Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)
Journal Track

OptStream: Releasing Time Series Privately (Extended Abstract)*

Ferdinando Fioretto[1] and **Pascal Van Hentenryck**[2]
[1]Syracuse University
[2]Georgia Institute of Technology
ffiorett@syr.edu, pvh@isye.gatech.edu

Abstract

Many applications of machine learning and optimization operate on *sensitive* data streams, posing significant privacy risks for individuals whose data appear in the stream. Motivated by an application in energy systems, this paper presents OPTSTREAM, a novel algorithm for releasing differentially private data streams under the w-event model of privacy. The procedure ensures privacy while guaranteeing bounded error on the released data stream. OPTSTREAM is evaluated on a test case involving the release of a real data stream from the largest European transmission operator. Experimental results show that OPTSTREAM may not only improve the accuracy of state-of-the-art methods by at least one order of magnitude but also support accurate load forecasting on the privacy-preserving data.

1 Introduction

This paper was motivated by a desire to release privacy-preserving streams of energy demands, also called *loads*, in transmission systems. The goal is to protect changes in consumer loads up to some desired amount within critical time intervals. Although customer identities are typically considered public information (e.g., each facility is served by some energy provider), their loads can be highly sensitive as they may reveal the economic activities of grid customers. For example, changes in load consumption may indirectly reveal production levels and strategic investments. Moreover, these time series are often input to complex analytic tasks, e.g., demand forecasting algorithms [Nogales *et al.*, 2002] and optimal power flows [Ochoa and Harrison, 2011]. As a result, the accuracy of the privacy-preserving datasets is critical and, as shown later in the paper, existing privacy-preserving algorithms for time series fall short in this respect for this application.

The main contribution of this paper is a new privacy mechanism that remedies these limitations and is sufficiently precise for use in forecasting and optimization applications. The new algorithm, called OPTSTREAM, is presented under the framework of w-event privacy and is a 4-step procedure consisting of sampling, perturbation, reconstruction, and post-processing modules. The *sampling* module selects a small set of points for privacy-preserving measurement in each period of interest, the *perturbation* module introduces noise to the sampled data points to guarantee privacy, the *reconstruction* module re-assembles the non-sampled data points from the perturbed ones, and the *post-processing* module uses convex optimization over the privacy-preserving output of the previous modules, as well as the privacy-preserving answers of additional queries on the data stream, to improve accuracy by redistributing the added noise. It is important to emphasize that, although OPTSTREAM was motivated by an energy application, it is potentially useful for many other domains since its design is independent of the underlying problem.

OPTSTREAM is evaluated on real datasets from *Réseau de Transport d'Électricité*, the French transmission operator and the largest in Europe. Experimental results show that OPTSTREAM improves the accuracy of state-of-the-art algorithms by at least one order of magnitude for this application domain and show that it supports accurate load forecasting on the privacy-preserving data.

2 Preliminaries

2.1 Privacy Model and Goals

A *data stream* is an infinite sequence of tuples (i, t), describing an event reported by user i that occurred at a discrete time t. This paper uses a simplified notation and denotes the data stream as a vector $\boldsymbol{x} = (x_1, x_2, \ldots)$ with each $x_t \in \mathbb{R}_+$ describing a positive quantity, such as that associated to the aggregated consumption of a set of customers at time t. A *stream prefix* $\boldsymbol{x}[t]$ describes the sequence $x_1, \ldots, x_t$ of all tuples observed on or before time t.

At every time step t, the data curator receives information about data x_t and wishes to privately report such quantity. In the target application of this paper, the data curator is interested in publishing every element x_t for a recurring period of w time steps. A *w-period* is a set of w contiguous time steps $t-w+1,\ldots,t$ ($w \geqslant 1$). Thus, private reports x_t are generated in real time for windows of w time steps.

*This paper is an extended abstract of the article *OptStream: Releasing Time Series Privately* in Journal of Artificial Intelligence Research [Fioretto and Van Hentenryck, 2019].

2.2 Differential Privacy and w-Privacy

Differential Privacy [Dwork, 2010] focuses on protecting the participation of an individual user in a computation. In a nutshell, an algorithm $\mathcal{A}$, that takes as input a dataset D and returns a response o from some output set, is ϵ-differentially private if, for all possible datasets D' differing from D by only one individual, and any output responses o,

$$\Pr[\mathcal{A}(D) \in o] \leqslant \exp(\epsilon) \Pr[\mathcal{A}(D') \in o].$$

The privacy level is controlled by parameter $\epsilon \geqslant 0$, describing the privacy loss, with small values denoting strong privacy.

