Privacy-Cost Management in Smart Meters Using Deep Reinforcement Learning

Mohammadhadi Shateri, Francisco Messina, Pablo Piantanida, Fabrice Labeau

Privacy-Cost Management in Smart Meters Using Deep Reinforcement Learning

Mohammadhadi Shateri
McGill University
Montreal, Canada

Francisco Messina
McGill University
Montreal, Canada

Pablo Piantanida
CentraleSupélec-CNRS-Université Paris Sud
Gif-sur-Yvette, France

Fabrice Labeau
McGill University
Montreal, Canada

***Abstract*—Smart meters (SMs) play a pivotal rule in the smart grid by being able to report the electricity usage of consumers to the utility provider (UP) almost in real-time. However, this could leak sensitive information about the consumers to the UP or a third-party. Recent works have leveraged the availability of energy storage devices, e.g., a rechargeable battery (RB), in order to provide privacy to the consumers with minimal additional energy cost. In this paper, a privacy-cost management unit (PCMU) is proposed based on a model-free deep reinforcement learning algorithm, called deep double Q-learning (DDQL). Empirical results evaluated on actual SMs data are presented to compare DDQL with the state-of-the-art, i.e., classical Q-learning (CQL). Additionally, the performance of the method is investigated for two concrete cases where attackers aim to infer the actual demand load and the occupancy status of dwellings. Finally, an abstract information-theoretic characterization is provided.**

***Index Terms*—Smart meters privacy, Privacy-cost trade-off, Deep reinforcement learning, Q-learning algorithm, Deep double Q-learning, Privacy-cost management unit.**

I. Introduction

SMART meters (SMs) play an important rule in the modern electricity network, known as the smart grid (SG), by providing fine-grained power consumption measurements of households to the utility provider (UP) almost in a real-time basis. Although these high-resolution SMs measurements can improve the efficiency, reliability, and flexibility of power infrastructures, they might be used by adversaries or malicious third-parties to violate the privacy of households [1]. For instance, by using non-intrusive load monitoring (NILM) approaches, an adversary with access to the power consumption load profiles can readily infer sensitive information related to consumers' daily habits and types of appliances owned.

To protect consumers' sensitive information, two main families of SM data privacy enabling techniques have been proposed in the literature: i) SM data manipulation methods [2]–[4]; and ii) demand shaping methods [5]–[10]. In the first family, the SM data are processed and manipulated, e.g., by adding random noise, before sharing that with the UP. However, this might cause a significant loss in terms of the utility of the SM data [1]. On the other hand, approaches of the second family rely on using physical resources available at the dwelling such as a rechargeable battery (RB), electric vehicles (EVs), heating, ventilation, and air conditioning (HVAC) units, renewable energy source (RES), or a combination of them, to shape the way that energy is consumed by the user from the grid. In this case, the goal is to find an optimum energy management strategy, under the physical limitations of the resources, which provides maximum privacy with minimal household energy expenses.

In [9], the problem of finding an optimum strategy was formulated as a Markov decision process (MDP) and solved using dynamic programming. Concretely, a single-letter expression of the minimum information leakage was provided for the case that demand load is independent and identically distributed (i.i.d.). However, this result is mostly of theoretical value since the demand load is not i.i.d. in practice. In addition, the energy cost was not included in this problem formulation. In another study, the joint optimization of privacy, measured by fluctuations of the grid load around a constant load, and energy cost was considered and modeled as an MDP [8]. To solve this MDP, a classical model-free algorithm from reinforcement learning called Q-learning [11] was used. This tabular algorithm iteratively learns the optimal state-action value function Q^* from which an optimal policy for using the physical resources is readily obtained. Nevertheless, the classical Q-learning (CQL) convergence properties can be poor when the state and/or action spaces are large or infinite.

