Electric Power Systems Research 189 (2020) 106719

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

Energy resource control via privacy preserving data

Xiao Chen*,[a], Thomas Navidi[b], Ram Rajagopal[a,b]

[a] Civil and Environmental Engineering, Stanford University, United States
[b] Electrical Engineering, Stanford University, United States

ARTICLE INFO

Keywords:
Smart meter
Privacy
Optimization
Battery storage

ABSTRACT

Although the frequent monitoring of smart meters enables granular control over energy resources, it also increases the risk of leakage of private information such as income, home occupancy, and power consumption behavior that can be inferred from the data by an adversary. We propose a method of releasing modified smart meter data so specific private attributes are obscured while the utility of the data for use in an energy resource controller is preserved. The method privatizes data by injecting noise conditioned on the private attribute through a linear filter learned via a minimax optimization. The optimization contains the loss function of a classifier for the private attribute, which we maximize, and the energy resource controller's objective formulated as a canonical form optimization, which we minimize. We perform our experiment on an aggregated dataset of household consumption with solar generation and another from the Commission for Energy Regulation (CER) that contains household smart meter data with sensitive attributes such as income and home occupancy. We demonstrate on the CER data that our method is able to reduce the ability of an adversary to classify a binary income label to that of random guessing while maintaining an objective value for an energy storage controller within 10% of optimal.

1. Introduction

Traditionally, the power grid has been managed by the producers and grid operators with information primarily exchanged among the large asset owners with little feedback from its end users. However, the push for renewable energy sources has brought about the rise of distributed energy resources (DERs) that lie under the control of many smaller and disparate users, causing a paradigm shift in the flow of information. The successful operation of DERs and other smart grid technologies depends on the exchange of large amounts of data from many different end users [1–3]. Due to increased regulations [4], it may be unrealistic to assume data will be available without consideration of the data owners' privacy. The increased granularity of data required for smart grid operation enables the inference of personal information [5] such as household income, which suggests data owners may be reluctant to exchange their data without some effort towards preserving privacy.

Many studies have investigated approaches to protect smart meter data privacy using a number of different techniques and metrics with detailed surveys given in [6–8]. The recent paper from Giaconi et al. [8] defines two general types of approaches, user demand shaping, and data manipulation, with the latter broken into further categories such as data obfuscation or aggregation.

Some papers in the data manipulation category, [9] and [10], perform a pre-processing step on the raw data in order to better prepare it for its end use; however, the pre-processing only considers conditioning the data for its utility without explicitly defining the objective of preserving privacy. Therefore, the pre-processing step may be insufficient to prevent sensitive information from being inferred from the processed data. On the other hand, the aggregation technique presented in [11] and [12], provides user privacy by aggregating data until the aggregate does not reflect on any specific meter data. However, the aggregation group size can be on the order of thousands and there is no consideration to the cost of data utility as a result of aggregation. The data obfuscation category of approaches often come with similar limitations. For example, many studies come from differential privacy (DP) [13], which is widely adopted in designing and analyzing privacy mechanisms in the context of energy data [14–19]. Specifically, studies [14–16] proposed several frameworks for reducing the mutual information between raw data and privatized data (e.g. power profiles), Eibl and Engel [17] investigated the differential privacy effect with some noise injection (e.g. Laplace noise), and Zhou et al. [18] explored how much noise must be added to the data in order to achieve a certain level of differential privacy for an existing Laplace mechanism in the context of solving optimal power flow. Similarly to the aggregation approach and opposite to the pre-processing approaches, these DP approaches

* Corresponding author.
E-mail addresses: markcx@stanford.edu (X. Chen), tnavidi@stanford.edu (T. Navidi), ramr@stanford.edu (R. Rajagopal).

https://doi.org/10.1016/j.epsr.2020.106719
Received 4 October 2019; Received in revised form 18 April 2020; Accepted 2 August 2020
Available online 08 August 2020
0378-7796/

typically only consider obfuscating the data for privacy without simultaneously considering the utility of the data. Therefore, after achieving privacy, the data may be too obfuscated to be useful. One paper that avoids this issue is [19], which proposes a DP mechanism to release the state parameters of power networks with a guarantee of the feasibility of the alternating current (AC) power flow problem. By guaranteeing AC-feasibility of their data, they are making a step in ensuring the data still retains utility after privatization.

The user demand shaping category of approaches often involves a balance between utility and privacy since the privacy is achieved via device operation as opposed to data manipulation [8]; however, achieving privacy through device operation comes with limitations such as its efficacy depends on the physical capabilities of the devices.

We distinguish our studies by developing a methodology that learns an optimal noise injection on the data that balances the trade off between privacy and data utility, thus, preserving as much utility in the data as possible. Our method falls within the data obfuscation category, but differs from strict differential privacy [13] because we use a general notion of privacy that reduces the correlation between private attributes and the data. This general notion of privacy gives us the flexibility to maintain the utility of the data while still eliminating an adversary's ability to recognize certain private attributes. Since many applications of smart meter data involve their use in optimization procedures, we define the utility as the performance achieved when such data is used for optimal control [20]. We consider a scenario where individual owners of DERs, such as battery storage systems, wish to privatize their data before releasing it to a DER aggregator to make optimal control decisions on their behalf, which can have applications in the context of [1–3],. This scenario makes our approach share some similarity to the user demand shaping category of privatization methods in that we provide balance between the utility of DERs and privacy; however, it differs in that our privatization occurs on the data before the operation of the DERs rather than on the actual power consumption after the operation of DERs.

Our work contributes to the research of smart meter privacy in following ways. We propose a minimax approach to generate realistic meter data that is decorrelated from sensitive attributes while maintaining limited performance loss of a cost minimization optimal control algorithm using battery storage. Additionally, we developed a parallelized method that can be easily incorporated in modern deep learning architectures. The correlation of data privatized by our method with sensitive attributes and the performance of a control algorithm is evaluated on two real datasets of residential power demand: one with synthetic sensitive labels and one with real labels. We demonstrate that our method is able to decrease the classification accuracy of an adversary by over 20% while maintaining the performance of the optimization to within 10% over both datasets.

The rest of the paper is organized as follows: we describe the energy resource control in Section 2, control with privatized data generated from the minimax learning algorithm in Section 3, experiments and results on the two datasets in Section 4, and the Conclusion in Section Section 5.

2. Energy resource control

2.1. Notation

We use bold letters for vectors and matrices and regular letters for scalars. Given two vectors $\boldsymbol{x}$ and $\boldsymbol{y} \in \mathbb{R}^n$, $\boldsymbol{x} \geq \boldsymbol{y}$ represents the element-wise order $\boldsymbol{x}(i) \geq \boldsymbol{y}(i)$ for $i \in [n]$ where $[n]$ denotes the set $[n] = \{1, \cdots, n\}$. And $\boldsymbol{x} \geq 0$ means all elements in the vector are not less than the scalar zero. We make the dependence on the underlying probability distribution P when we write expectations (e.g. $\mathbb{E}_P[X]$ where X denotes a random variable). The Frobenius norm of a matrix $\boldsymbol{A}$ is $||\boldsymbol{A}||_F$. We write $\nabla_\theta \mathcal{L}(\theta; X)$ or $\mathrm{d}\mathcal{L}(\theta; X)$, where we typically mean differentiation of the loss function $\mathcal{L}$ with respect to the parameter $\theta \in \mathbb{R}^n$. $\mathcal{N}$ stands for Normal (or Gaussian) distribution and $\mathbb{R}_+$ denotes the non-negative real numbers. We use $:=$ to represent "define as." All the vectors are column vectors by default unless we explicitly address otherwise in a specific context.