The w-event privacy framework [Kellaris *et al.*, 2014] extends the definition of differential privacy to protect data streams. The framework operates on stream prefixes and two data streams prefixes $\boldsymbol{x}[t]$ and $\boldsymbol{x}'[t]$ are said *w-neighbors*, denoted by $\boldsymbol{x}[t] \sim_w \boldsymbol{x}'[t]$, if

i. for each x_i, x'_i with $i \in [t]$, $x_i \sim x'_i$, and

ii. for each x_i, x'_i, x_j, x'_j such that $i < j \in [t]$ and $x_i \neq x'_i, x_j \neq x'_j$, $j - i + 1 \leqslant w$ holds.

In other words, two stream prefixes are w-neighbors if their elements are *pairwise* neighbors and all the differing elements are within a time window of up to w time steps. As a result, when ensuring the privacy guarantees, the w-event framework does not consider data streams where the differences are beyond a time window of size w: It only needs to consider windows of w elements.

Definition 1 (w-privacy) *Let $\mathcal{A}$ be a randomized algorithm that takes as input a stream prefix $\boldsymbol{x}[t]$ of arbitrary size and outputs an element o from a set of possible output sequences $\mathcal{O}$. Algorithm $\mathcal{A}$ satisfies w-event ϵ-differential privacy (w-privacy for short) if, for all t, all outputs $o \in \mathcal{O}$, and all w-neighboring stream prefixes $\boldsymbol{x}[t]$ and $\boldsymbol{x}'[t]$:*

$$Pr[\mathcal{A}(\boldsymbol{x}[t]) \in o] \leqslant \exp(\epsilon) Pr[\mathcal{A}(\boldsymbol{x}'[t]) \in o]. \quad (1)$$

An algorithm satisfying w-privacy protects the sensitive information that could be disclosed from a sequence of finite length w. All the classical properties of differential privacy, including composability and immunity to post-processing [Dwork and Roth, 2013] carry over to w-privacy.

The Laplace mechanism which adds Laplace noise to each element of the stream with parameter $w\Delta/\epsilon$ achieves w-privacy [Kellaris *et al.*, 2014], where Δ is the maximal contribution of an individual to the data stream.

3 OptStream For Stream Release

The proposed algorithm processes a data stream $\boldsymbol{x} = (x_1, x_2, \ldots)$, along with the period size w whose privacy is to be protected, and the total privacy loss ϵ, and outputs a privacy-preserving version $\hat{\boldsymbol{x}} = (\hat{x}_1, \hat{x}_2, \ldots)$ of the stream $\boldsymbol{x}$. OPTSTREAM processes the data stream in consecutive and disjoint w-periods consists of four steps: (1) data sampling, (2) perturbation, (3) reconstruction of the non-sampled data points, and (4) optimization-based post-processing, summarized below.[1] The algorithm seeks to balance two types of errors: a *perturbation error*, introduced by the application of additive noise at the sampled points, and a *reconstruction error*, introduced by the reconstruction procedure at the non-sampled points. The higher the number of samples in a w-period, the more perturbation error is introduced while the reconstruction error may be reduced, and vice-versa. The error generated by these two components, in combination with the number of samples that minimizes the error, are analyzed in the full paper.

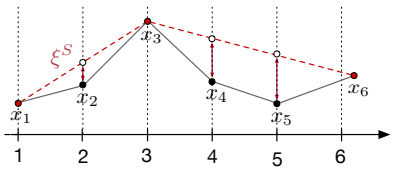

Figure 1: Illustration example of the SAMPLE step with $S = \{1, 3, 6\}$. The solid black curve connects the data point $\boldsymbol{x}$, the dashed red curve denotes the linear interpolation of the selected points, and the red arrows denote the L_1 errors.

The SAMPLE Procedure

The procedure selects a subsample S of k points for each w-period. Its goal is to perform a dimensionality reduction over the data stream whose sample points can be used to generate privacy-preserving stream data points with low error. It does so by minimizing the $L1$ error between the values associated to the original data points $\boldsymbol{x}$ and those associated to the points generated by a linear interpolation ξ^S of the k selected points, denoted as $\boldsymbol{x}^S = (x_i | i \in S)$. Formally, it minimizes the following quantity: $\sum_{i \in S} |\xi_i^S - \boldsymbol{x}_i^S|$. The idea is illustrated in Figure 1. The procedure uses a DP greedy algorithm to select a set S that produces a low $L1$ error. It is an instantiation of the *Sparse Vector Technique* [Hardt and Rothblum, 2010], an iterative algorithm that allows answering a sequence of queries consuming low privacy loss, and it uses a portion ϵ_s of the overall privacy loss budget ϵ.

