frequency components of usage profile, thereby still revealing important user privacy like sleep patterns or times of vacancy.

Hiding the low-frequency profile of usage as well has been attempted in [9]–[11] in effect. In those works, the real usage is assumed to be quantized to a finite number of discrete levels. The problem is then formulated as a Markov decision process by which meter reading is chosen to be different from the real usage using a battery in a way of minimizing the privacy leakage defined in terms of mutual information. This approach assumes that the underlying state transition model is known, which is unrealistic in most cases, and fails to work when the model changes. In addition, the size of a decision table grows fast according to the granularity of quantization, thus requiring heavy computation that makes such algorithms not suitable to run in small embedded systems.

The rechargeable battery that plays a key role in the BLH provides us with an opportunity to lower the energy bill, by exploiting the time-of-use (TOU) pricing feature of smart grids. This implies that the battery can be charged when the cost of energy is low and discharged to feed the appliances when the cost of energy is high. However, only a few works have considered this aspect in conjunction with privacy protection [7], [9] and they have their own limitations. For example, [7] fails to hide the low-frequency usage pattern, and [9] is difficult to deal with a complex pricing policy in addition to requiring a probability distribution model for the usage profile.

Our work also uses a rechargeable battery to hide the usage profile, but we address all the shortcomings of the previous approaches. First of all, we take the low-frequency usage pattern into account as well as the high-frequency one. Second, we do not assume to know the underlying statistical model of usage profile. Third, the value of energy use is not assumed

to be discrete (*i.e.*, no quantization). Fourth, we provide an energy cost saving framework along with privacy protection. The proposed algorithm learns a decision policy to achieve cost savings on the fly and effectively handle fine-granular TOU pricing with reinforcement learning. Lastly, we provide heuristic methods that reduce the learning time significantly.

X. Conclusion

This paper has studied a new battery-based load hiding (BLH) algorithm that not only addresses the limitations of existing solutions, but also achieves the optimal savings in the energy cost. The proposed BLH algorithm, named RL-BLH, hides both the low-frequency and high-frequency usage patterns by shaping the meter readings to rectangular pulses of varying magnitude. Energy expenditure is reduced in the optimal way that charge a battery when the energy price is low and uses the stored energy in the battery when the price is high. A reinforcement learning technique is applied to learn the decision policy that controls the battery level, without requiring a priori knowledge of usage profile. We approximate the optimal action-value function by a linear combination of a few selected features so that real usage value can be taken into account without quantization, and computation complexity is significantly reduced. The learning time of RL-BLH is significantly shortened by generating synthetic data on the fly and reusing original data at the beginning of learning.

For future work, we are interested in extending our work by integrating renewable energy sources into the picture, where we may produce profit by selling energy in the battery, not just saving the energy cost. Another interesting direction is to further reduce the convergence time of reinforcement learning by enhancing the way of generating synthetic data in the runtime.

Acknowledgment

This material is based in part upon work supported by the National Science Foundation under Grant Numbers CCF-1442726, ECCS-1509536, CNS-1548114, and CNS-1513197 and a contract from Northrop Grumman Corporation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

References

[1] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, Dec 1992.

[2] A. Networks, "Some Perspective on IoT Devices and DDoS Attacks," https://goo.gl/Yprqfk, 2016.

[3] Veracode, "The Internet of Things: Security Research Study," https://goo.gl/D4NOH5, 2016.

[4] Stop Smart Meters, "Smart Meter Lawsuits," http://stopsmartmeters.org/smart-meter-lawsuits, [accessed 28-Nov-2016].

[5] G. Kalogridis, C. Efthymiou, S. Z. Denic, T. A. Lewis, and R. Cepeda, "Privacy for smart meters: Towards undetectable appliance load signatures," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, Oct 2010, pp. 232–237.