2.2. Battery storage control

2.2.1. Control with deterministic demand

Consider a basic battery control problem with the goal of minimizing the energy cost given a prescribed price $\boldsymbol{p} \in \mathbb{R}^H$, where H is the time horizon, typically 24 for an hourly price. An uncontrollable electricity demand is specified as $\boldsymbol{d} \in \mathbb{R}_+^H$. We denote the decision variables for battery control to be $\boldsymbol{x}$ and expand it into $\boldsymbol{x}_{in}, \boldsymbol{x}_{out}, \boldsymbol{x}_s \in \mathbb{R}_+^H$ each of which represents the charging, discharging, and the amount of charge in storage, i.e. $\boldsymbol{x}^\intercal = [\boldsymbol{x}_{in}^\intercal, \boldsymbol{x}_{out}^\intercal, \boldsymbol{x}_s^\intercal]$. The battery optimal control is formulated as follows (Problem 1):

$$\min_{\boldsymbol{x}} \quad \boldsymbol{p}^\intercal(\boldsymbol{x}_{in} - \boldsymbol{x}_{out} + \boldsymbol{d})_+ + \beta_1 ||\boldsymbol{x}_{in}||_2^2 + \beta_2 ||\boldsymbol{x}_{out}||_2^2 + \beta_3 ||\boldsymbol{x}_s - \alpha B||_2^2 \tag{1a}$$

$$\text{s.t.} \quad \boldsymbol{x}_s(j+1) = \boldsymbol{x}_s(j) - \frac{1}{\eta_{out}}\boldsymbol{x}_{out}(j) + \eta_{in}\boldsymbol{x}_{in}(j) \quad \forall\, j \in [H] \tag{1b}$$

$$\boldsymbol{x}_s(1) = B_{init} \tag{1c}$$

$$0 \leq \boldsymbol{x}_{in} \leq c_{in} \tag{1d}$$

$$0 \leq \boldsymbol{x}_{out} \leq c_{out} \tag{1e}$$

$$0 \leq \boldsymbol{x}_s \leq B. \tag{1f}$$

The linear term (with respect to $\boldsymbol{x}$) in the objective is the cost of electricity when there is no value for selling the energy back to the grid. This condition represents a situation where there are no net-metering incentives. The quadratic penalty terms $\beta_1 ||\boldsymbol{x}_{in}||_2^2$ and $\beta_2 ||\boldsymbol{x}_{out}||_2^2$ are added to protect the battery state of health in the horizon [21]. The term $\beta_3 ||\boldsymbol{x}_s - \alpha B||_2^2$ is added to set the battery state to be close to the target value αB with B as the battery size and $\alpha \in (0, 1)$. $\beta_1, \beta_2, \beta_3$ are hyper-parameters to control these penalties. c_{in} and c_{out} are the charging-in and discharging-out power capacities. And the parameter η_{in} and η_{out} denote the charging and discharging efficiency (between 0 and 1). The constraint (1b) indicates that the battery state in the next timestep equals the current battery state adding up the net charging amount (summing up charging and discharging together). Constraint (1c) sets the initial state of the battery to be B_{init}. To simplify the notation, we define a set $\mathcal{X}$: $=\{\boldsymbol{x} | (1b) - (1f) \text{are feasible for some} \boldsymbol{x} \in \mathbb{R}^{3H}\}$. Hence, we use $\boldsymbol{x} \in \mathcal{X}$ to succinctly express that $\boldsymbol{x}$ satisfies the battery constraints. We convert the problem (1) into a canonical convex form in Appendix 6.2 and develop a paralleled algorithm that makes use of automatic differentiation, open-source convex solvers, and pytorch [22]–a popular deep learning framework.

2.2.2. Control with stochastic demand

When determining the control with an uncertain demand, we minimize the expected cost under some demand distribution P. The objective is slightly changed as follows (**Problem** 2):

$$\min \mathcal{L}_u(\boldsymbol{x}, \boldsymbol{d}) \ := \min_{\boldsymbol{x}} \mathbb{E}_{\boldsymbol{d} \sim P}\left[\boldsymbol{p}^\intercal(\boldsymbol{x}_{in} - \boldsymbol{x}_{out} + \boldsymbol{d})_+\right] + \beta_1 ||\boldsymbol{x}_{in}||_2^2 + \beta_2 ||\boldsymbol{x}_{out}||_2^2 + \beta_3 ||\boldsymbol{x}_s - \alpha B||_2^2 \tag{2a}$$

$$\text{s.t.} \quad \boldsymbol{x} \in \mathcal{X}. \tag{2b}$$

Since there is uncertainty behind what the privatized demand will be during training, we use the formulation of the stochastic problem to motivate the minimax problem used for training in Section 3.2. The details behind the training methodology is presented in the following section.

3. Control with privatized demand

Protecting privacy in our context means reducing the correlation between the smart meter data and the sensitive attribute of the data owner, e.g. income or square-footage of the house. We justify why such a consideration of privacy protection is useful in practice in Section 3.1.

3.1. Revealing privacy from data

In this section, we consider a simple scenario in which the sensitive information is a binary label, such as a small or large home, which can be inferred from smart meter data. Given the raw demand $\boldsymbol{d} \in \mathbb{R}_+^H$ and sensitive label $y \in \{0, 1\}$, the adversary builds a classifier f_ψ that takes in demand $\boldsymbol{d}$ to estimate y with a prescribed loss function $\mathcal{L}_a$. Specifically, we assume the adversary minimizes the classification loss

$$\min_{\psi} \mathcal{L}_a\Big(f_\psi(\boldsymbol{d}), y\Big)$$

to infer the private information y. A popular choice of classification loss is cross-entropy loss (or log-loss) [23]. That is

$$\min_{\psi} \Big\{-y \log(f_\psi(\boldsymbol{d})) - (1-y)\log\Big(1 - f_\psi(\boldsymbol{d})\Big)\Big\}$$

when y is a binary variable. The classifier f_ψ is parameterized by ψ and can be a neural network that outputs an estimate of the probability of the positive label. Previous studies [24,25] showed that estimating a sensitive label such as income or square-footage of the house reaches 69% accuracy using features of smart meter data and models like the support vector machine or random forest. We use an alternative neural network model that leverages the daily power consumption (demand) and achieves state-of-the-art accuracy of the private label. More details can be found in Section 4.

3.2. Control with private demand

Our goal is to minimize the cost of energy while incorporating privacy protection. Specifically, we design a *data generator* that creates a perturbed version of the raw demand data in a way that increases the adversarial classification loss, while enabling an optimal controller to minimize the energy cost. From a modeling perspective, we have a minimax problem (**Problem** 3):

$$\min_{\boldsymbol{G}} \mathcal{L}_u\Big(\tilde{\boldsymbol{x}}^*(\tilde{\boldsymbol{d}}), \boldsymbol{d}\Big) - \lambda_a \max_{\psi} \mathcal{L}_a(f_\psi(\tilde{\boldsymbol{d}}), \boldsymbol{y}) \tag{3a}$$

$$\text{s.t.} \quad \tilde{\boldsymbol{d}} = \boldsymbol{d} + \boldsymbol{G}\begin{bmatrix}\boldsymbol{\varepsilon}\\ \boldsymbol{y}\end{bmatrix}, \boldsymbol{\varepsilon} \sim \mathcal{N}(0, \boldsymbol{I}) \tag{3b}$$

$$\tilde{\boldsymbol{x}}^*(\tilde{\boldsymbol{d}}) = \arg\min_{\boldsymbol{x} \in X} \mathcal{L}_u(\boldsymbol{x}, \tilde{\boldsymbol{d}}), \tag{3c}$$

where the parameter $\boldsymbol{G}$ is a matrix that affects the distribution of $\tilde{\boldsymbol{d}}$. In this case, we consider a linear transformation of Gaussian noise $\boldsymbol{\varepsilon}$. Variable $\boldsymbol{y}$ is the one-hot encoding of the sensitive binary label, and f_ψ is a classifier that takes in the perturbed demand data and predicts the corresponding label. The $\mathcal{L}_u$ stands for utility loss. It is important to note that $\mathcal{L}_u$ in the objective uses the raw demand to evaluate the cost of the control decisions determined using the perturbed demand. This represents the case where the storage unit acts on the perturbed information, but the real world value is based on the original raw data.