In this paper, we adopt the MDP formulation introduced in [8] and use a deep learning approximation of the CQL algorithm called deep double Q-learning (DDQL) to overcome the previously discussed limitation of the classical algorithm. To the best of our knowledge, this is the first work that uses deep reinforcement learning to design a privacy-preserving mechanism for smart meters. For the sake of simplicity, we only consider an RB and assume that no energy can be sold to the grid (however, the problem formulation can be easily extended to different scenarios). In addition, the formulation of the cost is revised to take into account not only the total electricity cost but also the cost associated with the wear and tear of the battery. The performance of the DDQL algorithm is assessed using actual SMs data both from practical and theoretical perspectives. For the former evaluation, we consider two scenarios modeling different privacy threats. In the first scenario, an attacker observes the grid load and attempts to infer the consumer's demand load; while in the second scenario an attacker is trained to infer the occupancy status of the household. The latter evaluation is done by studying the trade-off between the cost and the mutual information between the demand and grid loads.

II. Problem Formulation

A. Demand Shaping Using Rechargeable Battery

Fig. 1 shows the privacy-preserving framework for a household based on a single rechargeable battery (RB). Here, the attacker can be the UP itself, an eavesdropper or a third-party. In this framework, a privacy-cost management unit (PCMU) is designed to determine the optimal charging/discharging rate of the battery with the purpose of masking the consumer's demand load while minimizing the total electricity cost.

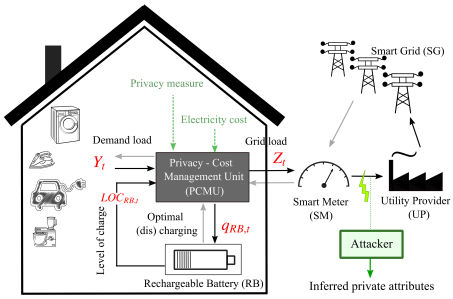

Fig. 1. Privacy-preserving model framework for smart meters based on a rechargeable battery.

Let y_t be the consumer's demand load/power at time t, where $t \in \mathcal{T} = \{1, \dots, T\}$. Given the demand load y_t and the level of charge of battery $\text{LOC}_{RB,t} \in [0, 1]$ at time t, the PCMU should determine the optimum charging/discharging rate $q_{RB,t}$ to hide the consumer's demand load. Therefore, the SM measures and reports the masked demand load (i.e., the grid load) given by

$$z_t = y_t + q_{RB,t}. \tag{1}$$

The PCMU should be able to limit the performance of a potential attacker trying to infer sensitive information (which could be either y_t or a correlated variable) from z_t. In this framework, there is a trade-off between privacy and cost associated to energy and wear and tear of the battery.

B. Markov Decision Process (MDP) Model

The problem of finding an optimal policy for the PCMU (i.e., the reinforcement learning agent in our problem) can be formalized using the framework of Markov decision processes (MDPs) [11], as in [8]–[10]. An MDP is determined by a state space $\mathcal{S}$, which defines the possible states; an action space $\mathcal{A}(s)$ for each $s \in \mathcal{S}$, which describes the feasible actions at state s (we use $\mathcal{A}$ to denote $\cup_{s\in\mathcal{S}} \mathcal{A}(s)$); the environment dynamics $p(s_{t+1}|s_t, a_t)$, which dictates how the current state s_t changes when action a_t is taken; and the reward function $r(s_t, a_t)$ which gives the immediate reward received at state s_t by taking action a_t; and the discount factor $\gamma \in [0, 1]$ which determines the decay rate for weighting future rewards. In this study, we consider a finite horizon time model, i.e. we assume $T < \infty$. Starting from an initial state s_1, and by following a policy $\pi(a|s) = p(a|s)$, the PCMU takes an action a_1, receives a reward r_1, and transitions to state s_2. This leads to generating a sequence of states, actions and rewards: $[s_1, a_1, r_1, \dots, s_T, a_T, r_T]$, which is referred to as an episode. The action-value function $\text{Q} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined as $\text{Q}(s, a) = \mathbb{E}_\pi \left[\sum_{t=1}^{T} \gamma^{t-1} r(s_t, a_t) \middle| s_1 = s, a_1 = a \right]$ and determines how good is to take action $a \in \mathcal{A}(s)$ in state $s \in \mathcal{S}$ and then following the policy π. An optimal policy π^* is one that maximizes $\text{Q}(s, a)$ for all possible s and a. In our problem, the state is defined as $s_t = [\text{LOC}_{RB,t}, y_t]^T$ while the action is defined as the RB charging/discharging rate, i.e. $a_t = q_{RB,t}$, where a positive $q_{RB,t}$ indicates the RB is charging and a negative $q_{RB,t}$ means discharging of RB for supplying the consumer's demand. In addition, the environment transition probability $p(s_{t+1}|s_t, a_t)$ can be written as follows:

$$p(s_{t+1}|s_t, a_t) \overset{\text{(i)}}{=} p\left(\text{LOC}_{RB,t+1}|\text{LOC}_{RB,t}, q_{RB,t}\right) \times p(y_{t+1}|y_t) \tag{2}$$

where (i) is due to the assumption that the consumer's demand load is independent of the action and level of charge of battery. The factor $p\left(\text{LOC}_{RB,t+1}|\text{LOC}_{RB,t}, q_{RB,t}\right)$ is determined by the dynamics and physical constraints of the battery, which can be summarized as follows [8]:

$$q_{\text{RB}}^{\min} \le q_{\text{RB},t} \le q_{\text{RB}}^{\max}, \tag{3}$$

$$\text{LOC}_{\text{RB},t+1} = \text{LOC}_{\text{RB},t} + \frac{q_{\text{RB},t} \times \Delta t \times e_{\text{RB}}}{C_{\text{RB}}}, \tag{4}$$

$$\text{LOC}_{\text{RB}}^{\min} \le \text{LOC}_{\text{RB},t} \le \text{LOC}_{\text{RB}}^{\max}, \tag{5}$$

where $q_{\text{RB}}^{\min}$ and $q_{\text{RB}}^{\max}$ are the minimum and maximum charging rate of the RB, $\text{LOC}_{\text{RB}}^{\min}$ and $\text{LOC}_{\text{RB}}^{\max}$ are the minimum and maximum level of charge of the RB, Δt is the load sampling rate, e_{RB} is the charging efficiency factor of the RB, and C_{RB} is the capacity of the RB. However, note that $p(y_{t+1}|y_t)$ is unknown and in general difficult to approximate [8]. Thus, we do not assume full knowledge of the environment dynamics.

It remains to determine the reward function, which can inversely be interpreted as a loss function: $\ell(s_t, a_t) = -r(s_t, a_t)$. Following [8], the loss function will be defined as a convex combination of a privacy measure term and an electricity cost term. In particular, privacy is defined as the distance between the reported load by the SM and a (predetermined) flat load. Intuitively, when (regardless of the consumer's demand) a constant load is reported by the SM, we expect that no private information about the consumer is leaked. Concretely, considering l_c as the desired constant load, and following policy π, privacy is defined as the average relative distance of the grid load from the target load as follows:

$$F_\pi := \frac{1}{T} \sum_{t=1}^{T} f(s_t, a_t) = \frac{1}{T} \sum_{t=1}^{T} \left| \frac{z_t - l_c}{l_c} \right|. \tag{6}$$

It should be noted that other notions of privacy have also been considered for this problem [1]. On the other hand, the cost that is incurred by the user can be related to the electricity cost and also to the cost associated to the battery wear and tear. In this study, we assume that no energy can be sold to the

grid. Therefore, the average electricity cost over time horizon T is computed as follows:

$$C_\pi = \frac{1}{T}\sum_{t=1}^{T} \Delta t \times h_t \times [z_t]^+, \tag{7}$$

where h_t is the price of 1 kWh of energy purchased from grid at time t and $[x]^+ = \max(x, 0)$. Since $[z_t]^+ = \left[y_t + q_{\mathrm{RB},t}\right]^+ \leq y_t + \left|q_{\mathrm{RB},t}\right|$, and the PCMU has no control on y_t, we conclude that minimizing the average additional cost

$$G_\pi := \frac{1}{T}\sum_{t=1}^{T} g(s_t, a_t) = \frac{1}{T}\sum_{t=1}^{T} \Delta t \times h_t \times \left|q_{\mathrm{RB},t}\right|, \tag{8}$$

minimizes an upper bound of C_π. In addition, notice that G_π takes into account the battery wear and tear cost since it grows when the battery use increases. Finally, considering equations (6) and (8), the one-step loss function is defined as follows:

$$\ell(s_t, a_t) = -r(s_t, a_t) = \lambda g(s_t, a_t) + (1-\lambda) f(s_t, a_t), \tag{9}$$

where $\lambda \in [0, 1]$ controls the privacy-cost trade-off.