The PERTURB Procedure

Next, OPTSTREAM uses the canonical Laplace mechanism to guarantee privacy perturbing the k data points sampled in the previous step. The perturbed data points $\tilde{\boldsymbol{x}}_S$ of $\boldsymbol{x}$ satisfy ϵ_p-differential privacy, with ϵ_p being a portion of the overall privacy loss budget ϵ.

The RECONSTRUCT Procedure

The goal of the reconstruction procedure is to re-assemble the non-sampled data point from the perturbed ones $\tilde{\boldsymbol{x}}_S$ using linear interpolation. The procedure uses exclusively privacy-preserving data and thus induces no additional privacy loss. In [Fioretto and Van Hentenryck, 2019] the paper analyzes the error induced by this step on the data stream.

The POST-PROCESS Procedure

Finally, the algorithm uses a key post-processing component to enforce consistency of salient features of the data. Its essence is a convex optimization program that employs the privacy-preserving output $\tilde{\boldsymbol{x}}_S$ of the above modules, as well as privacy-preserving answers to additional queries (called *feature queries*) on the data stream.

[1] For an in-depth description of the OPTSTREAM procedures and their theoretical analysis, please refer to the full paper [Fioretto and Van Hentenryck, 2019].

Formally, a *feature* is a partition of the w-period and we say that a feature $\mathbf{F}'$ is a *sub-feature* of $\mathbf{F}$, denoted by $\mathbf{F}' \prec \mathbf{F}$, if $\mathbf{F}'$ is obtained by sub-partitioning $\mathbf{F}$. The *feature query* $Q_{\mathbf{F}}(\boldsymbol{x})$ associated with feature $\mathbf{F} = \{\mathbf{d}_1, \ldots, \mathbf{d}_m\}$ returns an m-dimensional vector $(c_1, \ldots, c_m)$ where each c_i is the sum of the values x_j of $\boldsymbol{x}$ for $j \in \mathbf{d}_i$. For example, a feature $\mathbf{F}_i(\boldsymbol{x})$ may be described by the partition $\mathbf{F}_i = \{\{1, \ldots, \lfloor w/2 \rfloor\}, \{\lceil w/2 \rceil, \ldots, w\}\}$ that divides the w-interval in two equal segments, and its associated query set $Q_{\mathbf{F}_i} = \{\sum_{j=1}^{\lfloor w/2 \rfloor} \boldsymbol{x}_j, \sum_{j=\lceil w/2 \rceil}^{w} \boldsymbol{x}_j\}$ represents the aggregated count on the partition induced by $\mathbf{F}_i$ on $\boldsymbol{x}$. For notational simplicity, we assume that the first feature always partitions the data stream w-period into singletons, i.e., $\mathbf{F}_1 = \{\{i\} : i \in [w]\}$. Note that its associated privacy-preserving query $Q_{\mathbf{F}_1} = \tilde{\boldsymbol{x}}$ is the output of the perturbation procedure.

When viewed as queries, the inputs to the mechanism can be represented as a set of values $Q_{\mathbf{F}_i}(\boldsymbol{x}) = \mathbf{c}_i = (c_{i1}, \ldots, c_{im_i})$ $(1 \leqslant i \leqslant p)$ or, more concisely, as $\boldsymbol{c} = (c_{11}, \ldots, c_{pm_p})$, and $\tilde{\mathbf{c}}$ is used to represent their associated noisy (i.e., privacy-preserving) version, obtained via an application of the Laplace mechanism. We assume that a partial ordering $\prec$ of features is given, and notice that the feature queries $Q_{\mathbf{F}}(\boldsymbol{x})$ form a lattice on $\boldsymbol{x}$. The essence of this procedure is the following optimization program that finds a new vector $\boldsymbol{x}^*$ that minimizes:

$$\mathbf{min}_{\dot{\boldsymbol{x}}} \|\dot{\boldsymbol{x}} - \tilde{\boldsymbol{c}}\|_2^2 \tag{O1}$$

$$\textbf{s.t.: } \dot{x}_{ij} = \sum_{l:\mathbf{d}_{i'l} \subseteq \mathbf{d}_{ij}} \dot{x}_{i'l} \;\; \forall i', i : \mathbf{F}_{i'} \prec \mathbf{F}_i, \quad j \in [m_i] \tag{O2}$$