In order to solve the non-trivial optimization (3), we simplify the constraints (further explained in Section 3.3)) and make use of adversarial training, which is a common technique in studies of generative adversarial networks (GAN) and their applications [26,27]. We add a regularization term $\mathbb{E}\|\tilde{\boldsymbol{d}} - \boldsymbol{d}\|_2^2$ in the objective with an additional hyper-parameter κ,

$$\min_{\boldsymbol{G}} \mathcal{L}_u\Big(\tilde{\boldsymbol{x}}^*(\tilde{\boldsymbol{d}}), \boldsymbol{d}\Big) - \lambda_a \mathcal{L}_a(f(\tilde{\boldsymbol{d}}), \boldsymbol{y}) + \kappa \mathbb{E}\,\|\tilde{\boldsymbol{d}} - \boldsymbol{d}\|_2^2, \tag{4}$$

which helps convergence of the training and preserves parts of the demand that are not related to the privacy or utility loss instead of allowing them to be perturbed arbitrarily.

We denote matrix $\boldsymbol{G} = [\Gamma, \boldsymbol{V}]$ with $\Gamma \in \mathbb{R}^{H \times H}$ and $\boldsymbol{V} \in \mathbb{R}^{H \times 2}$. The altered demand then becomes $\tilde{\boldsymbol{d}} = \boldsymbol{d} + \Gamma\boldsymbol{\varepsilon} + \boldsymbol{V}y$. By denoting π to be the prior distribution of one-hot labels, e.g. $\pi = [p, 1-p]^\intercal$ where p is the prior probability of a positive label, we can rewrite the distortion regularization as

$$\begin{aligned}
&\mathbb{E}(\|\tilde{\boldsymbol{d}} - \boldsymbol{d}\|_2^2) = \mathbb{E}[\|\boldsymbol{d} + \Gamma\boldsymbol{\varepsilon} + \boldsymbol{V}\boldsymbol{y} - \boldsymbol{d}\|_2^2]\\
&= \mathbb{E}[(\Gamma\boldsymbol{\varepsilon} + \boldsymbol{V}\boldsymbol{y})^\intercal(\Gamma\boldsymbol{\varepsilon} + \boldsymbol{V}\boldsymbol{y})]\\
&= \mathbb{E}(\boldsymbol{\varepsilon}^\intercal\Gamma^\intercal\Gamma\boldsymbol{\varepsilon} + \boldsymbol{y}^\intercal\boldsymbol{V}^\intercal\boldsymbol{V}\boldsymbol{y} + \boldsymbol{y}^\intercal\boldsymbol{V}^\intercal\Gamma\boldsymbol{\varepsilon} + \boldsymbol{\varepsilon}^\intercal\Gamma^\intercal\boldsymbol{V}\boldsymbol{y})\\
&\overset{(i)}{=} \mathbb{E}\left[\mathrm{Tr}(\Gamma\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^\intercal\Gamma^\intercal) + \mathrm{Tr}(\boldsymbol{V}\boldsymbol{y}\boldsymbol{y}^\intercal\boldsymbol{V}^\intercal)\right]\\
&\overset{(ii)}{=} \mathrm{Tr}(\Gamma\underbrace{\mathbb{E}[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^\intercal]}_{\boldsymbol{I}}\Gamma^T)\\
&\quad + \mathrm{Tr}\left(\begin{bmatrix}| & |\\ \boldsymbol{v}_1 & \boldsymbol{v}_2\\ | & |\end{bmatrix}\underbrace{\begin{bmatrix}p^2 & p(1-p)\\ p(1-p) & (1-p)^2\end{bmatrix}}_{\mathbb{E}[\boldsymbol{y}\boldsymbol{y}^\intercal]}\begin{bmatrix}- & \boldsymbol{v}_1^\intercal & -\\ - & \boldsymbol{v}_2^\intercal & -\end{bmatrix}\right)\\
&\overset{(iii)}{=} \mathrm{Tr}(\Gamma\Gamma^\intercal) + \|p\boldsymbol{v}_1 + (1-p)\boldsymbol{v}_2\|_2^2\\
&= \|\Gamma\|_F^2 + \|\boldsymbol{V}\pi\|_2^2
\end{aligned} \tag{5}$$

Equality (i) uses the fact that $\boldsymbol{\varepsilon}$ has zero mean. Equality (ii) expands out $\boldsymbol{V}$ as column vectors $[\boldsymbol{v}_1, \boldsymbol{v}_2]$ and expresses $\mathbb{E}[\boldsymbol{y}\boldsymbol{y}^\intercal] = \pi\pi^\intercal = \begin{bmatrix}p\\ 1-p\end{bmatrix}[p \;\; 1-p]$. Rearranging the expressions yields equality (iii).

Therefore, we can equivalently penalize the Frobenius norm of Γ and l_2 norm of the vector $\boldsymbol{V}\pi$, i.e. $\|\Gamma\|_F^2 + \|\boldsymbol{V}\pi\|_2^2$, instead of taking the empirical mean of the demand difference when performing the regularization. To summarize, the data generator determines the filter weight $\boldsymbol{G}$ and outputs the perturbed demand $\tilde{\boldsymbol{d}}$, while the adversary takes in the altered demand $\tilde{\boldsymbol{d}}$ and private labels y to try to learn a classifier.

3.3. Minimax learning

We construct two neural networks to perform the roles of the two players, one is for the data generator and the other one is for the adversary. To train the adversary, we minimize the cross-entropy loss $\mathcal{L}_a$, i.e. $\min_\psi \mathcal{L}_a(f_\psi(\tilde{\boldsymbol{d}}), y)$, which follows the loss function mentioned in Section 3.1. For the generator, we decouple the training into two steps. First, we leverage the loss that is passed from the adversary to update the matrix weight $\boldsymbol{G} = [\Gamma, \boldsymbol{V}]$, i.e.

$$\begin{aligned}
&(\textbf{step1}) \min_{\boldsymbol{G}} -\lambda_a \mathcal{L}_a\Big(f_\psi\Big(\boldsymbol{d} + \Gamma\boldsymbol{\varepsilon} + \boldsymbol{V}\boldsymbol{y}\Big), \boldsymbol{y}\Big)\\
&+ \kappa(\|\Gamma\|_F^2 + \|\boldsymbol{V}\pi\|_2^2)\\
&\overset{(i)}{=} \min_{\boldsymbol{G}=[\Gamma,\boldsymbol{V}]} -\lambda_a \log\Big(1 - f_\psi\Big(\boldsymbol{d} + \Gamma\boldsymbol{\varepsilon} + \boldsymbol{V}\boldsymbol{y}\Big)\Big)\\
&+ \kappa(\|\Gamma\|_F^2 + \|\boldsymbol{V}\pi\|_2^2),
\end{aligned} \tag{6}$$

where κ is the hyper-parameter that penalizes the distance between $\tilde{d}$ and d implicitly. Equality (i) uses the log-loss as the classification loss for the binary label. The next step is to use the privatized demand $\tilde{\boldsymbol{d}} = \boldsymbol{d} + \hat{\boldsymbol{G}}\begin{bmatrix}\varepsilon\\ \boldsymbol{y}\end{bmatrix}$ to determine the control by running the following

optimization:

$$\textbf{(step2)} \arg\min_{\boldsymbol{x}} \mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(0,I)}\{\boldsymbol{p}^{\intercal}(\boldsymbol{x}_{in}-\boldsymbol{x}_{out}+\tilde{\boldsymbol{d}})_{+} + \beta_1\|\boldsymbol{x}_{in}\|_2^2+\beta_2\|\boldsymbol{x}_{out}\|_2^2+\beta_3\|\boldsymbol{x}_s-\alpha B\|_2^2\} \tag{7a}$$

