Differential Privacy for Class-based Data: A Practical Gaussian Mechanism

Raksha Ramakrishna, *Member, IEEE*, Anna Scaglione, *Fellow, IEEE*, Tong Wu, *Member, IEEE*, Nikhil Ravi, *Student Member, IEEE* and Sean Peisert, *Senior Member, IEEE*

***Abstract*—In this paper, we present a notion of differential privacy (DP) for data that comes from different classes. Here, the class-membership is private information that needs to be protected. The proposed method is an output perturbation mechanism that adds noise to the release of query response such that the analyst is unable to infer the underlying class label. The proposed DP method is capable of not only protecting the privacy of class-based data but also meets quality metrics of accuracy and is computationally efficient and practical. We empirically illustrate the proposed method's efficacy while outperforming the baseline Gaussian noise mechanism. We also examine a real-world application and apply the proposed DP method to the autoregression and moving average (ARMA) forecasting method, protecting the privacy of the underlying data source. Case studies on the real-world advanced metering infrastructure (AMI) measurements validate the excellent performance of the proposed DP method while also satisfying the accuracy of forecasted AMI measurements.**

***Index Terms*—Differential Privacy, class-based privacy, Gaussian mechanism, autoregression and moving average, smart meter data.**

I. Introduction

Differential privacy (DP) [1], [2] has become one of the most critical concepts in database privacy, gaining an essential foothold in ensuring that personal or data private to individual database entries remains indistinguishable after the database is queried. Broadly speaking, a privacy mechanism for a certain data query is differentially private if the output of this data query changes minimally, or with a very low probability, when a single database entry is added or removed. Numerous papers give a comprehensive overview of differential privacy. In [3], the authors treat the problem of differential privacy from a signal processing perspective and provide a review of the different mechanisms employed. Differential privacy mechanisms can be broadly classified as input perturbation, i.e., adding noise to the input or to the data before responding to the query, and output perturbation, noise addition after the query response is computed. Most of the time, Laplacian or Gaussian noise is added to the input or the output of the query. The so-called *exponential mechanism* [4], [5] is another often-used differential privacy mechanism with good guarantees, particularly in the case where the utility of differentially private query response is also taken into account.

Traditional or classical Gaussian mechanisms [1] add uncorrelated Gaussian noise, whose variance is calculated on the basis of sensitivity of neighboring sets of private data to the query and privacy-level parameters. It was shown in [6] that the variance could be further reduced given a privacy budget to obtain better utility. A post-processing or denoising mechanism is also introduced to improve the accuracy of the query.

Despite the popularity and use of DP in a variety of situations, there are cases where it is not possible to employ the original definition and mechanisms for DP. A main drawback with the traditional DP is that the definition of neighboring datasets which pertains to addition or removal of a single data point is not applicable to many use cases. For example, query response for a single data stream where some important attributes of this data stream are to be protected rather than hiding the presence or absence of a single data point in a large database or data stream. To address this drawback, there have been other notions of privacy that are more applicable for scenarios that do not necessarily cater to large databases. Two such frameworks are the Pufferfish [7] and Blowfish [8] privacy. Here, they define privacy with respect to pairs of 'secrets' that must remain indistinguishable after the privacy mechanism is used to release the data. These frameworks have been successfully utilized in many scenarios and particularly that of trajectory or location release of an individual [9]–[11] or activity monitoring [12]. However, the privacy mechanisms presented in the aforementioned papers are either computationally expensive, do not scale with the size of sample space, or do not incorporate the utility of the released data while designing the mechanism. Furthermore, none of the papers mentioned discuss an output perturbation method such as the Gaussian mechanism, which optimizes the accuracy of the query.

As an application, we consider the release of forecasts performed fitting an Auto-Regressive Moving-Average (ARMA) model. ARMA prediction models have many variants, which are widely used for time-series forecasts; the underlying assumption is that the time series is a realization of a Gaussian process, with a parametric structure for its covariance that

Raksha Ramakrishna is with Division of Network and Systems Engineering, EECS, KTH Royal Institute of Technology, Stockholm, Sweden. e-mail: rakshar@kth.se.

Anna Scaglione, Tong Wu and Nikhil Ravi are with the Department of Electrical and Computer Engineering, Cornell Tech, New York City, NY 10044 USA. e-mail:{as337, tw385, nr337}@cornell.edu.

Sean Peisert is with the Computer Sciences Research and Development Department, Lawrence Berkeley National Laboratory, Berkeley, CA 94720 USA. e-mail: sppeisert@lbl.gov

This research was supported by the Director, Cybersecurity, Energy Security, and Emergency Response, Cybersecurity for Energy Delivery Systems program, of the U.S. Department of Energy, under contract DE-AC02-05CH11231. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors of this work.

is determined by the parameters of an ARMA filter. The Gaussian assumption implies that the Minimum Mean Squared Error (MMSE) forecast is the conditional mean of the future samples given the past, and it is a linear affine function of the observation that depends on the first and second order statistics of the process. A useful application we explore is the release of the predicted power consumption for homes in a way that its statistics are (ϵ, δ) private relative to other homes which could be, for instance, in the same neighborhood. Releasing time-series in a private manner is yet another paradigm where the definition of neighboring data or adjacency is different, as it is in [13], [14]. The query response mechanisms in these papers correspond to down-sampling data points in a window, adding DP-noise and reconstructing the time-series which is shown to be differentially-private.

An alternative way to make forecasts private would be to carry out input perturbation or regression in a differentially-private manner. Recent work [15] studies linear regression in a differentially-private manner, where the input data and the corresponding labels are made private by perturbing the sufficient statistics. Related to the proposed Gaussian mechanism, in [16], a Bayesian linear regression is carried out, i.e., the posterior of the regression parameters is computed in a private manner, to protect the underlying data, by perturbing the terms comprised of private data that are required for estimation of regression parameters. Furthermore, the noisy posterior is 'denoised' or, in other words, a noise-aware inference is undertaken. In [17], both input data and labels are made private by the addition of noise to sufficient statistics when the query is Gaussian process regression. Like in [16], noise-aware posterior computation of parameters is discussed. Compared to the papers discussed above, the differences not only include a change to definition of neighborhood of data but also the output perturbation mechanism, whose privacy guarantees are easier to analyze.

A. *Differential privacy in smart grid systems*

As mentioned before, the practical application of our DP mechanism we highlight is sharing power consumption forecasts. Differential privacy mechanisms have been widely used in class-based smart grid data classification and forecasting. For example, [18] injects functional DP noises into the meter data to achieve a certain level of differential privacy. In [19], the authors proposed a novel randomized battery-based load hiding algorithm which assures differential privacy for smart metering data. In [20], an innovative DP compliant algorithm was developed to ensure that the data from consumer's smart meters are protected. Moreover, spectral DP is presented to protect the frequency content of power system time-series data in [21], [22]. DP techniques are also applied for power system operation to protect users' energy consumption patterns in [23]–[25]. In particular, [24], [25] consider a privacy-preserving optimal power flow (OPF) mechanism for distribution grids that secures customer privacy from unauthorized access to OPF solutions, e.g., current and voltage measurements. In [23], the authors investigate how to utilize DP techniques to release the data for power networks where the parameters of transmission lines and transformers are obfuscated. None of these works consider the release of forecasts that are DP.

B. *Contributions*

We introduce the concept of differential privacy for label or class data. Our framework is related to the Pufferfish privacy mechanism, as discussed above. The sensitive or private information here is the class label or the hypothesis, rather than the presence or absence of a single data point. Then, we discuss an additive noise mechanism for the release of the query response on such data so that the analyst is unable to infer the underlying class label. The optimality of the scheme rests on the assumption that the data from each class have a Gaussian multivariate distribution with a class specific mean and covariance. Furthermore, we propose a Gaussian query response mechanism that is computationally efficient and practical because it also meets accuracy requirements. We apply the DP mechanism for the release of ARMA forecasts, testing it numerically on synthetic and real data.

C. *Notation*

Boldfaced lower case letters, $\boldsymbol{x}$, are used to denote vectors whereas upper-case letters, $\boldsymbol{X}$, are for matrices. Calligraphic letters, $\mathcal{X}$, are used to denote sets.

II. Problem setting

In this section, we elucidate the problem and review the appropriate definitions from literature. The setting of the problem is that the *data owner* requires sensitive information to be private, while also wanting to answer queries by a third party analyst with as little error or distortion to the query response as possible.

In this work, we denote by $X \in \mathcal{X}$ the sensitive information which the *data owner* wants to hold private. One can also refer to this as the label, class or hypothesis X, that is hidden from the analyst. For brevity, we use *label* to refer to the sensitive information X. Let the probability distribution of data, $\boldsymbol{d} \in \mathbb{R}^n$, that is assumed to be generated given the label, X, be $f(\boldsymbol{d}|X)$.

The *analyst* wishes to apply queries or functions $\mathbb{Q} : \mathbb{R}^n \to \mathbb{R}^k$ to the data $\boldsymbol{d}$. We denote the outcome of the query as $\boldsymbol{q} \in \mathbb{R}^k$, i.e., $\boldsymbol{q} = \mathbb{Q}(\boldsymbol{d})$ which is a stochastic function or function of a random variable. We assume that the query at a certain instance only acts on data generated when the system is in a specific label X, i.e., the query does not combine information from multiple instances with system in different labels. Thus, we denote the probability distribution of $\boldsymbol{q}$ as $f(\boldsymbol{q}|X)$. Note that the query output is already modeled as a *random outcome*. However, releasing $\boldsymbol{q}$ as is could aid a malicious analyst in inferring the label X through standard classification schemes with prior information or belief about X. Therefore, rather than answering the query directly $\boldsymbol{q} = \mathbb{Q}(\boldsymbol{d})$, the *data owner* publishes the output of a randomized algorithm or mechanism denoted by $\mathbb{A}_{\mathrm{Q}}$. Thus, the *data owner* releases or publishes query response, $\tilde{\boldsymbol{q}} = \mathbb{A}_{\mathrm{Q}}(\boldsymbol{d}|X)$. Again, due to the dependence on a label X, we denote the probability density of the released query as $f(\tilde{\boldsymbol{q}}|X)$.

The goal of the randomized mechanism $\mathbb{A}_{\mathrm{Q}}$ is to confuse the analyst by making the query output given the label X indistinguishable from the query output when the underlying label is a *neighbor* to X. The definition of neighboring labels will be made clearer later.

For the published data to be useful, the answer to the query $\tilde{\boldsymbol{q}}$ is required to meet certain quality metrics, such as accuracy (denoted by ρ). For the rest of the paper, *accuracy* refers to accuracy in expectation over the domain of the privatized data.

Next, we review the relevant concepts from literature to compare and contrast the ideas from the problem setting described above.

III. Privacy: Definitions and Metrics

A. Neighborhood of a data label

The *data owner* needs to define a priori for every label X, a subset of labels $X' \in \mathcal{X}$ as neighbors. These neighbor labels are chosen such that privacy is maintained based on our class-membership based definition. In other words, the labels X and X' should be indistinguishable from the query answer alone. This is similar to the concept of pair of secrets in the Pufferfish privacy framework [12].