III. Methodology

A. Classical Q-Learning (CQL) Algorithm

The Q-Learning (QL) algorithm [11] is a simple method to learn the optimal state-action value function Q^*. Then, the best action at state s is readily obtained by maximizing $\mathrm{Q}^*(s, \cdot)$. In the process of learning, the Q-Learning algorithm takes an action a_t at state s_t using some policy, transitions to state s_{t+1} and updates the Q function as follows [11]:

$$\Delta Q(s_t, a_t) = \alpha\left(r(s_t, a_t) + \gamma \max_{a \in A(s_{t+1})} Q(s_{t+1}, a) - Q(s_t, a_t)\right), \tag{10}$$

where $\alpha \in [0, 1]$ is the learning rate, which should be selected wisely for the sake of convergence [12]. In general, the Q-Learning algorithm uses the $\epsilon-$greedy policy, in which a random action is taken with probability ϵ, and the action which maximizes the Q-value is taken with probability $1-\epsilon$. In this way, while the algorithm exploits the learned Q function, it also explores the action space to discover the benefits of taking other actions (this is known as the exploration-exploration dilemma). Therefore, starting from an initial state and initializing the Q function/table to some arbitrary values, we take an action using the $\epsilon-$greedy policy to move to the next state and update the Q function based on (10). This is continued for all the next states until the end of episode. The process is then repeated for several episodes to ensure all the state-actions are visited enough times.

B. Deep Double Q-Learning (DDQL) Algorithm

One main issue regarding the classical Q-Learning (CQL) algorithm is that it needs to visit all the state-action pairs several times to provide a good approximation of Q^*. Therefore, for large MDPs with many states and actions, convergence is very slow. To resolve this problem, the Q-function can be approximated by using a deep neural network (DNN). These new methods, where deep learning is used for approximating the Q-function, are called deep Q-Learning (DQL) methods [13], [14]. Recently, the DQL method was used for household energy management in [15]. The main idea of the DQL method studied here is to approximate $\mathrm{Q}^*(s, a) \approx \mathrm{Q}(s, a; \theta)$ where θ are the parameters of a DNN called the Q-Network. The Q-Network takes the state $s \in \mathcal{S}$ at the input and generates $\mathrm{Q}(s, a)$ at the output for different actions $a \in \mathcal{A}$. To define the objective function for this Q-network, we observe from (10) that convergence is obtained when the term in parenthesis is equal to zero. The term $r(s_t, a_t) + \gamma \max_{a \in \mathcal{A}(s_{t+1})} \mathrm{Q}(s_{t+1}, a; \theta)$ can be interpreted as the target, while $\mathrm{Q}(s_t, a_t; \theta)$ is the output of the Q-Network. Thus, the mean squared error loss between target and output can be used as the loss function for training the Q-Network. However, using the same network to compute the target and output often leads to instability [16]. To address this issue, the so-called double Q-learning algorithm was proposed in [17] and extended to the deep learning setting in [18]. In the DDQL algorithm, a second network called the Target Network (with parameters θ') is used to calculate the target term. The Target Network parameters θ' are periodically updated by simply copying the parameters from the Q-Network. Thus, using the Target Network, the objective function of the Q-Network can be written as follows:

$$\mathcal{L}_{QN}(\theta) = \mathbb{E}\left[\left(r(s_t, a_t) + \gamma \max_{a \in A(s_{t+1})} Q(s_{t+1}, a; \theta') - Q(s_t, a_t; \theta)\right)^2\right]. \tag{11}$$

The training of the DDQL is presented in Algorithm 1.