$$\text{s.t.} \quad \boldsymbol{x}\in\mathcal{X}. \tag{7b}$$

The optimal solution of the above convex problem (7) is $\tilde{\boldsymbol{x}}^*$, or more specifically $\tilde{\boldsymbol{x}}^*(\tilde{\boldsymbol{d}})$, because it is a function of the privatized demand, which is aligned with Eq. (3c). The third step calculates the loss, $\mathcal{L}_u(\tilde{\boldsymbol{x}}^*, \boldsymbol{d})$, using $\tilde{\boldsymbol{x}}^*(\tilde{\boldsymbol{d}})$ and the original raw demand expressed as:

$$\textbf{(step3)}\mathcal{L}_u(\tilde{\boldsymbol{x}}^*(\tilde{\boldsymbol{d}}),\boldsymbol{d}) = \boldsymbol{p}^{\intercal}(\tilde{\boldsymbol{x}}_{in}^*(\tilde{\boldsymbol{d}})-\tilde{\boldsymbol{x}}_{out}^*(\tilde{\boldsymbol{d}})+\boldsymbol{d})_{+} + \beta_1\|\tilde{\boldsymbol{x}}_{in}^*\|_2^2+\beta_2\|\tilde{\boldsymbol{x}}_{out}^*\|_2^2+\beta_3\|\tilde{\boldsymbol{x}}_s^*-\alpha B\|_2^2. \tag{8a}$$

We update $\boldsymbol{G}$ using gradient descent with the gradient determined by the chain rule. Recall that the generator outputs a privatized demand with reduced correlation to the sensitive label that is also used to yield the storage control decisions. Those decisions are evaluated on the cost given the raw demand, thus, the Jacobian of $\boldsymbol{G}$ is

$$g_G = \nabla_G\mathcal{L}_u(\tilde{\boldsymbol{x}}^*,\boldsymbol{d}) = \frac{\partial\mathcal{L}_u(\tilde{\boldsymbol{x}}^*,\boldsymbol{d})}{\partial\boldsymbol{x}}\frac{\partial\boldsymbol{x}}{\partial\tilde{\boldsymbol{d}}}\frac{\partial\tilde{\boldsymbol{d}}}{\partial\boldsymbol{G}}. \tag{9}$$

In the context of our storage control problem, the first term in (9) is

$$\frac{\partial\mathcal{L}_u(\boldsymbol{x},\boldsymbol{d})}{\partial\boldsymbol{x}} = \begin{cases}\boldsymbol{Q}\boldsymbol{x}+\begin{bmatrix}\boldsymbol{p}\\-\boldsymbol{p}\\\boldsymbol{0}\end{bmatrix}, & \text{if } \boldsymbol{D}\boldsymbol{x}-\boldsymbol{d}>0\\ \boldsymbol{Q}\boldsymbol{x} & \text{otherwise}\end{cases}, \tag{10}$$

where $\boldsymbol{Q}$ is given in the Appendix Eq. (21), $\boldsymbol{I}$ is the identity matrix, and $\boldsymbol{D}=[\boldsymbol{I}\ \ -\boldsymbol{I}\ \ \boldsymbol{0}]$.

The second term, i.e. $\frac{\partial\boldsymbol{x}}{\partial\tilde{\boldsymbol{d}}}$, in (9) hinges on automatic differentiation through a convex program[28,29]. Because an optimization problem can be viewed as a function mapping the problem data to the primal and dual solutions, we can convert problem (7) to a conic form and calculate the changes of the optimal solution given the perturbations of the problem data. The transformed formulation leverages the idea of finding a zero solution for the residual map of a homogeneous self-dual embedding derived from the KKT conditions of the convex program [29–31].

The third term in (9) is

$$\mathrm{d}\boldsymbol{G}: = \frac{\partial\tilde{\boldsymbol{d}}}{\partial\boldsymbol{G}} = \begin{bmatrix}\frac{\mathrm{d}\tilde{\boldsymbol{d}}}{\varepsilon_1} & \cdots & \frac{\mathrm{d}\tilde{\boldsymbol{d}}}{\varepsilon_H} & \frac{\mathrm{d}\tilde{\boldsymbol{d}}}{p} & \frac{\mathrm{d}\tilde{\boldsymbol{d}}}{1-p}\end{bmatrix}\in\mathbb{R}^{H\times(H+2)}, \tag{11}$$

since $\mathrm{d}\tilde{\boldsymbol{d}} = \mathrm{d}\boldsymbol{G}\begin{bmatrix}\boldsymbol{\varepsilon}\\\boldsymbol{y}\end{bmatrix}$. Thus, all three terms in Eq. (9) can be evaluated in the backward pass of the generator training and we can update the filter weight $\boldsymbol{G}$ using stochastic gradient decent[32]: $\boldsymbol{G}_{k+1}: = \boldsymbol{G}_k - \eta_l g_G$ where k is the iteration step and η_l is the learning rate.

Remark : To summarize, Step 1 shown in Eq. (6) updates the matrix $\boldsymbol{G}$ by minimizing the negative classification loss (equivalent to maximizing the classification loss) of the adversary, while maintaining the constraint determined in (5). Step 2 calculates the optimal control of the storage using the privatized demand. In Step 3, $\boldsymbol{G}$ is updated by evaluating the gradient of the energy cost given the control based on the privatized demand. The updates are expressed as

$$\text{(update1)}\hat{\boldsymbol{G}}_{k+1} = \boldsymbol{G}_k - \eta_l^{(k)}\nabla_G\mathcal{L}_a(f_\psi(\tilde{\boldsymbol{d}}),\boldsymbol{y}) \tag{12a}$$

$$\text{(update2)}\boldsymbol{G}_{k+1} = \hat{\boldsymbol{G}}_{k+1} - \eta_l^{(k)}\nabla_G\mathcal{L}_u(\tilde{\boldsymbol{x}}^*,\boldsymbol{d}) \tag{12b}$$

$$\text{(adversary update)}\psi_{k+1} = \psi_k - \eta_l\nabla_\psi\mathcal{L}_a(f_\psi(\tilde{\boldsymbol{d}}),\boldsymbol{y}), \tag{12c}$$

which run until convergence. We set the learning rates in each step to be equal for simplicity. The training procedure is described in Algorithm.

Algorithm 1: Minimax learning .

Input: Demand data $\mathcal{D}$, label data $\mathcal{Y}$, learning rate η_l, parameters $\{B,\alpha,\beta_1,\beta_2,\beta_3\}$, and hyper-parameters κ_1,κ_2
Initialize $\boldsymbol{G}_k,\psi_k$ at iteration $k=0$ with batch size m;
while *ψ or $\boldsymbol{G}$ has not converged* **do**
1. draw batches of pair $(\boldsymbol{d}^{(i)},\boldsymbol{y}^{(i)})$ from demand and label datasets $(\mathcal{D},\mathcal{Y})$, $\forall i=1,\ldots,m$;
2. Sample batch of Gaussian random vectors $\boldsymbol{\varepsilon}^{(1),\ldots,(m)}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})$;
3. $\psi_{k+1} := \psi_k - \eta_l\mathbb{E}[\nabla_\psi\mathcal{L}_a(f_\psi(\tilde{\boldsymbol{d}}),\boldsymbol{y})]$;
4. $\widehat{\boldsymbol{G}}_{k+1} := \boldsymbol{G}_k - \eta_l\mathbb{E}[\nabla_G\mathcal{L}_a(f_\psi(\tilde{\boldsymbol{d}}),\boldsymbol{y})]$;
5. $\boldsymbol{G}_{k+1} := \widehat{\boldsymbol{G}}_{k+1} - \eta_l\mathbb{E}[\nabla_G\mathcal{L}_u(\tilde{\boldsymbol{x}}^*,\boldsymbol{d})]$ where $\tilde{\boldsymbol{x}}^*$ is optimal solution of (7)
(The expected gradient value is approximated as the sample mean of the batch.)