$$\forall i, j : \dot{x}_{ij} \geqslant 0. \tag{O3}$$

Its decision variables are the post-processed values $\dot{\boldsymbol{x}} = (\dot{x}_{11}, \ldots, \dot{x}_{pm_p})$ and the objective minimizes the squared L_2-Norm of $\dot{\boldsymbol{x}} - \tilde{\boldsymbol{c}}$. The optimization is subject to a set of *consistency constraints* among comparable features (Constraints (O2)) and non-negativity constraints on the variables (Constraints (O3)). By definition of sub-features, there exists a set of elements in $\mathbf{F}_{i'}$ whose union is equal to $\mathbf{d}_{ij}$.

Fioretto and Van Hentenryck [2019] shows that the optimization-based post-process achieves ϵ_o-differential privacy, it bounds the expected error with respect to the original stream by a contact factor, and that OPTSTREAM satisfies w-event ϵ-differential privacy.

4 Evaluation

Dataset and Algorithms

The source data was obtained through a collaboration with *Réseau de Transport d'Électricité*, the largest energy transmission system operator in Europe. It consists of a one-year national-level load energy consumption data, which is aggregated at a regional level. Each data point in the stream represents the total load consumption of the customers served within a region during a 30 minute time period. For evaluation purposes, the experiments often consider a representative region (Auvergne - Rhône-Alpes) to analyze the data stream release.

The following sections compare OPTSTREAM against the *Laplace mechanism* and the *Discrete Fourier Transform (DFT)* algorithm [Rastogi and Nath, 2010]. All the algorithms release privacy-preserving data associated with the sub-streams $\boldsymbol{x}$ for each w-period and preserve the same level of privacy. An in-depth description of the algorithms, their parameters, and evaluation setting, as well as a more extensive evaluation, that uses additional algorithms and settings, are provided in the full paper.

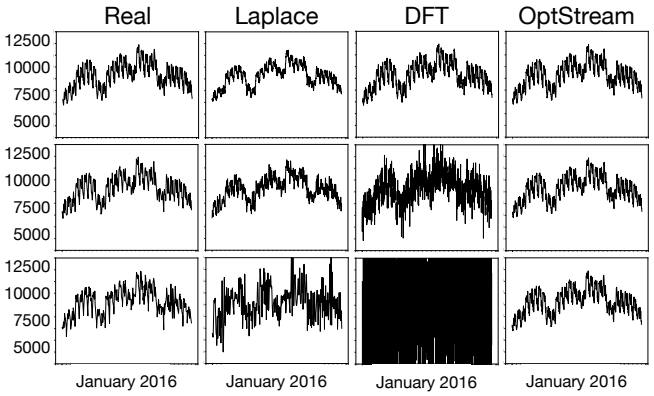

Figure 2: Real load consumption data, in MW, for the Auvergne-Rhône-Alpes region in January 2016 (first column) and its privacy-preserving versions obtained using Laplace (second column), DFT (third column), and OPTSTREAM (fourth column) with privacy loss $\epsilon = 1$ (top), $\epsilon = 0.1$ (middle), and $\epsilon = 0.01$ (bottom).

Privacy-Preserving Stream Release

Answering queries over contiguous w-periods corresponds to releasing the privacy-preserving stream over the entire available duration. Figure 2 illustrates the real and privacy-preserving versions of the data-stream for the Auvergne-Rhône-Alpes region in January, 2016. It uses w-periods of size 48 for given privacy losses $\epsilon = 1.0, 0.1$, and 0.01, shown in the top, middle, and bottom rows, respectively. The choice for the w-period allows the data curator to ensure the protection of the observed power consumptions within each period. Thus, the released stream protects loads in each entire day.

The real loads are illustrated in the first column. The figure compares our proposed OPTSTREAM algorithm (fourth column) against the *Laplace mechanism* (second column), and the DFT algorithm [Rastogi and Nath, 2010] (third column). The experiments set the number of Fourier coefficients in the DFT and sampling steps in OPTSTREAM to 10. The privacy loss allocated to perform each measurement is split equally. Additionally, for OPTSTREAM $\epsilon_s = \epsilon_p = \epsilon_o = \frac{1}{3}\epsilon$. Finally, OPTSTREAM uses feature queries representing salient moments in the day associated with different consumption patterns. These are proxy of consumer behaviors and thus energy consumption. Finally, if an algorithm reports negative noisy value for a stream point, we truncate it to zero.