A graph topology $\mathcal{G}$ whose nodes' set is $\mathcal{X}$ and the edges set, $\mathcal{E}$, can encode information on what ought to be hidden,

$$\mathcal{G} = (\mathcal{X}, \mathcal{E}), \quad \mathcal{E} = \{(X, X') | X \in \mathcal{X}, X' \in \mathcal{X}_X^{(1)}\}, \tag{1}$$

One could use a notion of distance $\mathbb{d}(X, X')$ that can define a geometric graph structure where the subset $\mathcal{X}_X^{(1)}$ that is the neighborhood $\mathcal{X}_X^{(1)}$ of each node X is defined as:

$$\mathcal{X}_X^{(1)} = \{X' | \mathbb{d}(X, X') = 1, X' \in \mathcal{X}\}, \tag{2}$$

describing all the X' that are *at distance one* from X and should be giving similar query answers as X. Furthermore, the graph structure is assumed to be undirected,

$$X' \in \mathcal{X}_X^{(1)} \Leftrightarrow X \in \mathcal{X}_{X'}^{(1)} \tag{3}$$

The *data owner* needs to design the neighborhood graph $\mathcal{G}$ that determines the neighborhood for each label X. Neighborhood can be designed based on desired indistinguishable labels. In the case of very strict privacy requirements, a fully connected graph can be used as a neighborhood graph so that every label has all other labels as neighbors.

B. Definition of Privacy

Conventionally, given the random published answer or realization of $\tilde{\boldsymbol{q}}$ in the differential privacy literature, the name *privacy loss* is used as a synonym for the log-likelihood ratio:

$$L_{xx'}(\tilde{\boldsymbol{q}}) \triangleq \ln \frac{f(\tilde{\boldsymbol{q}}|X)}{f(\tilde{\boldsymbol{q}}|X')}. \tag{4}$$

The reason for the name is that in classical statistical inference, $L_{xx'}(\tilde{\boldsymbol{q}}) > 0$ yields the decision that the query output is generated by the distribution $\boldsymbol{f}(\tilde{\boldsymbol{q}}|X)$. If this event is infrequent, then often an alternative hypothesis X' (where the distribution is $\boldsymbol{f}(\tilde{\boldsymbol{q}}|X')$) will be chosen as the right probabilistic model. We now introduce the notion of (ϵ, δ) differential privacy, which applies to any random vector $\tilde{\boldsymbol{q}}$ that is not conditionally independent of the private information X:

Definition 1 ((ϵ, δ) Probabilistic Differential Privacy (PDP) [26])**.** *Consider the probability density of released query or randomized mechanism $\tilde{\boldsymbol{q}} \sim f(\tilde{\boldsymbol{q}}|X)$ that changes depending on the class $X \in \mathcal{X}$. The randomized mechanism producing $\tilde{\boldsymbol{q}}$ is (ϵ, δ)- Probabilistic Differentially Private (PDP) iff:*

$$Pr\left(|L_{xx'}(\tilde{\boldsymbol{q}})| > \epsilon\right) \leq \delta \quad \forall (X, X') \in \mathcal{E}. \tag{5}$$

It can be shown that (ϵ, δ)-PDP is a strictly stronger condition than (ϵ, δ)-DP.

Theorem 1 (PDP implies DP [27])**.** *If a randomized mechanism is (ϵ, δ)-PDP, then it is also (ϵ, δ)-DP, i.e.,*

$$(\epsilon, \delta) - PDP \Rightarrow (\epsilon, \delta) - DP, \text{ but } (\epsilon, \delta) - DP \nRightarrow (\epsilon, \delta) - PDP.$$

Given that PDP provides a more intuitive understanding of privacy than (ϵ, δ) DP and is a strictly stronger condition, we make use of PDP throughout this paper. PDP is not closed under post-processing [28] only if, prior to the query, one applies a non-bijective transformation. Going forward, we drop the absolute value while writing $|L_{xx'}(\tilde{\boldsymbol{q}})| > \epsilon$ since

$$|L_{xx'}(\tilde{\boldsymbol{q}})| > \epsilon \implies L_{xx'}(\tilde{\boldsymbol{q}}) > \epsilon, L_{xx'}(\tilde{\boldsymbol{q}}) < -\epsilon \tag{6}$$

$$L_{xx'}(\tilde{\boldsymbol{q}}) < -\epsilon \implies -L_{xx'}(\tilde{\boldsymbol{q}}) > \epsilon \implies L_{xx'}(\tilde{\boldsymbol{q}}) > \epsilon \tag{7}$$

As the neighborhood graph is undirected, and $Pr\left(|L_{xx'}(\tilde{\boldsymbol{q}})| > \epsilon\right) \leq \delta \quad \forall (X, X') \in \mathcal{E}$, it suffices to drop the absolute value and $Pr\left(L_{xx'}(\tilde{\boldsymbol{q}}) > \epsilon\right) \leq \delta \quad \forall (X, X') \in \mathcal{E}$ is equivalent to eq. (5). It is important to remark that prior statistical information about X is generally available, i.e., $f(X)$ is a prior belief or distribution of X. The following remark explains how Definition 1 is sufficient in general.

Remark 1. *If the analyst operates in the Bayesian setting and there is a statistical prior distribution on the possible outcomes for X, i.e., a probability model on $\mathcal{X}$, a more meaningful privacy loss definition than eq. (4) is expressed in terms of posterior distributions $f(X|\tilde{\boldsymbol{q}})$. Note, however, that:*

$$\ln \frac{f(X|\tilde{\boldsymbol{q}})}{f(X'|\tilde{\boldsymbol{q}})} = L_{xx'}(\tilde{\boldsymbol{q}}) + \ln \frac{f(X)}{f(X')} \tag{8}$$

where $f(X)$ is the prior distribution. This means that (ϵ, δ) private according to Definition 1 then:

$$\delta \geq \sup_{X \in \mathcal{X}} \sup_{X' \in \mathcal{X}_X^{(1)}} Pr\left(L_{xx'}(\tilde{\boldsymbol{q}}) > \epsilon - \ln \frac{f(X)}{f(X')}\right). \tag{9}$$

The implication is that the privacy loss distribution is inherently a function of the statistics of $L_{xx'}(\tilde{\boldsymbol{q}})$, even when there are priors, which makes the extension to the case where priors are given relatively straightforward. For this reason, we will continue the discussion considering the latent information, or label X, as deterministic and unknown.

C. Publication accuracy constraints

One of the aspects in which our framework differs from most other DP work, is that we consider the constraint for the *data owner* to guarantee a certain level of accuracy in the query response, we next define *query accuracy*. This is different from the notion of *query sensitivity*, which is an intrinsic property of the data, and it is a measure of the utility of the DP query response.

For a *continuous query*, the *accuracy* is a measure of how dissimilar the answer to the query is, and it is desired to have $\tilde{\boldsymbol{q}} \approx \mathbb{Q}(\boldsymbol{d})$. A possible simple measure, the average mean squared (MS) error per entry, is specified in the following:

Definition 2 (Mean Square Error Accuracy)**.** *For a continuous function* $\mathbb{Q}:\mathbb{R}^n \to \mathbb{R}^k$*:*

$$\rho_{\mathbb{Q}|X} = \frac{1}{k}\mathbb{E}[\|\tilde{\boldsymbol{q}} - \mathbb{Q}(\boldsymbol{d})\|_2^2], \tag{10}$$

is the ***expected mean-square query accuracy****, where* $\mathbf{d} \sim f(\boldsymbol{d}|X)$ *and* $X \in \mathcal{X}$ *is the information to hold private.*

D. Privacy under accuracy constraints

In our work, we adopt the Definition 1 for (ϵ, δ) privacy along with accuracy constraint and thereby define the overall privacy in our framework as follows:

Definition 3. *A randomized algorithm* $\mathbb{A}_Q$ *with outcome* $\tilde{\boldsymbol{q}} = \mathbb{A}_Q(\boldsymbol{d}|X)$*, is* (ϵ, δ)*-**private** meeting an **accuracy budget*** ρ *for a query* $\mathbb{Q}$ *iff the condition in eq. (5) holds* $\forall(X, X') \in \mathcal{E}$ *along with* $\rho_{\mathbb{Q}|X} \leq \rho \;\; \forall X \in \mathcal{X}$*.*

Having given all the necessary definitions, we now introduce our method of publishing the query response. We specifically consider query responses that are continuous valued and Gaussian distributed given label X.

IV. A Publishing Mechanism for Privacy with Additive Noise

The most popular designs of DP algorithms amount to adding random noise to the true query, i.e.:

$$\tilde{\boldsymbol{q}} = \boldsymbol{q} + \boldsymbol{\eta}, \tag{11}$$

where $\boldsymbol{\eta}$ is drawn from a family of distributions that facilitate the calculation of the (ϵ, δ) curves; zero mean Gaussian and Laplacian noise are the most frequent choices. Furthermore, the entries of the vector $\boldsymbol{\eta}$ are independent and identically distributed (i.i.d.) and they are independent of X, i.e. $f(\boldsymbol{\eta}|X) = f(\boldsymbol{\eta})$. This makes only one parameter, the variance, in the Gaussian and Laplacian distributions available as a degree of freedom, which is set once one defines the values of (ϵ, δ).

The idea behind the proposed publication methodology $\mathbb{A}_Q$ is simple: we also add Gaussian noise to the actual query response, but the noise $\boldsymbol{\eta}$ vector of the mechanism we propose $f_{\boldsymbol{\eta}|X}(\boldsymbol{\eta}|X)$, is label X dependent in general, and its entries of $\boldsymbol{\eta}$ are not i.i.d. This leaves us with several additional degrees of freedom to meet an accuracy constraint, which is a function of the noise statistics $f_{\boldsymbol{\eta}|X}(\boldsymbol{\eta})$, as we show next.

A. Accuracy for the additive noise scheme

We use the symbol $\boldsymbol{\Sigma}$ to denote covariance matrices and $\boldsymbol{\mu}$ to denote mean vectors, and use the suffix to indicate what is the random vector that is averaged; $\mathrm{Tr}(\boldsymbol{A})$ denotes the trace of the matrix $\boldsymbol{A}$. The average MSE accuracy can be expressed as a function of the conditional mean and the covariance of the noise:

$$\rho^{\mathrm{MSE}}_{\mathbb{Q}|X} = \frac{1}{k}\mathbb{E}[\|\boldsymbol{\eta}\|^2] = \mathbb{E}_{\boldsymbol{q}|X}\left[\mathrm{Tr}(\boldsymbol{\Sigma}_{\boldsymbol{\eta}|\boldsymbol{q}}) + \|\boldsymbol{\mu}_{\boldsymbol{\eta}|\boldsymbol{q}}\|^2\right] \tag{12}$$

$$= \frac{1}{k}\mathrm{Tr}(\boldsymbol{\Sigma}_{\boldsymbol{\eta}|X}) + \frac{1}{k}\|\boldsymbol{\mu}_{\boldsymbol{\eta}|X}\|^2, \tag{13}$$

B. Illustration: Classical DP additive noise mechanism

In contrast to our framework, the vast majority of the DP literature considers queries that are a deterministic function because they do not explictly consider the distribution of data but only consider the sensitivity or the range. However, as an illustration, the computation of the (ϵ, δ) curves when the data is deterministic given the label X is shown, which means the true query answer is also deterministic given X. We denote this by $\boldsymbol{q} \equiv \mathbb{Q}(X)$. The DP algorithm also adds zero mean random noise that is independent of X. In this case the noise is added to a constant and:

$$\tilde{\boldsymbol{q}} = \mathbb{Q}(X) + \boldsymbol{\eta} \quad \Rightarrow \quad f(\tilde{\boldsymbol{q}}|X) = f_{\boldsymbol{\eta}}(\tilde{\boldsymbol{q}} - \mathbb{Q}(X)), \tag{14}$$

which implies that for any pair X, X' $f(\tilde{\boldsymbol{q}}|X)$ and $f(\tilde{\boldsymbol{q}}|X')$ differ only in their means, $\mathbb{Q}(X)$ and $\mathbb{Q}(X')$. Let:

$$\boldsymbol{\mu}_{XX'} \triangleq \mathbb{Q}(X) - \mathbb{Q}(X'). \tag{15}$$

Now, let us define query sensitivity for this case:

Definition 4 (Deterministic Query Sensitivity)**.** *The sensitivity of a deterministic query about the data* X *is:*

$$\Delta_p \triangleq \sup_{X \in \mathcal{X}} \sup_{X' \in \mathcal{X}_X^{(1)}} \|\mathbb{Q}(X) - \mathbb{Q}(X')\|_p. \tag{16}$$

where $\mathrm{d}_Q \triangleq \|\mathbb{Q}(X) - \mathbb{Q}(X')\|_p$ *is an appropriate notion of distance as* ℓ_p *norm that measures how much the queries applied differ when the label is* X *or* X'*.*

In this case, the (ϵ, δ) privacy curve is entirely defined by the noise distribution and its change due to a shift in the mean:

$$Pr(L_{XX'}(\tilde{\boldsymbol{q}}) > \epsilon) = Pr\left(\ln \frac{f_{\boldsymbol{\eta}}(\tilde{\boldsymbol{q}} - \mathbb{Q}(X))}{f_{\boldsymbol{\eta}}(\tilde{\boldsymbol{q}} - \mathbb{Q}(X'))} > \epsilon\right) \tag{17}$$

$$\equiv Pr\left(\ln \frac{f_{\boldsymbol{\eta}}(\boldsymbol{\eta})}{f_{\boldsymbol{\eta}}(\boldsymbol{\eta} + \boldsymbol{\mu}_{XX'})} > \epsilon\right). \tag{18}$$

From eq. (18) it is apparent that the interplay of the noise distribution and the possible values for the offset $\boldsymbol{\mu}_{XX'}$ are all that is needed to establish the (ϵ, δ) privacy trade-off. Note that in the case of independent noise added to different dimensions of the query:

$$L_{XX'}(\tilde{\boldsymbol{q}}) = \sum_{i=1}^{k} L_{XX'}(\tilde{q}_i). \tag{19}$$

Indicating $[\boldsymbol{\mu}_{XX'}]_i = \mu_i$, the expression in eq. (18) is equivalent to:

$$Pr\left(\sum_{i=1}^{k} \ln \frac{f(\eta_i)}{f(\eta_i + \mu_i)} > \epsilon\right) \tag{20}$$

Next we derive (ϵ, δ) bounds for Gaussian zero mean i.i.d. noise, In this case $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \sigma_\eta^2 \boldsymbol{I})$, it is easy to show that [6]:

$$L_{xx'}(\tilde{\boldsymbol{q}}) = \frac{\boldsymbol{\mu}_{xx'}^\intercal (\tilde{\boldsymbol{q}} - \mathbb{Q}(X))}{\sigma_\eta^2} + \frac{\|\boldsymbol{\mu}_{xx'}\|^2}{2\sigma_\eta^2},$$

which implies that the likelihood $L_{xx'}(\tilde{\boldsymbol{q}})$ is also Gaussian:

$$L_{xx'}(\tilde{\boldsymbol{q}}) \sim \mathcal{N}\left(\frac{\|\boldsymbol{\mu}_{xx'}\|^2}{2\sigma_\eta^2}, \frac{\|\boldsymbol{\mu}_{xx'}\|^2}{\sigma_\eta^2}\right). \tag{21}$$

denoting by $Q(v) = \frac{1}{\sqrt{2\pi}} \int_v e^{-\frac{u^2}{2}} \, \mathrm{d}u$, then:

$$Pr(L_{xx'}(\tilde{\boldsymbol{q}}) > \epsilon) = Q\left(\frac{\epsilon - \frac{\|\boldsymbol{\mu}_{xx'}\|_2^2}{2\sigma_\eta^2}}{\frac{\|\boldsymbol{\mu}_{xx'}\|_2}{\sigma_\eta}}\right) \tag{22}$$

In this case, since the trend of the probability is a monotonic function of $\|\boldsymbol{\mu}_{xx'}\|_2$ (i.e., the Euclidean distance of the queries) a meaningful definition for query sensitivity in Definition 4 is $\Delta_2 = \sup_X \sup_{X' \in \mathcal{X}_X^{(1)}} \|\boldsymbol{\mu}_{xx'}\|_2$, which implies:

$$\delta = Q\left(\frac{\epsilon - \frac{\Delta_2^2}{2\sigma_\eta^2}}{\frac{\Delta_2}{\sigma_\eta}}\right), \tag{23}$$

and if we set a limit ρ for the MSE, then $\sigma_\eta^2 \leq \rho/k$.

C. Stochastic queries with additive noise

When the query, $\boldsymbol{q} = \mathbb{Q}(\boldsymbol{d})$ is modeled as being an outcome of a random ensemble whose distribution depends on the label X, one can directly calculate the inherent privacy using $f(\boldsymbol{q}|X)$ in lieu of $f(\tilde{\boldsymbol{q}}|X)$, i.e., (ϵ, δ) curves without adding noise at all. If the query responses without noise do not reveal what the underlying X is then there is no need to alter the data, i.e., the query response is naturally private without employing any random mechanism. This has been described in the literature [29]. However, one cannot control the generative mechanism for the data, and we would also need to add suitable noise to further reduce δ. The addition of suitable noise can mask the hypotheses and yield a lower δ for a certain ϵ, sacrificing the accuracy of the response. Now, the distribution of the query response after the addition of noise is computed as follows:

$$f(\tilde{\boldsymbol{q}}|\boldsymbol{q}, X) = f(\tilde{\boldsymbol{q}}|X) = \int f_{\boldsymbol{\eta}|X}(\tilde{\boldsymbol{q}} - \boldsymbol{q}|X) f(\boldsymbol{q}|X) \, \mathrm{d}\boldsymbol{q}. \tag{24}$$

The analytical calculation of this convolution integral would be non-trivial in general.

This is why, in this paper, we discuss the case where the query response given the label X is Gaussian and the noise is also Gaussian. We do this because such models are ubiquitously used. In mathematical terms, if $\boldsymbol{q} \sim \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$ (where $\boldsymbol{\mu}_x$ is the mean and $\boldsymbol{\Sigma}_x$ is the covariance matrix of $\boldsymbol{q}$ with label X), and $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{\mu}_\eta, \boldsymbol{\Sigma}_{\boldsymbol{\eta}})$, then $\tilde{\boldsymbol{q}} \sim \mathcal{N}(\boldsymbol{\mu}_x + \boldsymbol{\mu}_{\boldsymbol{\eta}}, \boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_{\boldsymbol{\eta}})$. To streamline the notation we will use the conventions:

$$\boldsymbol{\mu}_x \triangleq \boldsymbol{\mu}_{\boldsymbol{q}|X}, \quad \tilde{\boldsymbol{\mu}}_x \triangleq \boldsymbol{\mu}_{\tilde{\boldsymbol{q}}|X} = \boldsymbol{\mu}_x + \boldsymbol{\mu}_{\boldsymbol{\eta}|X}, \tag{25}$$

$$\boldsymbol{\Sigma}_x \triangleq \boldsymbol{\Sigma}_{\boldsymbol{q}|X}, \quad \tilde{\boldsymbol{\Sigma}}_x \triangleq \boldsymbol{\Sigma}_{\tilde{\boldsymbol{q}}|X} = \boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_{\boldsymbol{\eta}|X}. \tag{26}$$

In this case we bound explicitly the probability of the privacy loss random variable for both for the query $\boldsymbol{q}$ itself, $Pr(L_{xx'}(\boldsymbol{q}) > \epsilon)$, and for the published query $\tilde{\boldsymbol{q}}$, i.e. $Pr(L_{xx'}(\tilde{\boldsymbol{q}}) > \epsilon)$. For the analysis, we use the following fact:

Proposition 1. *Let the log-likelihood be* $L_{xx'}(\boldsymbol{q}) = \ln \frac{f(\boldsymbol{q}|X)}{f(\boldsymbol{q}|X')}$. *For the queries that are drawn when the private information is* X, *i.e.* $\boldsymbol{q} \sim \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$, *equivalently we have:*

$$Pr(L_{xx'}(\boldsymbol{q}) > \epsilon) \equiv Pr(L_{xx'}(\boldsymbol{\xi}) > \epsilon) \tag{27}$$

where $\boldsymbol{\xi}$ *and* $L_{xx'}(\boldsymbol{\xi})$ *are:*

$$\boldsymbol{\xi} \triangleq \boldsymbol{U}_{xx'}^\intercal \boldsymbol{\Sigma}_X^{-1/2} (\boldsymbol{q} - \boldsymbol{\mu}_x) \;\sim\; \mathcal{N}(\mathbf{0}, \boldsymbol{I}); \tag{28}$$

$$L(\boldsymbol{\xi}) \triangleq -\frac{1}{2} \ln |\boldsymbol{\Gamma}_{xx'}| + \frac{1}{2} \boldsymbol{\xi}^\intercal (\boldsymbol{\Gamma}_{xx'} - \boldsymbol{I}) \boldsymbol{\xi} - \boldsymbol{\mu}_{xx'}^\intercal \boldsymbol{\Gamma}_{xx'} \boldsymbol{\xi} + \frac{1}{2} \boldsymbol{\mu}_{xx'}^\intercal \boldsymbol{\Gamma}_{xx'} \boldsymbol{\mu}_{xx'}, \tag{29}$$

$$\text{and } \boldsymbol{\mu}_{xx'} \triangleq \boldsymbol{U}_{xx'}^\intercal \boldsymbol{\Sigma}_x^{-1/2} (\boldsymbol{\mu}_{x'} - \boldsymbol{\mu}_x). \tag{30}$$

$\boldsymbol{U}_{xx'}, \boldsymbol{\Gamma}_{xx'}$ *are the eigenvectors and eigenvalue matrices respectively,*

$$\boldsymbol{U}_{xx'} \boldsymbol{\Gamma}_{xx'} \boldsymbol{U}_{xx'}^\intercal \triangleq \boldsymbol{\Sigma}_x^{1/2} \boldsymbol{\Sigma}_{x'}^{-1} \boldsymbol{\Sigma}_x^{1/2}, \tag{31}$$

with the diagonal matrix $\boldsymbol{\Gamma}_{xx'} = \mathrm{diag}(\boldsymbol{\gamma}_{xx'})$ *and the vector* $\boldsymbol{\gamma}_{xx'}$, *whose entries are the eigenvalues in descending order and the unitary matrix of eigenvectors,* $\boldsymbol{U}_{xx'}$.

Proof. See Appendix A. □

Naturally, the same expressions hold for $\tilde{\boldsymbol{q}}$ except that to define $\tilde{\boldsymbol{U}}_{xx'}, \tilde{\boldsymbol{\Gamma}}_{xx'}$ and $\tilde{\boldsymbol{\mu}}_{xx'}$ we are using eq. (25) and eq. (26) and corresponding expressions for X'. For the conditionally Gaussian case, in order to create a better level of (ϵ, δ) privacy the noise design should reduce the difference of the means and covariances by adding noise. In particular, the quadratic term in eq. (29) becomes zero if $\boldsymbol{\Gamma}_{xx'} = \boldsymbol{I}$, which is the case when the noise covariances of the two hypotheses under X and X' are the same, and the remaining linear and constant terms go to zero if the difference between the mean vectors is zero. In the case of $\boldsymbol{\Gamma}_{xx'} = \boldsymbol{I}$, $Pr(L_{xx'}(\boldsymbol{q}) > \epsilon)$ can be computed in closed form as shown

Corollary 1. *Assume* $\boldsymbol{\Sigma}_x = \boldsymbol{\Sigma}_{x'}$, *so that* $\boldsymbol{\Gamma}_{xx'} = \boldsymbol{I}$. *Then:*

$$L_{xx'}(\boldsymbol{\xi}) \sim \mathcal{N}\left(\frac{1}{2} \|\boldsymbol{\mu}_{xx'}\|^2, \|\boldsymbol{\mu}_{xx'}\|^2\right). \tag{32}$$

and with $Q(v) = \frac{1}{\sqrt{2\pi}} \int_v e^{-\frac{u^2}{2}} \, \mathrm{d}u$, *then:*

$$Pr(L_{xx'}(\boldsymbol{q}) > \epsilon) = Q\left(\frac{\epsilon - \frac{\|\boldsymbol{\mu}_{xx'}\|^2}{2}}{\|\boldsymbol{\mu}_{xx'}\|}\right) \tag{33}$$

which is a monotonically increasing function of $\|\boldsymbol{\mu}_{xx'}\|^2$.