return $\boldsymbol{G}$ and ψ

3.4. Convergence of the filter

This subsection focuses on the stability and boundedness of the iterates in our back-propagation that leverage stochastic gradient methods (or some related variants of first-order gradient methods). Using the subgradient property [33, Chapter 9.1], g is a subgradient of f at x if

$$f(y)\geq f(x)+\langle g,y-x\rangle \quad \forall\, y, \tag{13}$$

and assuming $\boldsymbol{G}^*$ is a local optimal point; when we apply the step1 and step3 updates $\boldsymbol{G}_{k+1}=\boldsymbol{G}_k-\eta_l^{(k)}\nabla\mathcal{L}_a^{(k)}-\eta_l^{(k)}\mathcal{L}_u^{(k)}$ at the k-th iteration, we can obtain the following relationship

$$\mathbb{E}[\|\boldsymbol{G}_{k+1}-\boldsymbol{G}^*\|_2^2] \tag{14a}$$

$$= \mathbb{E}[\|\boldsymbol{G}_k-\eta_l^{(k)}(\nabla\mathcal{L}_a^{(k)}+\nabla\mathcal{L}_u^{(k)})-\boldsymbol{G}^*\|_2^2] \tag{14b}$$

$$= \mathbb{E}[\|\boldsymbol{G}_k-\boldsymbol{G}^*\|_2^2]-2\eta_l^{(k)}\mathbb{E}\langle\nabla\mathcal{L}_a^{(k)}+\nabla\mathcal{L}_u^{(k)},\boldsymbol{G}_k-\boldsymbol{G}^*\rangle + (\eta_l^{(k)})^2\underbrace{\|\nabla\mathcal{L}_a^{(k)}+\nabla\mathcal{L}_u^{(k)}\|_2^2}_{\delta_k^2} \tag{14c}$$

$$\overset{(i)}{=}\mathbb{E}[\|\boldsymbol{G}_k-\boldsymbol{G}^*\|_2^2]-2\eta_l^{(k)}\mathbb{E}\langle\nabla\mathcal{L}_a^{(k)},\boldsymbol{G}_k-\boldsymbol{G}^*\rangle - 2\eta_l^{(k)}\mathbb{E}\langle\nabla\mathcal{L}_u^{(k)},\boldsymbol{G}_k-\boldsymbol{G}^*\rangle+(\eta_l^{(k)})^2\delta_k^2 \tag{14d}$$

$$\overset{(ii)}{\leq}\mathbb{E}[\|\boldsymbol{G}_k-\boldsymbol{G}^*\|_2^2]-2\eta_l^{(k)}\left(\mathcal{L}_a(\boldsymbol{G}_k)-\mathcal{L}_a^*\right) - 2\eta_l^{(k)}(\mathcal{L}_u(\boldsymbol{G}_k)-\mathcal{L}_u^*)+(\eta_l^{(k)})^2\delta_k^2. \tag{14e}$$

Equality (i) expands the inner product of the loss gradients and iterates using δ_k for the norm of the sum of loss gradients. The inequality (ii) uses the subgradient condition in Eq. (13), $\mathcal{L}(\boldsymbol{G}_k)-\mathcal{L}(\boldsymbol{G}^*)\geq\langle\nabla\mathcal{L}^{(k)},\boldsymbol{G}_k-\boldsymbol{G}^*\rangle$ (both for $\mathcal{L}_a$ and $\mathcal{L}_u$). Rearranging

Eq. (14a) and Eq. (14f), we get

$$2\eta_l^{(k)}(\mathcal{L}_a(\boldsymbol{G}_k) - \mathcal{L}_a^*) + 2\eta_l^{(k)}(\mathcal{L}_u(\boldsymbol{G}_k) - \mathcal{L}_u^*) \leq \mathbb{E}[\|\boldsymbol{G}_k - \boldsymbol{G}^*\|_2^2] - \mathbb{E}[\|\boldsymbol{G}_{k+1} - \boldsymbol{G}^*\|_2^2] + (\eta_l^{(k)})^2\delta_k^2. \quad (15)$$

By summing iterates up to step K, we get

$$2\left(\sum_{k=1}^{K}\eta_l^{(k)}\right)\min_{k\in[k]}[\mathcal{L}_a(\boldsymbol{G}_k) - \mathcal{L}_a^*] + \min_{k\in[k]}[\mathcal{L}_u(\boldsymbol{G}_k) - \mathcal{L}_u^*] \quad (16a)$$

$$\overset{\text{(iii)}}{\leq} 2\sum_{k=1}^{K}\eta_l^{(k)}[\mathcal{L}_a(\boldsymbol{G}_k) - \mathcal{L}_a^*] + [\mathcal{L}_u(\boldsymbol{G}_k) - \mathcal{L}_u^*] \quad (16b)$$

$$\overset{\text{(iv)}}{\leq} \|\mathbf{G}_1 - \mathbf{G}^*\|_2^2 + \sum_{k=1}^{K}(\eta_l^{(k)})^2\delta_k^2 \quad (16c)$$

where (iii) is valid since we take the minimum over all iterations and (iv) is derived from the summation of Eq. (15). Then, arranging Eq. (16a) and Eq. (16c) gives

$$\min_{k\in[k]}[\mathcal{L}_1(\boldsymbol{G}_k) - \mathcal{L}_1^*] + \min_{k\in[k]}[\mathcal{L}_2(\boldsymbol{G}_k) - \mathcal{L}_2^*] \leq \frac{\|\boldsymbol{G}_1 - \boldsymbol{G}^*\|_2^2 + \sum_{k=1}^{K}(\eta_l^{(k)})^2\delta_k^2}{2\sum_{k=1}^{K}\eta_l^{(k)}} \quad (17a)$$

Thus, if the 2-norm of the vectorized version of $\boldsymbol{G}_1 - \boldsymbol{G}^*$ is bounded by r, and with learning rate $\sum_k \eta_l^{(k)} \to \infty$ but $\sum_k (\eta_l^{(k)})^2 < \infty$, the right hand-side of Eq. (17a) becomes $\frac{r^2 + \sum_k (\eta_l^{(k)})^2\delta_k^2}{2\sum_k \eta_l^{(k)}} \to 0$. Therefore, using the gradient updates in step1 and step3 minimizes the losses $\mathcal{L}_a$, $\mathcal{L}_u$ and converges to a local optimal point.

4. Experiments

In this section, we evaluate the capability of our linear filter to (1) generate perturbed smart meter data that reduces the prediction accuracy of sensitive attributes; (2) maintain the minimum energy cost from an optimal control decision using the perturbed data; (3) integrate into a contemporary deep learning architecture with parallelism. The code for our experiments is available at https://github.com/markcx/DER_ControlPrivateTimeSeries.

4.1. Setup

We build up two neural networks to form the adversarial classifier and generator. The adversarial classifier is composed of two fully connected layers with ELU (Exponential Linear Unit) activation to estimate the sensitive attribute from demand. The first layer contains the same number of neurons as the time steps of the meter data series used by the battery optimal controller, and the second layer has half of the neuron numbers of the first layer and outputs a two dimensional vector representing the probability of the associated categories of the label. The generator module is composed of a single linear layer that takes a standard normal random vector and the private labels as inputs, and outputs noise to be added to the original demand. The parameters of the single linear layer form matrix $\boldsymbol{G}$. Additionally, we specify $\boldsymbol{G}$ to be block diagonal to reduce the number of learning parameters, i.e. $\boldsymbol{G} = [\Gamma, \boldsymbol{V}]$ where Γ is a diagonal matrix. Given the number of columns in our weight matrix is c_w (e.g. the c_w for $\boldsymbol{G}$ is 26 for the solar dataset and 50 in our residential experiments), we use uniform initialization[34] between $(-\frac{1}{c_w}, \frac{1}{c_w})$ for both the adversary and generator networks. We use 85% of the data for training and the remaining 15% for testing the performance of the filter. Later in Section 4.4, we demonstrate that our method is robust to different training and testing splits. We set hyper-parameters $\beta_1 = \beta_2 = \beta_3 = 10^{-5}$, $\varkappa = 10^{-3}$ throughout the experiments. The learning rate for the classifier is 10^{-3} and the learning rate for the generator starts from 0.1 and decays 20% for every 100 steps. We present the classification accuracy to indicate the correlation, as a lower accuracy implies a lower value of mutual information[35], thus, there is less correlation between the demand and sensitive labels. We set the initial battery state of charge to 1% of its maximum energy capacity, i.e. $B_{init} = 0.01B$. We use a time-of-use price structure with two tiers: a high price of $0.463 per KWh from 4pm-9pm and $0.202 per KWh for the rest of the day.