Algorithm 1: Training of the deep double Q-learning algorithm. Copy step k and training step k' are hyperparameters.

```
Initialize Q-network and target network.
for number of training episodes do
    Set the initial state s_1 = [LOC_RB,1, y_1].
    for t = 1, ..., T do
        Observe the state s_t = [LOC_RB,t, y_t]
        Select feasible action a_t using ε−greedy algorithm.
        Calculate reward r(s_t, a_t) from equation (9).
        Update the next state s_{t+1} based on (4) and (5).
        Import (s_t, a_t, r(s_t, a_t), s_{t+1}) into the replay buffer.
        Every k' step, update the Q-network by minimizing (11) using
        samples from the replay buffer.
        Every k step, update the target network by copying the
        Q-network parameters.
    end for
end for
```

IV. Results and Discussion

A. Description of data set and parameters

In this study the electricity consumption and occupancy (ECO) data set published by [19] is used. This data set includes 1 Hz electricity usage measured by smart meters and occupancy labels gathered through a tablet computer and a passive infrared sensor. In this study, the data set is sampled with a sampling rate $\Delta t = 15$ min and an episode is defined over a day. Totally 2700 samples (each a vector with length of 96) are used and split into training, validation, and test sets

with ratio 70:10:20, respectively. The validation dataset is used to set the values of the hyperparameters for each algorithm, which are discussed in the next section. The desired constant load is $l_c = 0.7$ kW and the parameters of the battery are as follows: $C_{\text{RB}} = 10$ kWh, $e_{\text{RB}} = 1$, $q_{\text{RB}}^{\max} = -q_{\text{RB}}^{\min} = 4$ kW, $\text{LOC}_{RB}^{\max} = 1$ and $\text{LOC}_{\text{RB}}^{\min} = 0$. We consider the winter Time-of-Use tariff offered by Ontario/Canada, in which the off-peak price is \$0.101 kWh during 19:00 to 7:00, the mid-peak price is \$0.144 kWh during 11:00 to 17:00, and the on-peak price is \$0.208 kWh during 7:00 to 11:00 and during 17:00 to 19:00.

B. *Deep double* Q-*learning versus Classical* Q-*learning*

In this section, the performance of the DDQL method is empirically compared with CQL. For both methods, the network configuration and the value of the hyperparameters are determined empirically by the best privacy-cost trade-off over the validation dataset.

Considering the tabular setting of the CQL algorithm, the state and action states need to be quantized. To this end, a total number of 160 and 100 quantization levels were used for action and demand load, respectively. Therefore, according to the equation (4) and our definition of state s_t the Q function would be a table with size $800 \times 100 \times 160$. A total number of 25K episodes ($\approx$ 2.5M steps) are used with a discount factor $\gamma = 0.8$ and an adaptive learning rate α decreasing linearly from 0.5 to 0.05 over 1M steps.

On the other hand, in the DDQL method, a multilayer perceptron (MLP) with two hidden layers, each including 64 neurons and rectified linear unit (ReLU) as activation function, is used for both the Q and Target Networks. A total number of 800 episodes are used and the discount factor γ is set as 0.99. The size of the experience replay memory is 10K tuples. The memory gets sampled to update the Q-network every 8 steps (k'=8), with minibatches of size 128, and a Target Network copy step k of 500 steps. The RMSProp optimizer with learning rate 0.00025 is selected to train the network.

In the following, the performances of the CQL and DDQL algorithms used to train the PCMU unit are evaluated. Fig. 2 shows how the total episodic reward evolves during the training of each algorithm (for $\lambda = 0$). From this figure, it is clear that, compared with the CQL, the DDQL algorithm leads to higher rewards in a much smaller number of episodes.

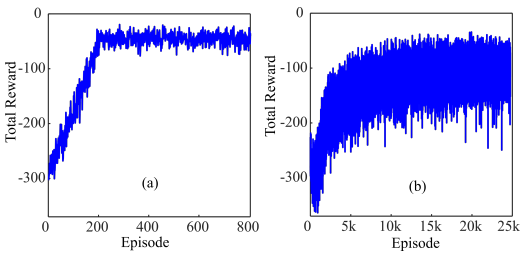

Fig. 2. Total episodic reward for (a) DDQL versus (b) CQL during training.