However, more generally when $\boldsymbol{\Sigma}_x \neq \boldsymbol{\Sigma}_{x'}$, we use the Chernoff bound for $Pr(L_{xx'}(\boldsymbol{q}) > \epsilon)$ to help evaluate the (ϵ, δ) privacy levels. The same expression can also be used as a bound for $Pr(L_{xx'}(\tilde{\boldsymbol{q}}) > \epsilon)$ using $\tilde{\boldsymbol{\mu}}_{xx'}$, and $\tilde{\boldsymbol{U}}_{xx'}, \tilde{\boldsymbol{\Gamma}}_{xx'}$ instead:

Lemma 1. *For all* $s > \max(1,\gamma_1)$ *where* $\gamma_1 = \lambda_{\max}(\boldsymbol{\Sigma}_X^{1/2}\boldsymbol{\Sigma}_{X'}^{-1}\boldsymbol{\Sigma}_X^{1/2})$*, it holds:*

$$Pr(L_{XX'}(\boldsymbol{q})>\epsilon) \leq \frac{(s-1)^{\frac{k}{2}}}{|\boldsymbol{\Gamma}_{XX'}|^{\frac{1}{2(s-1)}}|s\boldsymbol{I}-\boldsymbol{\Gamma}_{XX'}|^{1/2}} \times e^{-\frac{\epsilon}{(s-1)}+\frac{s}{2(s-1)}\boldsymbol{\mu}_{XX'}^{\intercal}(s\boldsymbol{I}-\boldsymbol{\Gamma}_{XX'})^{-1}\boldsymbol{\Gamma}_{XX'}\boldsymbol{\mu}_{XX'}} \quad (34)$$

Proof. See Appendix B. □

In order to understand the trends in the bound, we can specify it for some value of s in a corollary of Lemma 1:

Corollary 2. *For* $s \gg \max(1,\gamma_1)$*,*

$$Pr(L_{XX'}(\boldsymbol{q})>\epsilon) \lessapprox \frac{1}{|\boldsymbol{\Gamma}_{XX'}|^{\frac{1}{2s}}} e^{-\frac{\epsilon}{s}} e^{\frac{\boldsymbol{\mu}_{XX'}^{\intercal}\boldsymbol{\Gamma}_{XX'}\boldsymbol{\mu}_{XX'}}{2s}} \quad (35)$$

$$= e^{\frac{1}{2s}\left[\boldsymbol{\mu}_{XX'}^{\intercal}\boldsymbol{\Gamma}_{XX'}\boldsymbol{\mu}_{XX'}-\ln|\boldsymbol{\Gamma}_{XX'}|\right]} e^{-\frac{\epsilon}{s}} \quad (36)$$

The bounds suggests that, more generally, $Pr(L_{XX'}(\boldsymbol{q})>\epsilon)$ increases monotonically, as $\boldsymbol{\mu}_{XX'}^{\intercal}\boldsymbol{\Gamma}_{XX'}\boldsymbol{\mu}_{XX'} - \ln|\boldsymbol{\Gamma}_{XX'}|$ increases, which is consistent with the case where the expression is exact and $\boldsymbol{\Gamma}_{XX'} = \boldsymbol{I}$. This metric will be leveraged in our optimal design. In particular, we can assert that for the conditionally Gaussian case we can use the following (ϵ,δ) bound:

Corollary 3. *Let the Gaussian sensitivity be defined as:*

$$\Delta_G = \sup_{X} \sup_{X'\in\mathcal{X}_X^{(1)}} \frac{1}{2}\tilde{\boldsymbol{\mu}}_{XX'}^{\intercal}\tilde{\boldsymbol{\Gamma}}_{XX'}\tilde{\boldsymbol{\mu}}_{XX'} - \frac{1}{2}\ln|\tilde{\boldsymbol{\Gamma}}_{XX'}| \quad (37)$$

The following trend is a (ϵ,δ) *bound for the privacy loss:*

$$\delta_{\tilde{\boldsymbol{q}}}^{\epsilon} \leq \delta = e^{\frac{\Delta_G-\epsilon}{s}}. \quad (38)$$

The bound is simple but quite loose; however it does help to identify the worst case scenario which, in turn, helps to optimize the noise parameters.

D. Optimal design

In this section, we propose an algorithm to optimize the parameters of the additive noise mechanism algorithm $\mathbb{A}_{\mathbb{Q}}(\boldsymbol{d})$. Considering the expression of mean and covariance of $\tilde{\boldsymbol{q}}$ in eqs. (25) and (26) the expression we derived for $\boldsymbol{q}$ can be applied to derive $Pr(L_{XX'}(\tilde{\boldsymbol{q}})>\epsilon)$ by defining the corresponding vector:

$$\tilde{\boldsymbol{\mu}}_{XX'} \triangleq \tilde{\boldsymbol{U}}_{XX'}\tilde{\boldsymbol{\Sigma}}_X^{-\frac{1}{2}}(\tilde{\boldsymbol{\mu}}_{X'}-\tilde{\boldsymbol{\mu}}_X), \quad (39)$$

and define the unitary matrix $\tilde{\boldsymbol{U}}_{XX'}$ and diagonal matrix $\tilde{\boldsymbol{\Gamma}}_{XX'}$ through the following eigenvalue decomposition:

$$\tilde{\boldsymbol{U}}_{XX'}\tilde{\boldsymbol{\Gamma}}_{XX'}\tilde{\boldsymbol{U}}_{XX'}^{\intercal} \triangleq \tilde{\boldsymbol{\Sigma}}_X^{\frac{1}{2}}\tilde{\boldsymbol{\Sigma}}_{X'}^{-1}\tilde{\boldsymbol{\Sigma}}_X^{\frac{1}{2}}. \quad (40)$$

From the analysis in the previous section, and Corollaries 2 and 3, it seems that a good surrogate metric for achieving the best (ϵ,δ) privacy is:

$$\tilde{\boldsymbol{\mu}}_{XX'}^{\intercal}\tilde{\boldsymbol{\Gamma}}_{XX'}\tilde{\boldsymbol{\mu}}_{XX'} - \ln|\tilde{\boldsymbol{\Gamma}}_{XX'}| = (\tilde{\boldsymbol{\mu}}_{X'}-\tilde{\boldsymbol{\mu}}_X)^{\intercal}\tilde{\boldsymbol{\Sigma}}_{X'}^{-1}(\tilde{\boldsymbol{\mu}}_{X'}-\tilde{\boldsymbol{\mu}}_X) - \ln|\tilde{\boldsymbol{\Gamma}}_{XX'}| \quad (41)$$

In fact, with $s \gg \max(\gamma_1,1)$ for all $\tilde{\boldsymbol{\Gamma}}_{XX'}$:

$$\log\delta_{\tilde{\boldsymbol{q}}}^{\epsilon} \lessapprox \sup_{X\in\mathcal{X}} \sup_{X'\in\mathcal{X}_X^{(1)}} \left(-\frac{\epsilon}{2s}+\frac{\tilde{\boldsymbol{\mu}}_{XX'}^{\intercal}\tilde{\boldsymbol{\Gamma}}_{XX'}\tilde{\boldsymbol{\mu}}_{XX'}-\ln|\tilde{\boldsymbol{\Gamma}}_{XX'}|}{s}\right)$$
$$= -\frac{\epsilon}{2s}+\frac{1}{s}\sup_{X\in\mathcal{X}} \sup_{X\in\mathcal{X}_X^{(1)}} (\tilde{\boldsymbol{\mu}}_{XX'}^{\intercal}\tilde{\boldsymbol{\Gamma}}_{XX'}\tilde{\boldsymbol{\mu}}_{XX'}-\ln|\tilde{\boldsymbol{\Gamma}}_{XX'}|). \quad (42)$$

We therefore seek to find the noise means $\boldsymbol{\mu}_{\boldsymbol{\eta}|X}$ and covariances $\boldsymbol{\Sigma}_{\boldsymbol{\eta}|X}$ that solve the following problem:

$$\min_{\boldsymbol{\mu}_{\eta|X},\boldsymbol{\Sigma}_{\eta|X},\forall X} \left(\sup_{X\in\mathcal{X}} \sup_{X'\in\mathcal{X}_X^{(1)}} \tilde{\boldsymbol{\mu}}_{XX'}^{\intercal}\tilde{\boldsymbol{\Gamma}}_{XX'}\tilde{\boldsymbol{\mu}}_{XX'}-\ln|\tilde{\boldsymbol{\Gamma}}_{XX'}|\right),$$

$$\text{subj. to } \tilde{\boldsymbol{\mu}}_X = \boldsymbol{\mu}_X+\boldsymbol{\mu}_{\boldsymbol{\eta}|X}, \tilde{\boldsymbol{\Sigma}}_X = \boldsymbol{\Sigma}_X+\boldsymbol{\Sigma}_{\boldsymbol{\eta}|X}, \quad (43)$$

$$\rho = \frac{1}{k}\mathrm{Tr}(\boldsymbol{\Sigma}_{\boldsymbol{\eta}|X})+\frac{1}{k}\|\boldsymbol{\mu}_{\boldsymbol{\eta}|X}\|^2, \quad (44)$$

For the time being, we assume that noise means $\boldsymbol{\mu}_{\boldsymbol{\eta}|X} = \boldsymbol{0}$. Expanding the terms within the sup, we get,

$$\tilde{\boldsymbol{\mu}}_{XX'}^{\intercal}\tilde{\boldsymbol{\Gamma}}_{XX'}\tilde{\boldsymbol{\mu}}_{XX'}-\ln|\tilde{\boldsymbol{\Gamma}}_{XX'}| = (\tilde{\boldsymbol{\mu}}_{X'}-\tilde{\boldsymbol{\mu}}_X)^{\intercal}\tilde{\boldsymbol{\Sigma}}_{X'}^{-1}(\tilde{\boldsymbol{\mu}}_{X'}-\tilde{\boldsymbol{\mu}}_X)-\ln\frac{|\tilde{\boldsymbol{\Sigma}}_X|}{|\tilde{\boldsymbol{\Sigma}}_{X'}|}$$
$$= (\tilde{\boldsymbol{\mu}}_{X'}-\tilde{\boldsymbol{\mu}}_X)^{\intercal}(\boldsymbol{\Sigma}_{X'}+\boldsymbol{\Sigma}_{\boldsymbol{\eta}|X'})^{-1}(\tilde{\boldsymbol{\mu}}_{X'}-\tilde{\boldsymbol{\mu}}_X)$$
$$-\ln\frac{|\boldsymbol{\Sigma}_X+\boldsymbol{\Sigma}_{\boldsymbol{\eta}|X}|}{|\boldsymbol{\Sigma}_{X'}+\boldsymbol{\Sigma}_{\boldsymbol{\eta}|X'}|} \quad (45)$$

The surrogate metric in eq. (45) is convex in $\tilde{\boldsymbol{\Sigma}}_X$ when all the other terms are kept constant. It is also convex in the variable $\tilde{\boldsymbol{\Sigma}}_{X'}^{-1}$. We continue the discussion by setting the optimization variable as $\boldsymbol{A}_X \triangleq \tilde{\boldsymbol{\Sigma}}_X^{-1}$ and later estimate $\boldsymbol{\Sigma}_{\boldsymbol{\eta}|X}$ by subtracting $\boldsymbol{\Sigma}_X$ from $\tilde{\boldsymbol{\Sigma}}_X$ and projecting onto the set of positive semi-definite (PSD) matrices.