Figure 2 clearly illustrates that, for a given privacy disclosure level, OPTSTREAM *produces privacy-preserving streams that are substantially more accurate than its competitors when visualized*.

A quantification of the errors reported by the algorithms is reported in Figure 3. It shows the L_1-errors (in logarithmic scale) between the original $\boldsymbol{x}$ and the privacy preserving $\hat{\boldsymbol{x}}$ streams obtained by the algorithms analyzed for the months of February (left), June (middle), and October (right). These

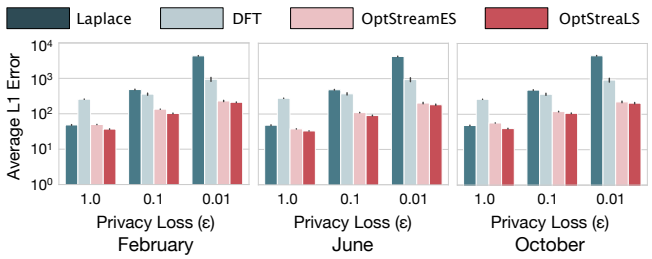

Figure 3: L_1-errors: Load stream data for the months of February (left), June (middle), and October (right). The y-axis reports $\log_{10}$ of the average L_1-error for the stream data.

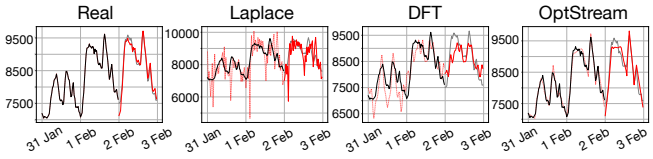

Figure 4: Prediction error: Forecast for a one day load consumption through an ARMA model on the real load consumption data (Real) and its privacy-preserving versions obtained using Laplace, DFT, and OPTSTREAM with privacy loss $\epsilon = 0.1$.

months capture the different customers load profile behaviors due to different weather patterns and durations of the day light. In addition to OPTSTREAM (denoted with suffix LS, in the Figure) an additional version (OPTSTREAMES) is presented; It differs from OPTSTREAM only for its sampling process, which is equally-spaced and thus spends no privacy loss budget (i.e., $\epsilon_s = 0$). The figure highlights that OPTSTREAM consistently outperforms competitor algorithms.

Impact of Privacy on Forecasting Demand

The final results evaluate the capability of the released privacy-preserving streams to accurately predict future consumptions. To do so, the paper adopts the Autoregressive Moving Average (ARMA) model [Alwan and Roberts, 1988; Zhang, 2003], which is a popular stochastic model used for predicting future points in a time series. The ARMA model with parameters p and q refers to the model with p autoregressive terms and q moving-average terms: It estimates a future time step value x_t as $c + \beta_t + \sum_{i=1}^{p} \phi_i x_{t-i} + \sum_{i=1}^{q} \theta_i \beta_{t-i}$, where where c is a constant, β_t is a random variable modeling white noise at time t, ϕ_i and θ_i are, respectively, the autoregressive and moving average model parameters.

The experiments use an ARMA model with parameters $p = q = 1$ to estimate the future 48 time steps (corresponding to a day) when trained with the past four weeks of the privacy-preserving data stream estimated using Laplace, DFT, and OPTSTREAM with $L1$-sampling. All models use the same parameters adopted in the previous sections.

Figure 4 visualizes the forecast for the load consumptions in the Auvergne-Rhône-Alpes region for February 2, 2016. The black and gray solid lines illustrate, respectively, the real load values observed so far and those of the day to be forecasted. The dotted red lines illustrate the privacy-preserving stream data estimated so far (and used as input to the prediction model) and the solid red lines depict the prediction obtained using the ARMA model. Figure 4 shows the forecast results using the real data (Real) and the privacy-preserving stream obtained through Laplace, DFT, and OPTSTREAM, respectively. *The figure clearly shows that* OPTSTREAM *is able to produce substantially better estimates for the next day forecast.*

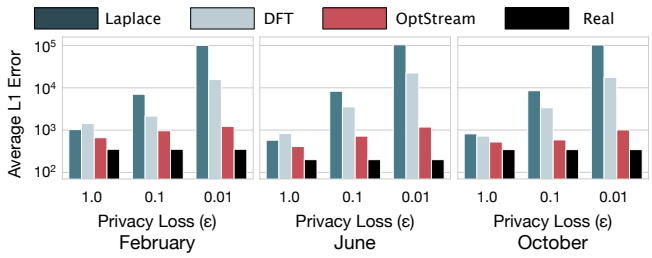

Figure 5: L_1 error analysis: ARMA forecasting model on stream data for the energy loads of the months of February, June, and October for the Auvergne-Rhône-Alpes region.