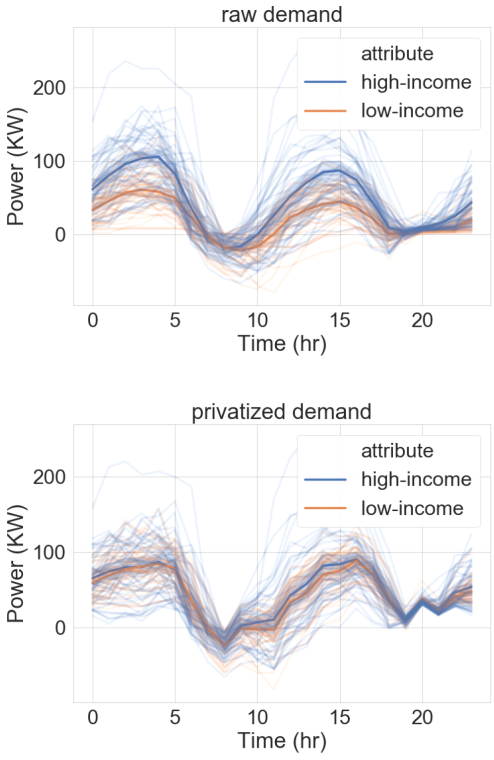

Fig. 1. A batch of 24-hour demand with solar generation that is net negative in certain hours allowing storage to minimize the cost through an optimal charge and discharge sequence. The **upper panel** shows the raw demand. The **lower panel** shows the privatized demand.

4.2. Examples

4.2.1. Deployment of storage on aggregated demand with solar generation

For our first experiment, we aggregated 24-hour demand consumption from thousands of homes into groups of 100–200 homes and added solar generation. The aggregations represent the demand seen at a secondary transformer from the perspective of a utility company. The goal is to minimize the energy cost for the aggregation of homes by running the optimal charging and discharging controls for battery storage located at the secondary transformer given a prescribed price. Before the experiment, each demand profile is assigned a binary label indicating if it is from a high- or low-income group, with high-income groups having a peak demand above a certain threshold. During the experiment, we wish to privatize the demand before sending it to the storage operator to perform cost minimization, so the operator cannot infer whether the aggregation of customers comes from a high or low-income group. The upper panel of Fig. 1 shows the income attribute can be easily inferred from the raw demand as the height of the peaks are clearly distinguishable. The lower panel of Fig. 1 shows that the privatized demands are perturbed such that two labels overlap making it harder to tell which demand has high or low income. However, there is a trade-off between privacy and utility when perturbing the data. We use the hyper-parameter λ_a to balance the adversarial loss and the utility loss i.e. smaller λ_a means less weight for privacy and more for utility, as shown in Fig. 2. When λ_a increases from 8 to 128, the classification accuracy of the income label drops from 89.4% to 73% as we expected. The raw classification accuracy with zero weight is 95.2%. The loss of performance of the cost minimization by using

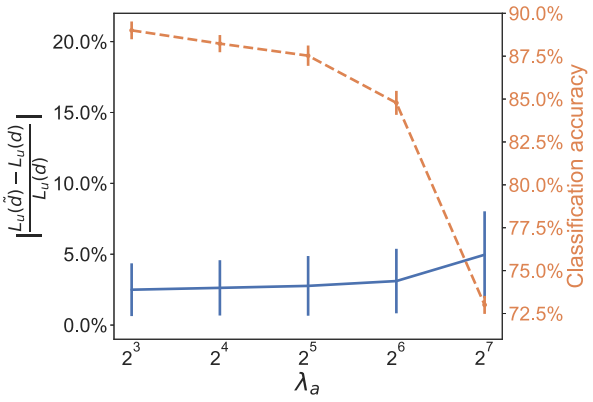

Fig. 2. The trade-off between privacy and utility controlled by parameter λ_a, which places weight on the private attribute classification loss.

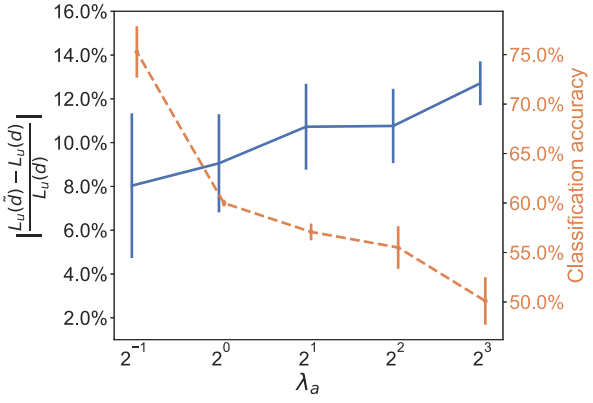

Fig. 3. The trade-off between the utility and privacy for the CER dataset [36]. The privacy label indicates a large or small home. λ_a weighs the privacy loss.

privatized demand instead of raw demand ranges from 2.5% at $\lambda_a = 8$ to almost 5% at $\lambda_a = 128$ on average, which shows that high privacy comes with a performance cost for this battery control problem.

4.2.2. Deployment of storage on residential users

The second experiment considers residential customers adopting batteries to minimize their energy cost without selling excess to the grid. The control of the battery is performed by an outside program, so the owner wishes to privatize their demand before sending it to the controller. The dataset is from the Irish CER Smart Metering Project [24,36]. We select a year of meter data for meters that contain a record indicating if they belong to a large or small home and partition it into daily sequences with 48 entries for each day. We end up with 54,478 records in total. Recall that our goal is to create altered demand that won't degrade the cost savings while removing the correlation between the demand and the attribute indicating a small or large home. Differences between this experiment and the previous one are that this experiment uses data from only a single home versus an aggregation of homes, and this experiment uses real world labeled data instead of synthetic labels. Fig. 3 depicts the trade-off between utility degradation and privacy gain for different weights on privacy loss. The accuracy of classifying large or small homes based on the raw demand is 77.5%. When we have low weight on the privacy loss (e.g. $\lambda_a = 0.5$), the classification accuracy only drops a little to 75%, with a greater sacrifice on cost saving performance (e.g. increased to 8% more cost on average). In the high privacy weight scenario, the classification accuracy drops down to 50% as desired, while the utility performance gap only increases up to 12%. When comparing this experiment to the previous one, we find that the adversary has more difficulty determining home size for individual homes than for aggregations of homes with comparable loss of cost minimization performance.

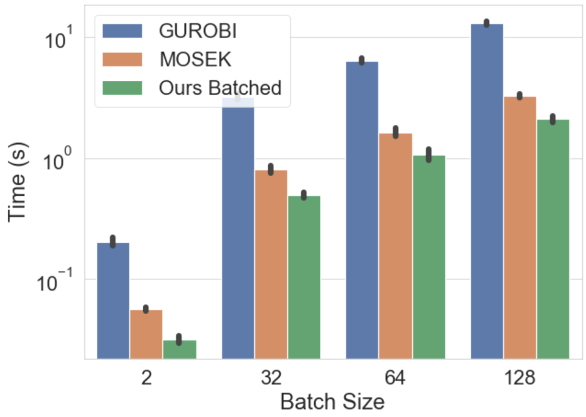

Fig. 4. CPU run time of a batched optimization using Gurobi v8.1.0, Mosek v8.1.0.60, and our parallel module.