Now, let us focus on the overall cost in the optimization problem in eq. (44). The terms are $\forall X\in\mathcal{X},\ X'\in\mathcal{X}_X^{(1)}$:

$$g_{XX'} \triangleq (\tilde{\boldsymbol{\mu}}_{X'}-\tilde{\boldsymbol{\mu}}_X)^{\intercal}\boldsymbol{A}_{X'}(\tilde{\boldsymbol{\mu}}_{X'}-\tilde{\boldsymbol{\mu}}_X)-\ln\frac{|\boldsymbol{A}_{X'}|}{|\boldsymbol{A}_X|}. \quad (46)$$

The cost associated with the minimization then becomes

$$J = \sup_{X} \sup_{X'\in\mathcal{X}_X^{(1)}} g_{XX'} \quad (47)$$

Note that there is no symmetry in the surrogate metric, $g_{XX'} \neq g_{X'X}$ and therefore, we need to consider them separately. One can think of a block-coordinate descent algorithm in order to reach a minima (local or global). The algorithm we propose is given in Algorithm 1. Having obtained an $\boldsymbol{A}_X$, one can determine the noise covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\eta}|X}$ such that the constraint $\mathrm{Tr}(\boldsymbol{\Sigma}_{\boldsymbol{\eta}|X})$ is satisfied as follows:

$$\boldsymbol{\Sigma}_{\boldsymbol{\eta}|X} = \frac{\rho\ \mathrm{Proj}_{\mathcal{S}_k^+}\{(\boldsymbol{A}_X^{\star})^{-1}-\boldsymbol{\Sigma}_X\}}{\mathrm{Tr}\left(\mathrm{Proj}_{\mathcal{S}_k^+}\{(\boldsymbol{A}_X^{\star})^{-1}-\boldsymbol{\Sigma}_X\}\right)} \quad (50)$$

where $\mathrm{Proj}_{\mathcal{S}_k^+}(\boldsymbol{X})$ is the projection of the matrix $\boldsymbol{X}$ onto the set of positive semi-definite matrices. We know that the optimization problem of interest is a non-convex one and what we seek is an improvement over the standard Gaussian mechanism, which amounts to adding white noise to the query response. Our algorithm initializes the noise covariance matrices to be white. Then, the minimization we perform with

Algorithm 1: Estimation of $\boldsymbol{A}_{X} = \tilde{\boldsymbol{\Sigma}}_{X}^{-1}, \forall X \in \mathcal{X}$

Result: $\boldsymbol{A}_{X}^{\star}, \quad \forall X \in \mathcal{X}$

Initialize $\tilde{\boldsymbol{\Sigma}}_{X}^{(0)} = \boldsymbol{\Sigma}_{X} + (\rho/k)\boldsymbol{I}$, $\alpha_{X}^{(0)} = 0.1 \ \forall X$ and $\boldsymbol{A}_{X}^{(0)} = \left(\tilde{\boldsymbol{\Sigma}}_{X}^{(0)}\right)^{-1}$ for all $X \in \mathcal{X}, t = 0$.

while $\alpha_{X_\bullet}^{(t)} > 0$ **do**

Identify pair of labels:

$$(X_\bullet, X_\bullet') = \arg_{(X,X')} \sup_{X} \sup_{X' \in \mathcal{X}_X^{(1)}} \{g_{XX'}(t)\}.$$

Compute $J^{(t)} = g_{X_\bullet X_\bullet'}(t)$ using $(\boldsymbol{A}_{X_\bullet}^{(t)}, \boldsymbol{A}_{X_\bullet}^{(t)})$ in eq. (46).

Update $\boldsymbol{A}_{X_\bullet}^{(t+1)}$ with $\alpha_{X_\bullet}^{(t)}$ in Lemma 2:

$$\boldsymbol{A}_{X_\bullet}^{(t+1)} \leftarrow \boldsymbol{A}_{X_\bullet}^{(t)} - \alpha_{X_\bullet}^{(t)} \left[\nabla_{\boldsymbol{A}_{X_\bullet}} g_{X_\bullet X_\bullet}(t)\right], \tag{48}$$

where

$$\nabla_{\boldsymbol{A}_{X_\bullet}} g_{X_\bullet X_\bullet}(t) = (\tilde{\boldsymbol{\mu}}_{X_\bullet}^{(t)} - \tilde{\boldsymbol{\mu}}_{X_\bullet}^{(t)})(\tilde{\boldsymbol{\mu}}_{X_\bullet}^{(t)} - \tilde{\boldsymbol{\mu}}_{X_\bullet}^{(t)})^{\intercal} - (\boldsymbol{A}_{X_\bullet}^{(t)})^{-1}.$$

Project $\boldsymbol{A}_{X_\bullet(t)}(t+1)$ onto the set of positive-semi definite matrices:

$$\boldsymbol{A}_{X_\bullet}^{(t+1)} \leftarrow \text{Proj}_{\mathcal{S}_k^+}\{\boldsymbol{A}_{X_\bullet}^{(t+1)}\}. \tag{49}$$

$$t = t + 1$$

respect to different variables is such that there is always a *decrease* in the cost function. For this, it is pertinent that we choose the appropriate step-size $\alpha_{X_\bullet(t)}$ in order to converge to a local minimum.

Lemma 2. *Let $J^{(t)} = g_{X_\bullet X_\bullet}(t)$ and the variable to be updated $\boldsymbol{A}_{X_\bullet}^{(t)}$. If the step size $\alpha_{X_\bullet^{(t)}}$ at iteration t satisfies:*

$$\alpha_{X_\bullet}^{(t)} \leq \quad \max\left\{0, \min_{X \in \mathcal{X}_{X_\bullet(t)}^{(1)}} \{b_{X_\bullet X}(t), d_{X_\bullet X}(t)\}\right\} \tag{51}$$

then, the update to $\boldsymbol{A}_{X_\bullet}^{(t)}$ given in eq. (48) will lead to $J^{(t)} - J^{(t+1)} \geq 0$. The constants in eq. (51) are,

$$b_{X_\bullet X}(t) = \frac{\tilde{\boldsymbol{v}}_{XX_\bullet}^{\intercal} \boldsymbol{A}_{X_\bullet}^{(t)} \tilde{\boldsymbol{v}}_{XX_\bullet} - \tilde{\boldsymbol{v}}_{XX_\bullet}^{\intercal} \boldsymbol{A}_{X}^{(t)} \tilde{\boldsymbol{v}}_{XX_\bullet}}{\text{Tr}\left((\boldsymbol{A}_{X_\bullet}^{(t)})^{-2}\right) - \tilde{\boldsymbol{v}}_{XX_\bullet}^{\intercal} (\boldsymbol{A}_{X_\bullet}^{(t)})^{-1} \tilde{\boldsymbol{v}}_{XX_\bullet}} + \frac{\ln|\boldsymbol{A}_{X_\bullet}^{(t)}| + \ln|\boldsymbol{A}_{X}^{(t)}| - 2\ln|\boldsymbol{A}_{X_\bullet}^{(t)}|}{\text{Tr}\left((\boldsymbol{A}_{X_\bullet}^{(t)})^{-2}\right) - \tilde{\boldsymbol{v}}_{XX_\bullet}^{\intercal} (\boldsymbol{A}_{X_\bullet}^{(t)})^{-1} \tilde{\boldsymbol{v}}_{XX_\bullet}}, \tag{52}$$

$$d_{X_\bullet X}(t) = \frac{\tilde{\boldsymbol{v}}_{XX_\bullet}^{\intercal} \boldsymbol{A}_{X_\bullet}^{(t)} \tilde{\boldsymbol{v}}_{XX_\bullet} + \ln|\boldsymbol{A}_{X_\bullet}^{(t)}| - \ln|\boldsymbol{A}_{X}^{(t)}|}{\tilde{\boldsymbol{v}}_{XX_\bullet}^{\intercal} (\boldsymbol{A}_{X_\bullet}^{(t)})^{-1} \tilde{\boldsymbol{v}}_{XX_\bullet} - \text{Tr}\left((\boldsymbol{A}_{X_\bullet}^{(t)})^{-2}\right) - (\tilde{\boldsymbol{v}}_{XX_\bullet}^{\intercal} \tilde{\boldsymbol{v}}_{XX_\bullet})^2}, \tag{53}$$

where $\tilde{\boldsymbol{v}}_{XX'} = (\tilde{\boldsymbol{\mu}}_{X'} - \tilde{\boldsymbol{\mu}}_{X})$.

Proof. Proof can be found in Appendix C. □

V. Application to ARMA forecasts

In this section, we first discuss how one could apply the proposed DP method for class-based data to stationary Gaussian processes, where each class-label X is associated

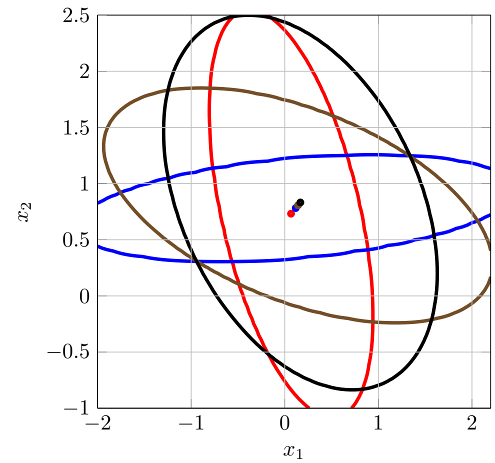

Fig. 1. The contour plot of synthetic data.

with a Gaussian process with certain mean and covariance. Then we discuss DP-based ARMA forecasts, where each class has a different set of ARMA parameters. Finally, we outline how one could produce DP power consumption forecasts while hiding the identity of a certain household.