[6] W. Yang, N. Li, Y. Qi, W. Qardaji, S. McLaughlin, and P. McDaniel, "Minimizing private data disclosures in the smart grid," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, ser. CCS '12. New York, NY, USA: ACM, 2012, pp. 415–427.

[7] L. Yang, X. Chen, J. Zhang, and H. V. Poor, "Cost-effective and privacy-preserving energy management for smart meters," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 486–495, Jan 2015.

[8] J. Zhao, T. Jung, Y. Wang, and X. Li, "Achieving differential privacy of data disclosure in the smart grid," in *2014 IEEE Conference on Computer Communications, INFOCOM 2014, Toronto, Canada, April 27 - May 2, 2014*, 2014, pp. 504–512.

[9] J. Koo, X. Lin, and S. Bagchi, "Privatus: Wallet-friendly privacy protection for smart meters." in *ESORICS*, ser. Lecture Notes in Computer Science, vol. 7459. Springer, 2012, pp. 343–360.

[10] S. Li, A. Khisti, and A. Mahajan, "Privacy-optimal strategies for smart metering systems with a rechargeable battery," *CoRR*, vol. abs/1510.07170, 2015.

[11] G. Giaconi and D. Gündüz, "Smart meter privacy with renewable energy and a finite capacity battery," *CoRR*, vol. abs/1605.04814, 2016.

[12] K. T. Weaver, "A perspective on how smart meters invade individual privacy," https://takebackyourpower.net/comprehensive-report-how-smart-meters-invade-privacy/, [accessed 28-Nov-2016].

[13] J. Taneja, V. Smith, D. E. Culler, and C. Rosenberg, "A comparative study of high renewables penetration electricity grids," in *IEEE Fourth International Conference on Smart Grid Communications, SmartGridComm 2013, Vancouver, BC, Canada, October 21-24, 2013*, 2013, pp. 49–54.

[14] D. P. Bertsekas and S. E. Shreve, *Stochastic Optimal Control: The Discrete-Time Case*. Athena Scientific, 2007.

[15] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[16] L. P. Kaelbling, M. L. Littman, and A. P. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[17] F. S. Melo, S. P. Meyn, and M. I. Ribeiro, "An analysis of reinforcement learning with function approximation," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 664–671.

[18] M. G. Lagoudakis and R. Parr, "Least-squares policy iteration," *J. Mach. Learn. Res.*, vol. 4, pp. 1107–1149, Dec. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=945365.964290

[19] B. Sapp, A. Saxena, and A. Y. Ng, "A fast data collection and augmentation procedure for object recognition," in *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, ser. AAAI'08. AAAI Press, 2008, pp. 1402–1408.

[20] UMass, "UMassTraceRepository," http://traces.cs.umass.edu/index.php/Smart/Smart, [accessed 28-Nov-2016].

[21] SRP, "SRP Time-of-Use Price Plan," http://www.srpnet.com/prices/home/tou.aspx, [accessed 28-Nov-2016].

[22] Eric Wesoff, "How Soon Can Tesla Get Battery Cell Costs Below $100 per Kilowatt-Hour?" https://www.greentechmedia.com/articles/read/How-Soon-Can-Tesla-Get-Battery-Cell-Cost-Below-100-per-Kilowatt-Hour, [accessed 28-Nov-2016].

[23] L. Sankar, S. R. Rajagopalan, S. Mohajer, and S. Mohajer, "Smart meter privacy: A theoretical framework," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 837–846, June 2013.

[24] Y. Kim, E. C. H. Ngai, and M. B. Srivastava, "Cooperative state estimation for preserving privacy of user behaviors in smart grid," in *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, Oct 2011, pp. 178–183.

[25] C. Efthymiou and G. Kalogridis, "Smart grid privacy via anonymization of smart metering data," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, Oct 2010, pp. 238–243.

[26] J. M. Bohli, C. Sorge, and O. Ugus, "A privacy model for smart metering," in *2010 IEEE International Conference on Communications Workshops*, May 2010, pp. 1–5.