4.2.3. Integration into real world systems

This approach can be integrated into existing storage control systems such as those proposed in [2,3] in the following manner. First, the privacy filter is trained offline using an anonymous batch of private data from many sources before the installation of the storage system and control algorithm. Then, the learned filter weights are given to the data owner who wishes to use the system. Next, during operation, the data owner locally privatizes the power demand data by locally computing the matrix product between the learned filter weights and the power demand data. The matrix product can be computed locally with minimal computation since the filter weight matrix is diagonal. Finally, the storage control algorithm receives the privatized data that is computed locally and performs the cost minimization optimization on the privatized data just as it would with raw data.

4.3. Parallelism

The training for the experiments in this section are run on a six-core Intel Core i7 CPU @2.2GHz. Current standard solvers like Gurobi or Mosek without support of in-batch parallelism can be computationally expensive for solving a quadratic problem. Our filter makes use of automatic differentiation for a cone program (DIFFCP) [29] and leverages multiprocessing to speed up the forward and backward calculations.

Fig. 4 displays the mean and standard deviation of running each trial 8 times, showing that our batched module outperforms Gurobi or Mosek, which are highly tuned commercial solvers for reasonable batch sizes. For a minibatch size of 128, we solve all problems in an average of 1.31 s, whereas Gurobi takes an average of 11.7 s. This speed improvement for a single minibatch makes the difference between a practical and an unusable solver in the context of training a deep learning architecture.

4.4. Sensitivity analysis

In this section, we evaluate the sensitivity of our method to: (i) the inherent trade-off between data privacy and utility, and (ii) the ratio of training data to testing data. First, we summarize our findings on the trade-off between data privacy and utility. As discussed, the tunable hyper-parameter, λ_a, allows us to scale the importance of privacy. In the first example, when λ_a increases from 8 to 128, the classification accuracy of the income label drops from 89.4% to 73% while loss of performance of the cost minimization increases from 2.5% to almost 5% on average as seen in Fig. 2. In the second example, when λ_a increases from 0.5 to 4, the classification accuracy drops from 75% to 50% while loss of cost saving performance increases from 8% to 12% as seen in Fig. 3. These performance values represent a Pareto optimal set parameterized by λ_a with the best point depending on the specific external values assigned to privacy and utility for the given scenario.

Table 1
Evaluation of performance on various train/test splits of the Irish CER data when $\lambda_a = 2$.

Train/Test(%)	baseline	65/35	70/30	75/25	80/20	85/15
acc. (%)[†]	77.5	63.3	61.1	57.6	58.5	56.9
cost gap (%)[*]	0	11.7	9.4	12.2	10.0	10.9

[*] Accuracy of private attribute. [*] Gap above optimal energy cost of controlling batteries with raw data. Lower values are preferred for accuracy and optimal objective gap.

Here, we demonstrate the robustness of our method to the training data by evaluating the performance of battery control on the residential dataset via various ratios of training/testing split with fixed $\lambda_a = 2$. The results are shown in Table 1. We find that the classification accuracy of the private attribute and the sacrificed cost gap are consistently around 57–61% and 9–12% respectively. This difference is small compared to other sources of variation such as the choice of λ_a, and comparable to the variation seen from different batches within the data. Such a result indicates our approach is relatively robust to different training and testing splits of the dataset.

5. Conclusion

We have presented a method for the privatization of personal data that maintains its utility in the optimal control of energy resources. Our method comprises a small linear filter that adds random noise to the data conditional on the private attributes we wish to protect. The linear filter is trained using a minimax optimization procedure that balances the trade-off between classifcation accuracy of the private attributes and the performance of an optimal controller. Additionally, we include a distortion penalty to preserve aspects of the data that are not specified by the utility or privacy functions in order to avoid adding arbitrary noise. We have demonstrated that this method is effective in two datasets and easy to integrate into real world DER control solutions. In the first dataset on aggregations of homes, the private label accuracy dropped by 26% while the utility performance gap only increased by 5%. The second dataset on individual homes saw the classification accuracy for the binary label drop down to the minimum of 50%, while the utility performance gap only increased up to 12%. Limitations of this method include the requirement to solve an optimization in the training loop, which can be computationally intensive for large problems; however, we suspect only a few iterations of the optimization are needed to achieve the desired gradients, which will dramatically reduce the computation required. Future work will look intro reducing the training computation time with fewer optimization iterations, increasing the variety of experiments with additional private labels and utility optimizations, and the consideration of additional noise due to poor data quality.

6. Appendix

6.1. Battery control details

We present a snapshot of the results for the storage control based on raw and private demand data. Fig. 5 displays the storage control for our experiment with aggregated homes and solar generation. The upper-left and lower-left panel show the 24-hour charging and discharging decisions with each color representing one sample in a batch. The control decisions made with raw versus privatized demand data are closely aligned in general, but have different charging and discharging amounts of power due to perturbation. However, such an altered charging profile doesn't increase the minimum cost of energy too much as we can see from the upper-right and lower-right panels of Fig. 5. The electricity cost increases by a maximum of $22 USD per day given that the highest daily cost is around US $390 USD. (Each bin spans the range

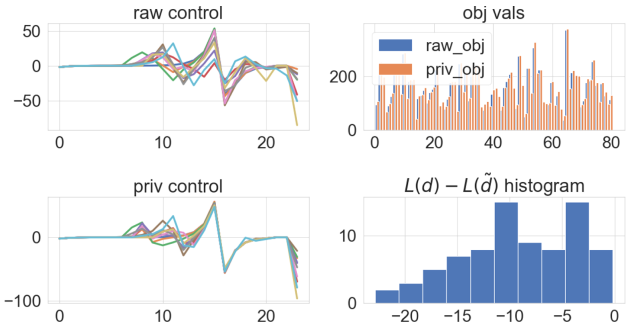

Fig. 5. Analysis of storage control for the aggregated homes experiment with $\lambda_a = 128$. The **upper-** and **lower-left** panel show the charging and discharging power in kilowatts (KW). Different colored curves represent different samples in the batch. The **upper-right** panel shows the daily electricity cost when operating the battery using raw or private demand (x-axis is the sample number, y-axis is in dollars ($)). The **lower-right** panel shows a histogram of the loss gap. (The x-axis is the increased cost in $; the y-axis is the number of days that show similar cost increases in a batch.) .

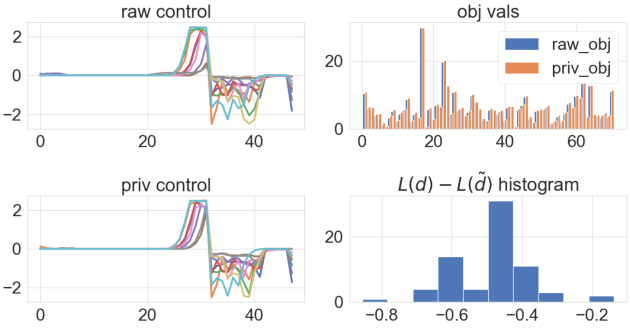

Fig. 6. Analysis of storage control for the CER data experiment with $\lambda_a = 8$. Each panel has the same x- and y-axis as Fig. 5.

of $2.5 USD for Fig. 5.) Fig. 6 shows the same information, but for the second experiment on individual home data.