Let $x[k]$ be a stochastic process, whose statistics depend on a label X we want to conceal; let us define the vector $\boldsymbol{x}$ of samples that contains both K observed samples $\boldsymbol{x}^o$ and T future samples $\boldsymbol{x}^f$ whose forecast we want to share without revealing the label X, i.e.:

$$\boldsymbol{x} = [x[0], \ldots, x[K+T-1]]^{\intercal} = \begin{bmatrix} \boldsymbol{x}^o \\ \boldsymbol{x}^f \end{bmatrix}. \tag{54}$$

The query is the forecast, $\boldsymbol{x}^f$. A well known result from statistics is that the minimum-mean squared error (MMSE) estimator of the forecast is the conditional expectation of $\boldsymbol{x}^f$, given $\boldsymbol{x}^o$, i.e.:

$$\boldsymbol{q} \equiv \hat{\boldsymbol{x}}^f = \mathbb{E}[\boldsymbol{x}^f | \boldsymbol{x}^o]. \tag{55}$$

For a Gaussian process the statistics of $\boldsymbol{x}^f$ conditioned on $\boldsymbol{x}^o$ are normal as well. To compute the mean and covariance of the conditional mean we need to also define the covariance of $\boldsymbol{x}$:

$$\mathbb{E}[\boldsymbol{x}] = \begin{bmatrix} \boldsymbol{\mu}_X^o \\ \boldsymbol{\mu}_X^f \end{bmatrix}, \quad \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top] = \begin{bmatrix} \boldsymbol{\Sigma}_X^o & \boldsymbol{\Sigma}_X^{fo} \\ \boldsymbol{\Sigma}_X^{of} & \boldsymbol{\Sigma}_X^f \end{bmatrix}. \tag{56}$$

This implies that for the query $\boldsymbol{q}$ under the class-label X, the mean and covariance are :

$$\begin{aligned} \boldsymbol{\mu}_X &= \boldsymbol{\mu}_X^f + \boldsymbol{\Sigma}_X^{fo}(\boldsymbol{\Sigma}_X^o)^{-1}(\boldsymbol{x}^o - \boldsymbol{\mu}_X^o) \\ \boldsymbol{\Sigma}_X &= \boldsymbol{\Sigma}_X^f - \boldsymbol{\Sigma}_X^{fo}(\boldsymbol{\Sigma}_X^o)^{-1}\boldsymbol{\Sigma}_X^{of} \end{aligned} \tag{57}$$

At this point, it should be clear that it is possible to adopt the scheme we proposed to share a DP forecast for a Gaussian process. Note that the same formulas provide the optimum linear-least square forecast. The typical situation is that the covariance and mean are not known and need to be estimated from the data. This is why it is popular to assume a parametric model for the second and first order statistics. The ARMA model for the second order statistics is an example.

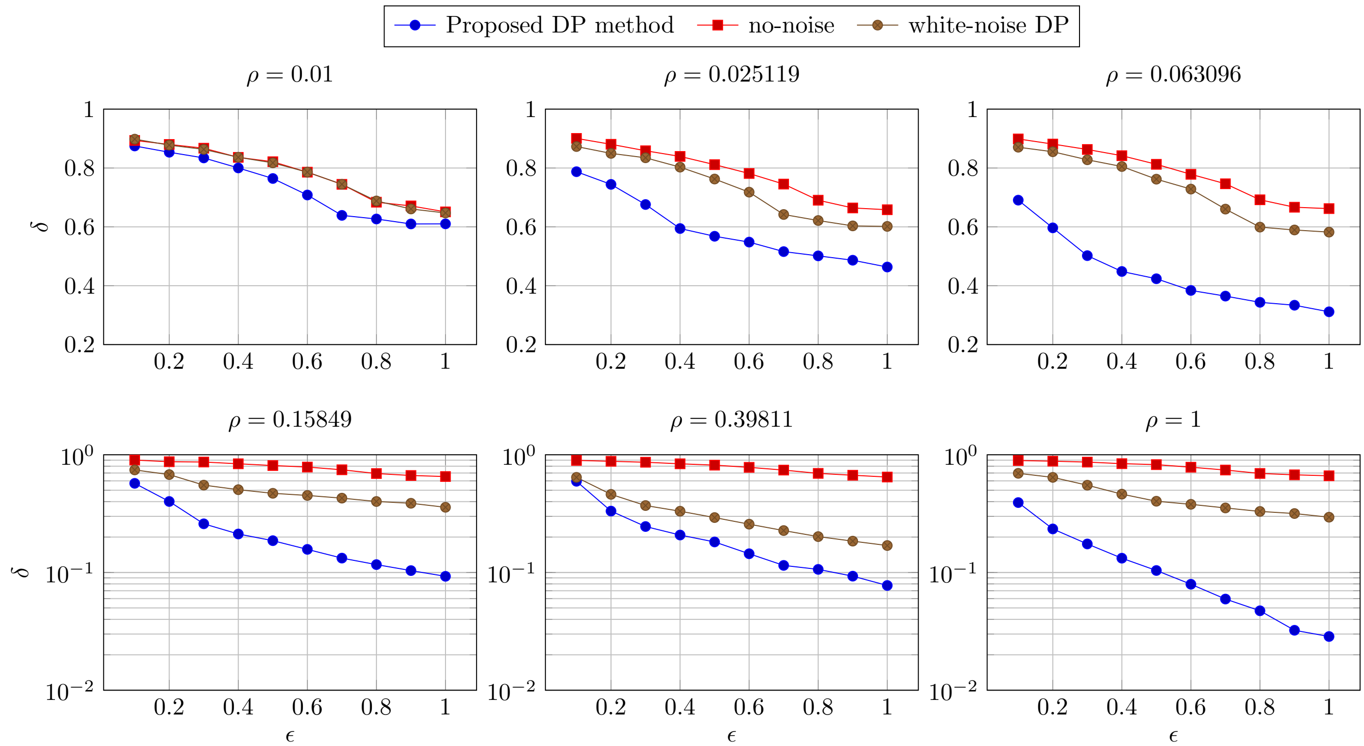

Fig. 2. The (ϵ, δ)-private curves with different **accuracy budgets** (see Def. 3) ρ for the synthetic data.

A. ARMA forecasts

An ARMA filter is a discrete time filter whose frequency response can be parametrized as follows:

$$H(e^{-j\omega}) = \frac{\sum_{k=1}^{m} a_k e^{-j\omega k}}{\sum_{k=0}^{n} b_k e^{-j\omega k}}. \tag{58}$$

An ARMA process is a zero mean stationary process that is generated by filtering with an ARMA filter i.i.d. zero mean noise $\xi[k]$ with standard deviation equal to 1; equivalently,

$$x[k] = -\sum_{i=1}^{m} a_i x[k-i] + \sum_{i=0}^{n} b_i \xi[k-i]. \tag{59}$$

If the process is not zero mean, then $x[k]$ models the residual after recentering the actual process by subtracting the mean. From the estimate of the parameters of the ARMA filter one can calculate the impulse response $h[k]$ whose frequency response is given in eq. (58) (we skip the expression for brevity). With that, one can construct the Toeplitz matrix of the covariance matrix whose ik^{th} entry is

$$\left[\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]\right]_{ik} = \sum_{i=-\infty}^{+\infty} h[i]h[k-i] \tag{60}$$

and proceed to calculate the forecast. Note that it is possible to answer the query or compute $\boldsymbol{q}$ using the filter defined above. However, since our method of numerical calculation relies on the covariance of the process, we rely on the expression in eq. (57) to compute the optimum noise distribution for the forecast $\boldsymbol{q}$. We summarize the process of adding optimal DP noise to ARMA forecasts in Algorithm 2.

Algorithm 2: Addition of DP noise to ARMA forecasts

1. Compute the ARMA forecasting covariance matrix $\boldsymbol{\Sigma}_x$ in eq. (57) $\forall X \in \mathcal{X}$.
2. Estimate $\boldsymbol{A}_x \forall X \in \mathcal{X}$ according to Algorithm 1.
3. Obtain optimal DP noise covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\eta}|x}$ in eq. (50).
4. Add the optimal DP noise to the query $\boldsymbol{q}$, i.e., $\tilde{\boldsymbol{q}} = \boldsymbol{q} + \boldsymbol{\eta}$ where $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}|x})$.

B. Application to power measurements data

To validate our method in the numerical section, we will consider the case in which the data queried is electric load consumption from a specific household. Daily patterns of power consumption, widely available now to utilities, do not fit a Gaussian multivariate distribution directly. In prior work [30], however, we showed that they fit well a multivariate log-Normal distribution, which implies that the logarithm of the daily pattern is a multivariate Gaussian vector, which is what we need to apply our method. Let $p[k]$ be the power consumption at hour k during the day. The model we adopt is as follows:

$$x[k] = \log(p[k]) - \mu[k] \tag{61}$$

where $\mu[k]$ is the seasonal mean estimated averaging $\log(p[k])$ over each day so that $x[k]$ is zero mean[1]. The assumption we make is that $x[k]$ is an ARMA zero mean stationary Gaussian process, which allows us to apply the scheme we proposed in

[1]This is not exactly the case since we are dealing with estimates but is our modeling assumption.

the previous sections. Note that here the query is the forecast of the samples $p[k],\ k = K, \dots, K+T$ which are not Gaussian, i.e.:

$$\begin{aligned} \boldsymbol{q} &= [\hat{p}[K], \dots, \hat{p}[K+T-1]]^\top \\ \hat{p}[k] &= e^{\hat{x}[k]+\mu[k]}, \quad k = K, \dots, K+T-1. \end{aligned} \tag{62}$$

Nonetheless the DP answer $\tilde{\boldsymbol{q}}$ can be computed by applying the optimum Gaussian noise to the samples of the process $\hat{x}[k] + \mu[k], k = K, \dots, K+T-1$ and then applying the exponential function to them. Since the exponential function is a bijective function, the PDP (ϵ, δ) privacy guarantee is maintained after this type of post-processing. What we can no longer guarantee is that the accuracy of $\tilde{\boldsymbol{q}}$ is $\rho^{\text{MSE}}_{\mathbb{Q}|X}$ in eq. (13). The resulting accuracy can be calculated, but we skip the derivation for brevity.

VI. Numerical Results

In this section, we illustrate the proposed algorithm with a synthetic dataset and real-world power system AMI data. In the synthetic case, there are several classes and the query corresponding to each class is bi-dimensional Gaussian with a mean vector and 2×2 covariance matrix. It is assumed that each class is the neighbor of the other class. The goal is to conceal the class-label by adding DP noise to the query output.

In the real-world AMI case, we first map the forecast of this seasonal and non-Gaussian process onto an ARMA forecasting problem, and then utilize the $(m = 6, n = 5)$ ARMA model for training and testing. Finally, we utilize the proposed optimal DP noise to protect the ARMA forecasts from households that belong to the same cluster.

A. Synthetic Data

The synthetic data is generated assuming that there are four classes and all classes are considered as neighbors of the others. The query corresponding to each class follows a two-dimensional Gaussian distribution with mean and covariance matrix. The mean vectors per-class are chosen to lie on a line in the 2-D space, and the covariance matrix is drawn from a Wishart distribution with scale matrix as the identity and two degrees of freedom. This is shown as a contour plot in Figure 1. Such a case was chosen to highlight the advantage of the proposed method over adding white noise with variance to meet the accuracy requirement.

We compare our method with two baselines. Firstly, the "no-noise" baseline to understand DP guarantee if no noise was added to the query. One could still expect some inherent privacy, solely due to the fact that the query itself is stochastic and the distributions corresponding to each class-label could be close. Our method starts at this baseline and further improves the privacy guarantee. The second baseline is the addition of white Gaussian noise to synthetic data with zero-mean denoted by "white noise DP" where the variance of noise $\sigma^2 = \rho/D$, so that the accuracy requirement in eq. (13) is met. Note that the addition of zero-mean white Gaussian noise is commonplace in DP literature, with more focus on choosing the appropriate variance σ^2. One can think of the baseline as setting of variance in order to meet the accuracy requirement.

We plot the simulation results in Figure 2 with varying levels of accuracy ρ. Note that the no-noise baseline does not depend on the accuracy, and is therefore the same curve throughout. The proposed method outperforms the white-noise baseline irrespective of the accuracy level, although the improvement decreases slightly with larger ρ values. The privacy-accuracy (utility) trade-off is visible in the curves where larger ρ values are associated with better privacy (ϵ, δ) values.