To compare the performance of these two algorithms on the test dataset, the trade-off between electricity cost in equation (7) and privacy measure in equation (6) are examined for both algorithms in Fig. 3. It can be seen that, although for values of λ close to 1 both CQL and DDQL produce very similar results, for higher privacy levels DDQL significantly outperforms CQL. The reason can be understood by looking at Fig. 2, where the DDQL shows a much better convergence compared with the CQL. In other words, due to the big state and action spaces, the CQL algorithm is not able to converge to the same level (even with a much larger number of episodes) and so provides degraded results compared with DDQL. These results confirm that, in our problem, the DDQL algorithm can provide better results than the classical CQL method.

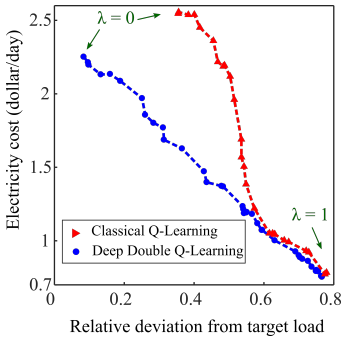

Fig. 3. Average daily electricity cost versus average absolute relative deviation from target load (constant load) on test dataset for $\lambda \in [0, 1]$.

C. *Deep double* Q-*learning versus Attacker*

In this section, we evaluate how DDQL results used in the smart meter privacy-preserving model can reduce attacker inference about user's private attributes. To this end, two scenarios are studied. In the first scenario, an attacker modeled as a neural network with three hidden layers (each with 32 neurons and ReLU activation functions) uses the grid load sequence z^T (with length $T = 96$) to infer the user's demand load y^T. In the second scenario, an attacker modeled as a neural network with two hidden layers (each with 44 neurons and ReLU activation functions) uses the sequences of grid load z^T to infer the occupancy status of households. Both attackers are given a labeled dataset for training (which is a worst-case assumption), and for both of them the RMSProp optimizer with learning rate 0.001 is used. The performances of both attackers are presented in Fig. 4. Notice from the results that the PCMU, using the DDQL algorithm with a RB, can effectively reduce the inference of private information by an attacker by controlling λ. For example, in the second scenario in which privacy does not matter ($\lambda = 1$), the attacker can infer occupancy labels with a balanced accuracy of more than 85%, while for $\lambda = 0$ it is reduced to less than 65%. It should be noted that, by either using a larger battery or other resources, one could in principle approach full privacy (e.g., balanced accuracy of 50%).

Finally, to investigate the overall leakage of information about the consumers' demand load from the grid load, the mutual information (MI) between demand load and grid load

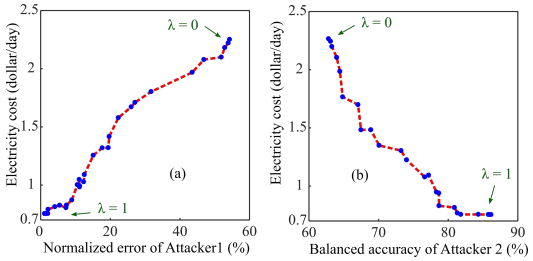

Fig. 4. Attackers performances: (a) inferring actual demand load, (b) inferring occupancy status of household.

is calculated based on the Kraskov–Stögbauer–Grassberger (KSG) estimation method (with 4 neighbors) [20]. It should be noted that KSG is a non-parametric estimation method of the MI and is based on the k-th nearest neighbor. For more details on the KSG the readers is referred to [20]. Results of the privacy-cost trade-off are presented in Fig. 5. This confirms that the DDQL algorithm can reduce the information leakage by incurring a higher energy cost.

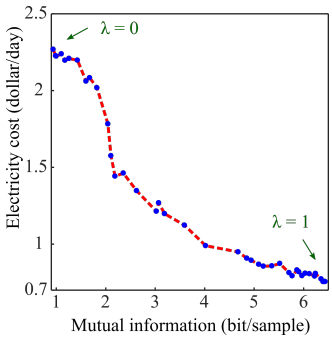

Fig. 5. Electricity cost versus MI between demand load and grid load.