6.2. Quadratic problem

A canonical form of the quadratic constrained minimization problem (QP) is expressed as follows:

$$\min_x \quad \frac{1}{2}x^T Q x + q^T x \tag{18a}$$

$$\text{s.t} \quad Ax = b \tag{18b}$$

$$Gx \leq h. \tag{18c}$$

We first show that the basic battery storage problem can be considered as a special case of QP. We start with the 24-hour horizon storage problem in Problem 1. We can express the constraints from Eq. (1d) to Eq. (1f) as

$$\underbrace{\begin{bmatrix} I & 0 & 0 \\ -I & 0 & 0 \\ 0 & I & 0 \\ 0 & -I & 0 \\ 0 & 0 & I \\ 0 & 0 & -I \\ -I & I & 0 \end{bmatrix}}_{G} \begin{bmatrix} x_{in} \\ x_{out} \\ x_s \end{bmatrix} \leq \begin{bmatrix} c_{in} \\ 0 \\ c_{out} \\ 0 \\ B \\ 0 \\ \boldsymbol{d} \end{bmatrix} \Leftrightarrow Gx \leq h. \tag{19}$$

We add a constraint that the net of the demand and storage is greater than or equal to 0, so we can formulate the objective as a QP. This constraint does not modify the original problem as long as it is feasible because the optimal solution will implicitly make the net of demand and storage greater than or equal to 0. The constraints in

Eq. (1b)-Eq. (1c) are expressed as

$$\underbrace{\begin{bmatrix} 0 & 0 & 1,\cdots 0 \\ [\eta_{in} I,\ 0] & [-\frac{1}{\eta_{out}} I,\ 0] & [I,\ 0]-[0,\ I] \end{bmatrix}}_{A} \begin{bmatrix} x_{in} \\ x_{out} \\ x_s \end{bmatrix} = \begin{bmatrix} B_{init} \\ 0 \end{bmatrix}$$

$$\Leftrightarrow Ax = b, \tag{20}$$

with $[I,\ 0] \in \mathbb{R}^{23\times 24}$. The objective Eq. (1a) can be converted to a standard QP by letting

$$\boldsymbol{Q} = \begin{bmatrix} \beta_1 I & 0 & 0 \\ 0 & \beta_2 I & 0 \\ 0 & 0 & \beta_3 I \end{bmatrix}, \quad q = \begin{bmatrix} p \\ -p \\ -2\beta_3 \alpha B\mathbf{1} \end{bmatrix}. \tag{21}$$

Therefore, it is straightforward to discover that $x^T Qx + q^T x$ is the new form of the objective.

Declaration of Competing Interest

None.

References

[1] A. Bernstein, E. Dall'Anese, Bi-level dynamic optimization with feedback, IEEE Global Conference on Signal and Information Processing (GlobalSIP), Montreal, Canada, (2017).
[2] Y. Shi, B. Xu, D. Wang, B. Zhang, Using battery storage for peak shaving and frequency regulation: joint optimization for superlinear gains, IEEE Trans. Power Syst. 33 (2018) 2882–2894.
[3] T. Navidi, A. El Gamal, R. Rajagopal, A two-layer decentralized control architecture for der coordination, 2018 IEEE Conference on Decision and Control, (2018), pp. 6019–6024.
[4] P. Voigt, A.v.d. Bussche, The EU General Data Protection Regulation (GDPR): A Practical Guide, first ed., Springer Publishing Company, Incorporated, 2017.
[5] M. Lisovich, D. Mulligan, S. Wicker, Inferring personal information from demand-response systems, Secur. Priv. IEEE 8 (2010) 11–20.
[6] M. Jawurek, F. Kerschbaum, G. Danezis, Sok: Privacy Technologies for Smart Grids–A Survey of Options, 1 (2012) 1–16.
[7] N. Komninos, E. Philippou, A. Pitsillides, Survey in smart grid and smart home security: issues, challenges and countermeasures, IEEE Commun. Surv. Tut. 16 (4) (2014) 1933–1954.
[8] G. Giaconi, D. Gunduz, H.V. Poor, Privacy-aware smart metering: progress and challenges, IEEE Signal Process. Mag. 35 (6) (2018) 59–78.
[9] A. Halder, X. Geng, P. Kumar, L. Xie, Architecture and algorithms for privacy preserving thermal inertial load management by a load serving entity, IEEE Trans. Power Syst. 32 (4) (2016) 3275–3286.
[10] M. Keshk, E. Sitnikova, N. Moustafa, J. Hu, I. Khalil, An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems, IEEE Trans. Sustain. Comput. (2019).
[11] N. Buescher, S. Boukoros, S. Bauregger, S. Katzenbeisser, Two is not enough: privacy assessment of aggregation schemes in smart metering, Proc. Priv. Enhanc. Technol. 2017 (4) (2017) 198–214.
[12] H. Corrigan-Gibbs, D. Boneh, Prio: private, robust, and scalable computation of aggregate statistics, 14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17), (2017), pp. 259–282.
[13] C. Dwork, Differential privacy: a survey of results, in: M. Agrawal, D. Du, Z. Duan, A. Li (Eds.), Theory and Applications of Models of Computation, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 1–19.
[14] L. Sankar, S.R. Rajagopalan, S. Mohajer, H.V. Poor, Smart meter privacy: a theoretical framework, IEEE Trans. Smart Grid 4 (2) (2012) 837–846.
[15] S. Han, U. Topcu, G.J. Pappas, Event-based information-theoretic privacy: a case study of smart meters, 2016 American Control Conference (ACC), IEEE, 2016, pp. 2074–2079.
[16] J.-X. Chin, T.T. De Rubira, G. Hug, Privacy-protecting energy management unit through model-distribution predictive control, IEEE Trans. Smart Grid 8 (6) (2017) 3084–3093.
[17] G. Eibl, D. Engel, Differential privacy for real smart metering data, Comput. Sci.-Res. Dev. 32 (1–2) (2017) 173–182.
[18] F. Zhou, J. Anderson, S.H. Low, Differential privacy of aggregated DC optimal power flow data, 2019 American Control Conference (ACC), (2019), pp. 1307–1314.
[19] F. Fioretto, T.W. Mak, P. Van Hentenryck, Differential privacy for power grid obfuscation, IEEE Trans. Smart Grid (2019).
[20] D.P. Bertsekas, Dynamic Programming and Optimal Control 4th edition, Volume II, Athena Scientific, 2015.
[21] M. Liu, P.K. Phanivong, D.S. Callaway, Customer-and network-aware decentralized EV charging control, 2018 Power Systems Computation Conference (PSCC), IEEE, 2018, pp. 1–7.
[22] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch, NIPS 2017 Workshop Autodiff Program, (2017).
[23] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, Focal loss for dense object detection, The IEEE International Conference on Computer Vision (ICCV), (2017).
[24] C. Beckel, L. Sadamori, T. Staake, S. Santini, Revealing household characteristics from smart meter data, Energy 78 (2014) 397–410.
[25] X. Chen, P. Kairouz, R. Rajagopal, Understanding compressive adversarial privacy, 2018 IEEE Conference on Decision and Control (CDC), IEEE, 2018, pp. 6824–6831.
[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Advances in Neural Information Processing Systems, (2014), pp. 2672–2680.
[27] Y. Chen, X. Wang, B. Zhang, An unsupervised deep learning approach for scenario forecasts, 2018 Power Systems Computation Conference (PSCC), IEEE, 2018, pp. 1–7.
[28] B. Amos, J.Z. Kolter, Optnet: Differentiable optimization as a layer in neural networks, Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, 2017, pp. 136–145.
[29] A. Agrawal, S. Barratt, S. Boyd, E. Busseti, W.M. Moursi, Differentiating through a conic program, J. Appl. Numer. Optim. 1 (2019) 107–115.
[30] Y. Ye, M.J. Todd, S. Mizuno, An o($\sqrt{n}L$)-iteration homogeneous and self-dual linear programming algorithm, Math. Oper. Res. 19 (1) (1994) 53–67.
[31] E. Busseti, W.M. Moursi, S. Boyd, Solution refinement at regular points of conic problems, Comput. Optim. Appl. (2018) 1–17.
[32] L. Bottou, Large-scale machine learning with stochastic gradient descent, Proceedings of COMPSTAT'2010, Springer, 2010, pp. 177–186.
[33] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[34] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on imagenet classification, Proceedings of the IEEE International Conference on Computer Vision, (2015), pp. 1026–1034.
[35] X. Chen, T. Navidi, S. Ermon, R. Rajagopal, Distributed generation of privacy preserving data with user customization, Safe Machine Learning Workshop at ICLR, (2019).
[36] C. for Energy Regulation., Smart Metering Project-Electricity Customer Behaviour Trial, 2009–2010, 2012, (http://www.ucd.ie/issda/data/commissionforenergyregulationcer/).