B. AMI Data Forecasting by ARMA Model

1) ARMA Forecasting: First, we collect 100 households' demands, and then remove the seasonal values (in a weekly scale) from the original demands, and further take the log values of the nonseasonal demands for training, as described in subsection V-B. Then, we utilize the k-means algorithm to cluster the 100 household's demands into six clusters. In each cluster, we consider each household as the neighbor of all other homes. Specifically, we choose the 15-day data for training to estimate a_i and b_j parameters in eq. (59), where the 15-day data have 1440 samples such that the sampling rate is once per 15 minutes. The forecasting time is three hours with 12 samples, i.e., the size of the query is 12. The forecast trajectory of $x[k]$ of many households are shown in Figure 3. The gray lines show the past 24 hours' demands, and the blue lines show the forecast three hours' demands, and the red lines show the ground-truth three hours' demands. In Figure 3, the MSE of House 1, House 2 and House 3 are $2.6986e^{-4}$, $2.3339e^{-4}$ and $1.4769e^{-4}$, respectively. The results show that the ARMA model forecasts the future demands with high accuracy, which provides opportunities for analysts to infer the true demands.

2) Differential Privacy for ARMA Forecasting: As in the synthetic data case, we show the simulation results with different ρ in Figure 4. Without noise, δ is always 1, which means that the probability of ARMA forecasted demand being associated with a certain household (class-label) is 100%. Therefore, it is necessary to design the DP noise for the query of ARMA forecasting. With the white noise, δ is reduced to 0.8 when the accuracy $\rho = 0.68571$ and $\epsilon = 0.1$. With ρ fixed and ϵ increasing from 0.1 to 1.0, δ decreases slowly from 0.8 to 0.6. In contrast, with the same $\rho = 0.68571$ and the ϵ increasing from 0.1 to 1.0, δ of the proposed optimal DP algorithm decreases fast from 0.72 to 0.35. It indicates that the proposed algorithm has much better performance regarding the privacy protection. Similar to the synthetic data, with ρ increasing (i.e., relaxing the accuracy level), the (ϵ, δ) privacy is strengthened. Another observation is that with the accuracy budget ρ increasing from 0.45 to 1.4 and with $\epsilon = 1$ fixed, the δ of the proposed DP algorithm decreases from 0.84 to 0.35, which shows a trade-off between accuracy and privacy. Note that the proposed method offers better guarantee than the white-noise DP method, especially when the accuracy constraint is stronger, i.e., ρ is small. This is quite important considering the utility of the query while also concealing the class label.

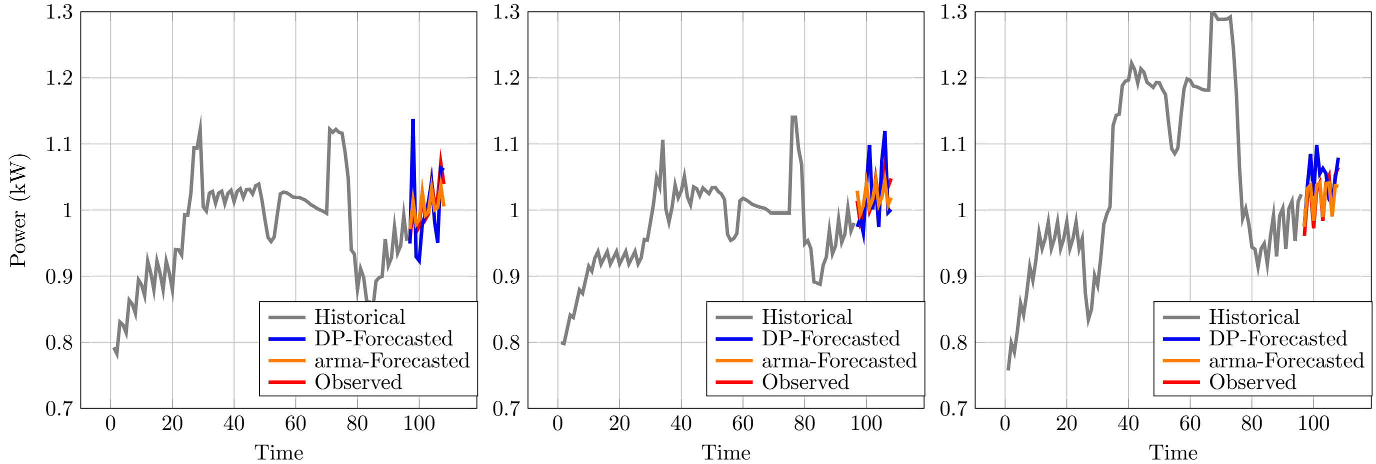

Fig. 3. ARMA Forecasting for power demands of three houses.

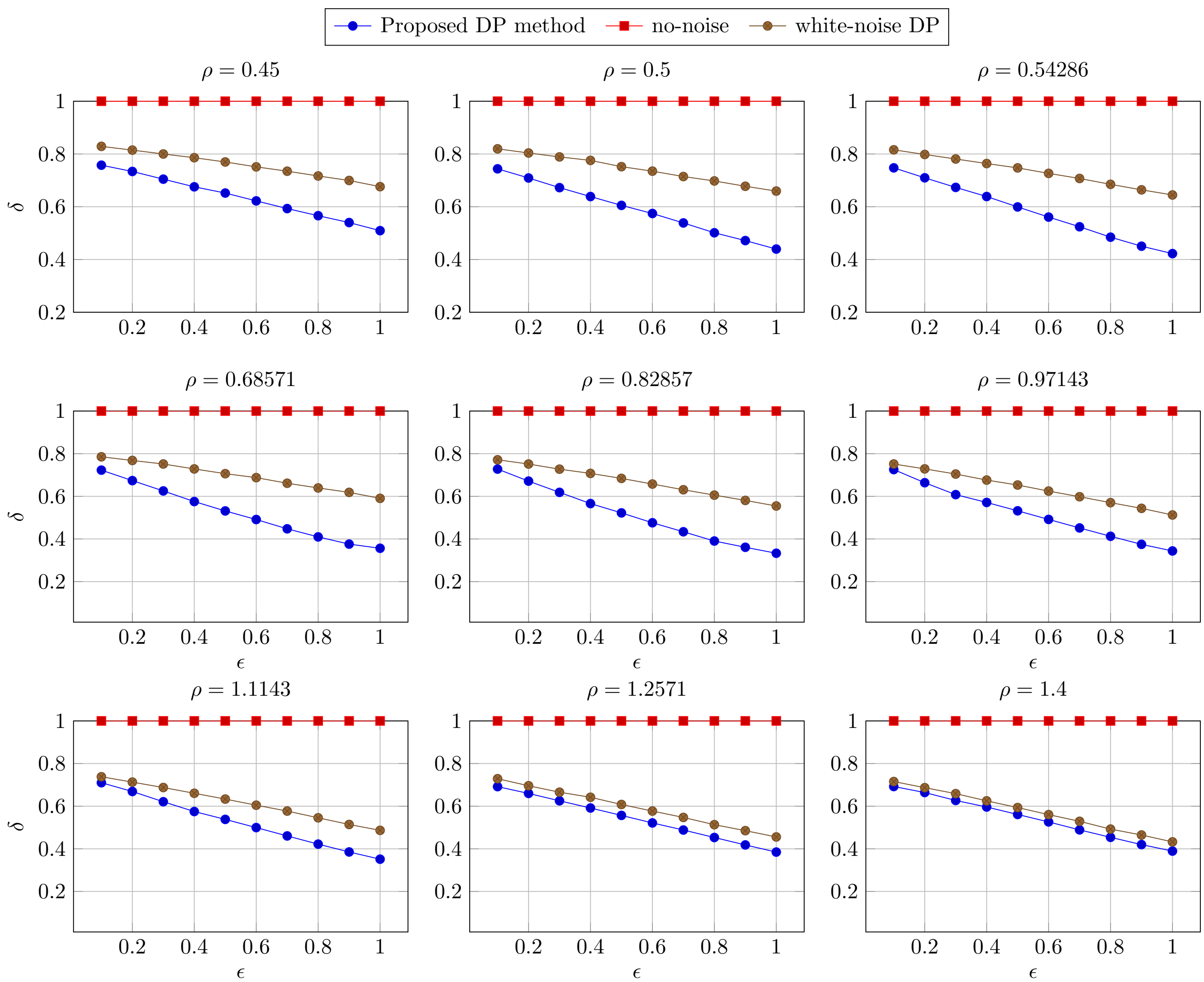

Fig. 4. The (ϵ, δ)-private curves with different accuracy budgets ρ for the real-world AMI data.

VII. Conclusion

In this paper, we proposed a new method of differential privacy for label or class-based data, aiming at adding a functional DP noise for the release of query response such that the analyst is unable to infer the underlying class label. Moreover, we demonstrated that the (ϵ, δ)-private is guaranteed to be satisfied while meeting the predefined accuracy budget ρ. We illustrate the effectiveness of the proposed method on the synthetic data, which outperforms the baseline additive white Gaussian noise mechanism. We further consider the ARMA forecasting problem for the AMI measurements. Then, we implement the proposed DP method to the ARMA forecasting method in order to protect the privacy of the AMI measurements forecasts. Our empirical case studies on both the synthetic data and real-world AMI measurements validate the effectiveness and advantages of the proposed method.

Appendix

A. Proof of Proposition 1

Proof. First, note that we can arrange the terms in $L_{xx'}(\boldsymbol{q})$ as:

$$\begin{aligned} L_{xx'}(\boldsymbol{q}) =& \frac{1}{2}\ln|\boldsymbol{\Sigma}_{x'}| - \frac{1}{2}\ln|\boldsymbol{\Sigma}_x| \\ &+ \frac{1}{2}(\boldsymbol{q}-\boldsymbol{\mu}_x)^\intercal(\boldsymbol{\Sigma}_{x'}^{-1}-\boldsymbol{\Sigma}_x^{-1})(\boldsymbol{q}-\boldsymbol{\mu}_x) \\ &+ (\boldsymbol{\mu}_x-\boldsymbol{\mu}_{x'})^\intercal\boldsymbol{\Sigma}_{x'}^{-1}(\boldsymbol{q}-\boldsymbol{\mu}_x) \\ &+ \frac{1}{2}(\boldsymbol{\mu}_{x'}-\boldsymbol{\mu}_x)^\intercal\boldsymbol{\Sigma}_{x'}^{-1}(\boldsymbol{\mu}_{x'}-\boldsymbol{\mu}_x). \end{aligned} \tag{63}$$

For simplicity, we omit the suffix XX'. First, eq. (28) re-centers, whitens and then rotates the normal vector $\boldsymbol{q}$, which results in the i.i.d normal vector $\boldsymbol{\xi}$. To derive the expression of $L(\boldsymbol{\xi})$ from eq. (63) we observe that:

$$\begin{aligned} (\boldsymbol{\Sigma}_{x'}^{-1}-\boldsymbol{\Sigma}_x^{-1}) &= \boldsymbol{\Sigma}_x^{-1/2}(\boldsymbol{\Sigma}_x^{1/2}\boldsymbol{\Sigma}_{x'}^{-1}\boldsymbol{\Sigma}_x^{1/2}-\boldsymbol{I})\boldsymbol{\Sigma}_x^{-1/2} \\ &= \boldsymbol{\Sigma}_x^{-1/2}\boldsymbol{U}\boldsymbol{U}^\intercal(\boldsymbol{\Sigma}_x^{1/2}\boldsymbol{\Sigma}_{X'}^{-1}\boldsymbol{\Sigma}_x^{1/2}-\boldsymbol{I})\boldsymbol{U}\boldsymbol{U}^\intercal\boldsymbol{\Sigma}_x^{-1/2} \\ &= \boldsymbol{\Sigma}_x^{-1/2}\boldsymbol{U}(\boldsymbol{\Gamma}-I)\boldsymbol{U}^\intercal\boldsymbol{\Sigma}_x^{-1/2} \end{aligned}$$

which implies that

$$(\boldsymbol{q}-\boldsymbol{\mu}_x)^\intercal(\boldsymbol{\Sigma}_{x'}^{-1}-\boldsymbol{\Sigma}_x^{-1})(\boldsymbol{q}-\boldsymbol{\mu}_x) = \boldsymbol{\xi}^\intercal(\boldsymbol{\Gamma}-I)\boldsymbol{\xi}.$$