The experiments also quantitatively evaluate the average L_1-error for each prediction produced by the mechanisms. They adopt the same setting as above for the prediction and report, in Figure 5, the average L_1-errors for predicting each day in the month of February, June, and October for the Auvergne-Rhône-Alpes region. Each histogram reports the $\log_{10}$ value of the average error of 30 random trials. We observe that OPTSTREAM reports substantially smaller errors compared to all other privacy-preserving algorithms, and that the error made by OPTSTREAM in reporting the next day forecast is closer to the error made in the forecast prediction using the real data than when using another method.

5 Conclusions

This paper presented OPTSTREAM, a novel algorithm for privately releasing stream data in the w-event privacy model. OPTSTREAM is a 4-step procedure consisting of sampling, perturbation, reconstruction, and post-processing modules. OPTSTREAM was evaluated on a real dataset from the largest transmission operator in Europe. Experimental results on multiple test cases show that OPTSTREAM improves the accuracy of the state-of-the-art by at least one order of magnitude in this application domain. The accuracy improvements are measured, not only in terms of the error distance to the original stream but also in the accuracy of a popular load forecasting algorithm trained on privacy-preserving data substreams. In the full paper, the results additionally show that OPTSTREAM exhibits similar benefits on hierarchical stream data which is also highly desirable in practice. An important direction of future work is to generalize these results to the streaming setting where a data element is emitted at each time step. Future work will also focus on ensuring that salient properties of an optimization problem of interest hold, when the problem relies on inputs that include the load consumption data, e.g., as in [Fioretto and Van Hentenryck, 2018; Fioretto and Van Hentenryck, 2018; Fioretto *et al.*, 2020; Mak *et al.*, 2020].

References

[Alwan and Roberts, 1988] Layth C Alwan and Harry V Roberts. Time-series modeling for statistical process control. *Journal of Business & Economic Statistics*, 6(1):87–95, 1988.

[Dwork and Roth, 2013] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Theoretical Computer Science*, 9(3-4):211–407, 2013.

[Dwork, 2010] Cynthia Dwork. Differential privacy in new settings. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 174–183. SIAM, 2010.

[Fioretto and Van Hentenryck, 2018] Ferdinando Fioretto and Pascal Van Hentenryck. Constrained-based differential privacy: Releasing optimal power flow benchmarks privately. In *Proceedings of the Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR)*, pages 215–231, 2018.

[Fioretto and Van Hentenryck, 2019] Ferdinando Fioretto and Pascal Van Hentenryck. Optstream: Releasing time series privately. *Journal of Artificial Intelligence Research*, 65:423–456, 2019.

[Fioretto *et al.*, 2020] Ferdinando Fioretto, Terrence W. K. Mak, and Pascal Van Hentenryck. Differential privacy for power grid obfuscation. *IEEE Transactions on Smart Grid*, 11(2):1356–1366, March 2020.

[Hardt and Rothblum, 2010] Moritz Hardt and Guy N Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *Foundations of Computer Science*. IEEE, 2010.

[Kellaris *et al.*, 2014] Georgios Kellaris, Stavros Papadopoulos, Xiaokui Xiao, and Dimitris Papadias. Differentially private event sequences over infinite streams. *Proceedings of the VLDB Endowment*, 7(12):1155–1166, 2014.

[Mak *et al.*, 2020] Terrence W. K. Mak, Ferdinando Fioretto, Lyndon Shi, and Pascal Van Hentenryck. Privacy-preserving power system obfuscation: A bilevel optimization approach. *IEEE Transactions on Power Systems*, 35(2):1627–1637, March 2020.

[Nogales *et al.*, 2002] Francisco Javier Nogales, Javier Contreras, Antonio J Conejo, and Rosario Espínola. Forecasting next-day electricity prices by time series models. *IEEE Transactions on power systems*, 17(2):342–348, 2002.

[Ochoa and Harrison, 2011] Luis F Ochoa and Gareth P Harrison. Minimizing energy losses: Optimal accommodation and smart operation of renewable distributed generation. *IEEE Transactions on Power Systems*, 26(1):198–205, 2011.

[Rastogi and Nath, 2010] Vibhor Rastogi and Suman Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 735–746. ACM, 2010.

[Zhang, 2003] G Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.