V. Discussion and Concluding Remarks

We examined the privacy-cost trade-off for the SMs system equipped with a rechargeable battery (RB). The optimization problem was formulated as a Markov decisions problem (MDP) where in the reward function the additional cost due to using RB was included while privacy was measured as the average relative distance of the grid load from a constant load. A model-free deep reinforcement learning algorithm, called deep double Q-learning (DDQL) was used to tackle the problem and was compared with state of the art, classical Q-learning (CQL). The results using actual SM data showed that the DDQL algorithm outperforms the CQL method. In addition, the control of privacy attained by the DDQL was evaluated in two concrete case studies and in a general information-theoretic sense. Future work will focus on extending our method to accommodate several resources and studying other privacy measures (e.g. MI) to enhance performance.

Acknowledgment

This work was supported by Hydro-Quebec, the Natural Sciences and Engineering Research Council of Canada, and McGill University in the framework of the NSERC/Hydro-Quebec Industrial Research Chair in Interactive Information Infrastructure for the Power Grid (IRCPJ406021-14). This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 797805.

References

[1] G. Giaconi, D. Gunduz, and H. V. Poor, "Privacy-aware smart metering: Progress and challenges," *IEEE Signal Processing Magazine*, vol. 35, no. 6, pp. 59–78, 2018.

[2] L. Sankar, S. R. Rajagopalan, S. Mohajer, and H. V. Poor, "Smart meter privacy: A theoretical framework," *IEEE Transactions on Smart Grid*, vol. 4, pp. 837–846, June 2013.

[3] H. Yang, L. Cheng, and M. C. Chuah, "Evaluation of utility-privacy trade-offs of data manipulation techniques for smart metering," in *2016 IEEE Conference on Communications and Network Security (CNS)*, pp. 396–400, IEEE, 2016.

[4] M. Shateri, F. Messina, P. Piantanida, and F. Labeau, "Deep directed information-based learning for privacy-preserving smart meter data release," in *2019 IEEE SmartGridComm*, pp. 1–7, IEEE, 2019.

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

[6] J. Gomez-Vilardebo and D. Gündüz, "Smart meter privacy for multiple users in the presence of an alternative energy source," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 132–141, 2014.

[7] G. Giaconi, D. Gündüz, and H. V. Poor, "Smart meter privacy with renewable energy and an energy storage device," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 1, pp. 129–142, 2017.

[8] Y. Sun, L. Lampe, and V. W. Wong, "Smart meter privacy: Exploiting the potential of household energy storage units," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 69–78, 2017.

[9] S. Li, A. Khisti, and A. Mahajan, "Information-theoretic privacy for smart metering systems with a rechargeable battery," *IEEE Transactions on Information Theory*, vol. 64, no. 5, pp. 3679–3695, 2018.

[10] E. Erdemir, P. L. Dragotti, and D. Gündüz, "Privacy-cost trade-off in a smart meter system with a renewable energy source and a rechargeable battery," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019.

[11] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning series, MIT Press, 2018.

[12] C. Szepesvári, "The asymptotic convergence-rate of q-learning," in *Advances in Neural Information Processing Systems*, 1998.

[13] M. Volodymyr, K. Koray, S. David, A. R. Andrei, and V. Joel, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[14] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, J. Pineau, *et al.*, "An introduction to deep reinforcement learning," *Foundations and Trends® in Machine Learning*, vol. 11, no. 3-4, pp. 219–354, 2018.

[15] E. Mocanu, D. C. Mocanu, P. H. Nguyen, A. Liotta, M. E. Webber, M. Gibescu, and J. G. Slootweg, "On-line building energy optimization using deep reinforcement learning," *IEEE Tr. on Smart Grid*, 2018.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[17] H. V. Hasselt, "Double q-learning," in *Advances in Neural Information Processing Systems*, pp. 2613–2621, 2010.

[18] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *30th AAAI conference on artificial intelligence*, 2016.

[19] C. Beckel, W. Kleiminger, R. Cicchetti, T. Staake, and S. Santini, "The eco data set and the performance of non-intrusive load monitoring algorithms," in *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, pp. 80–89, ACM, 2014.

[20] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Physical review E*, vol. 69, no. 6, p. 066138, 2004.