The other two terms are obtained similarly:

$$\begin{aligned} (\boldsymbol{\mu}_x-\boldsymbol{\mu}_{x'})^\intercal\boldsymbol{\Sigma}_{x'}^{-1}(\boldsymbol{q}-\boldsymbol{\mu}_x) &= (\boldsymbol{\mu}_x-\boldsymbol{\mu}_{x'})^\intercal\boldsymbol{\Sigma}_x^{-1/2}\boldsymbol{U} \\ \cdot\boldsymbol{U}^\intercal(\boldsymbol{\Sigma}_x^{1/2}\boldsymbol{\Sigma}_{x'}^{-1}\boldsymbol{\Sigma}_x^{1/2})\boldsymbol{U}\boldsymbol{U}^\intercal\boldsymbol{\Sigma}_x^{-/2}(\boldsymbol{q}-\boldsymbol{\mu}_x) &= -\boldsymbol{\mu}^\intercal\boldsymbol{\Gamma}\boldsymbol{\xi}. \end{aligned}$$

$$\begin{aligned} (\boldsymbol{\mu}_{x'}-\boldsymbol{\mu}_x)^\intercal\boldsymbol{\Sigma}_{x'}^{-1}(\boldsymbol{\mu}_{x'}-\boldsymbol{\mu}_x) &= (\boldsymbol{\mu}_{x'}-\boldsymbol{\mu}_x)^\intercal\boldsymbol{\Sigma}_x^{-1/2}\boldsymbol{U} \\ \cdot\boldsymbol{U}^\intercal(\boldsymbol{\Sigma}_x^{1/2}\boldsymbol{\Sigma}_{x'}^{-1}\boldsymbol{\Sigma}_x^{1/2})\boldsymbol{U}\boldsymbol{U}^\intercal\boldsymbol{\Sigma}_x^{-1/2}(\boldsymbol{\mu}_{x'}-\boldsymbol{\mu}_x) &= \boldsymbol{\mu}^\intercal\boldsymbol{\Gamma}\boldsymbol{\mu}. \end{aligned}$$

□

B. Proof of Lemma 1

Proof. Again, we omit the suffixes XX' to streamline the notation. The idea is to leverage Proposition 1 and find a bound for $Pr(L(\boldsymbol{\xi})>\epsilon)$ through the Chernoff bound. More specifically, considering the expression eq. (29), for $\overline{s}>0$:

$$\begin{aligned} Pr(L(\boldsymbol{\xi})>\epsilon) &= Pr(e^{\frac{\overline{s}}{2}(\boldsymbol{\xi}^\intercal(\boldsymbol{\Gamma}-\boldsymbol{I})\boldsymbol{\xi}-2\boldsymbol{\mu}\boldsymbol{\Gamma}\boldsymbol{\xi}+\boldsymbol{\mu}^\intercal\boldsymbol{\Gamma}\boldsymbol{\mu})} > e^{\overline{s}\epsilon+\frac{\overline{s}}{2}\ln|\boldsymbol{\Gamma}|}) \\ &\leq \frac{\mathbb{E}\left[e^{\frac{\overline{s}}{2}(\boldsymbol{\xi}^\intercal(\boldsymbol{\Gamma}-\boldsymbol{I})\boldsymbol{\xi}-2\boldsymbol{\mu}\boldsymbol{\Gamma}\boldsymbol{\xi}+\boldsymbol{\mu}^\intercal\boldsymbol{\Gamma}\boldsymbol{\mu})}\right]}{e^{\overline{s}\epsilon}|\boldsymbol{\Gamma}|^{\overline{s}/2}} \end{aligned}$$

The expectation $\mathfrak{J}(\boldsymbol{\Gamma},\boldsymbol{\mu}) \triangleq \mathbb{E}[e^{\frac{\overline{s}}{2}(\boldsymbol{\xi}^\intercal(\boldsymbol{\Gamma}-\boldsymbol{I})\boldsymbol{\xi}-2\boldsymbol{\mu}^\intercal\boldsymbol{\Gamma}\boldsymbol{\xi}+\boldsymbol{\mu}^\intercal\boldsymbol{\Gamma}\boldsymbol{\mu})}]$ can be evaluated in closed form solving the following integral:

$$\mathfrak{J}(\boldsymbol{\Gamma},\boldsymbol{\mu}) = \int \frac{e^{\frac{\overline{s}}{2}(\boldsymbol{\xi}^\intercal(\boldsymbol{\Gamma}-\boldsymbol{I})\boldsymbol{\xi}-2\boldsymbol{\mu}^\intercal\boldsymbol{\Gamma}\boldsymbol{\xi}+\boldsymbol{\mu}^\intercal\boldsymbol{\Gamma}\boldsymbol{\mu})-\frac{\boldsymbol{\xi}^\intercal\boldsymbol{\xi}}{2}}}{(2\pi)^{k/2}}\,\mathrm{d}\boldsymbol{\xi}. \tag{64}$$

The goal is to express the exponent as a quadratic form. Combining the quadratic terms and extracting the factor $-\frac{\overline{s}}{2}$ the expression becomes:

$$-\frac{\overline{s}}{2}\left(\boldsymbol{\xi}^\intercal\left((1+\overline{s}^{-1})\boldsymbol{I}-\boldsymbol{\Gamma}\right)\boldsymbol{\xi}+2\boldsymbol{\mu}^\intercal\boldsymbol{\Gamma}\boldsymbol{\xi}-\boldsymbol{\mu}^\intercal\boldsymbol{\Gamma}\boldsymbol{\mu}\right)$$

The expression is streamlined by the following substitution for $\overline{s}$ and definition of $\hat{\boldsymbol{\mu}}$:

$$s = 1+\overline{s}^{-1} \;\Rightarrow\; \overline{s} = (s-1)^{-1}, \overline{s}>0 \implies s>1 \tag{65}$$

$$\hat{\boldsymbol{\mu}} \triangleq -(s\boldsymbol{I}-\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}\boldsymbol{\mu}. \tag{66}$$

so that the exponent can then be rearranged as follows:

$$\begin{aligned} &-\frac{\boldsymbol{\xi}^\intercal(s\boldsymbol{I}-\boldsymbol{\Gamma})\boldsymbol{\xi}-2\hat{\boldsymbol{\mu}}^\intercal(s\boldsymbol{I}-\boldsymbol{\Gamma})\boldsymbol{\xi}\pm\hat{\boldsymbol{\mu}}^\intercal(s\boldsymbol{I}-\boldsymbol{\Gamma})\hat{\boldsymbol{\mu}}-\boldsymbol{\mu}^\intercal\boldsymbol{\Gamma}\boldsymbol{\mu}}{2(s-1)} = \\ &-\frac{(\boldsymbol{\xi}-\hat{\boldsymbol{\mu}})^\intercal(s\boldsymbol{I}-\boldsymbol{\Gamma})(\boldsymbol{\xi}-\hat{\boldsymbol{\mu}})}{2(s-1)}+\frac{s}{2(s-1)}\boldsymbol{\mu}^\intercal(s\boldsymbol{I}-\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}\boldsymbol{\mu} \end{aligned}$$

hence, if and only if $s\boldsymbol{I}-\boldsymbol{\Gamma}$ is positive definite, which means that $s>\gamma_1=\lambda_{\max}(\boldsymbol{\Sigma}_x^{1/2}\boldsymbol{\Sigma}_{x'}^{-1}\boldsymbol{\Sigma}_x^{1/2})$, we have:

$$\begin{aligned} \mathfrak{J}(\boldsymbol{\Gamma},\boldsymbol{\mu}) &= \frac{e^{\frac{s}{2(s-1)}\boldsymbol{\mu}^\intercal(s\boldsymbol{I}-\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}\boldsymbol{\mu}}}{(2\pi)^k}\int e^{-\frac{(\boldsymbol{\xi}-\hat{\boldsymbol{\mu}})^\intercal(s\boldsymbol{I}-\boldsymbol{\Gamma})(\boldsymbol{\xi}-\hat{\boldsymbol{\mu}})}{2(s-1)}}\,\mathrm{d}\boldsymbol{\xi} \\ &= \frac{e^{\frac{s}{2(s-1)}\boldsymbol{\mu}^\intercal(s\boldsymbol{I}-\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}\boldsymbol{\mu}}(s-1)^{k/2}}{|s\boldsymbol{I}-\boldsymbol{\Gamma}|^{1/2}} \end{aligned}$$

We can conclude that:

$$Pr(L(\boldsymbol{\xi})>\epsilon) \leq \frac{(s-1)^{k/2}e^{-\frac{\epsilon}{(s-1)}+\frac{s}{2(s-1)}\boldsymbol{\mu}^\intercal(s\boldsymbol{I}-\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}\boldsymbol{\mu}}}{|\boldsymbol{\Gamma}|^{\frac{1}{2(s-1)}}|s\boldsymbol{I}-\boldsymbol{\Gamma}|^{1/2}} \tag{67}$$

where the bound is the expression in eq. (34). □

C. Proof of Lemma 2

Proof. Upon updating $\boldsymbol{A}_{x_\bullet(t)}(t)$, the cost J_{t+1} is,

$$J_{t+1} = \sup_{x}\sup_{x'\in\mathcal{X}_X^{(1)}}\left\{g_{x_\bullet(t)x_\bullet(t)}, g_{xx_\bullet(t)}, g_{x_\bullet(t)x}, g_{xx'}\right\} \tag{68}$$

We now divide the analysis into different cases to derive the condition on the step size.

- If $J_{t+1}=g_{xx'}$ where $x\neq x_\bullet(t), x'\neq x_\bullet(t)$, then $J_t-J_{t+1}\geq 0$ since terms $g_{xx'}$ were unaffected by the update and at step t $g_{x_\bullet(t)x_\bullet(t)}$ was the supremum thus, $J_t-J_{t+1}=g_{x_\bullet(t)x_\bullet(t)}-g_{xx'}\geq 0$.
- If $J_{t+1}=g_{x_\bullet(t)x_\bullet(t)}$, it means that the chosen term to minimize is still the supremum,

$$\begin{aligned} J_t-J_{t+1} = g_{x_\bullet(t)x_\bullet(t)}&(\boldsymbol{A}_{x_\bullet(t)}(t),\boldsymbol{A}_{x_\bullet(t)}(t)) \\ &- g_{x_\bullet(t)x_\bullet(t)}(\boldsymbol{A}_{x_\bullet(t)}(t+1),\boldsymbol{A}_{x_\bullet(t)}(t)) \end{aligned} \tag{69}$$

  Since $g_{x_\bullet(t)x_\bullet(t)}(\boldsymbol{A}_{x_\bullet(t)}(t),\boldsymbol{A}_{x_\bullet(t)}(t))$ is convex in $\boldsymbol{A}_{x_\bullet(t)}(t)$, the update for $\boldsymbol{A}_{x_\bullet(t)}(t)$ ensures that $J_t-J_{t+1}>0$. However, for an appropriate step-size, one can either resort to backtracking line search or use a small fixed step size.

- If $J_{t+1} = g_{\mathsf{x}\mathsf{x}_\bullet(t)}$, to derive the condition for α_t such that $J_t - J_{t+1} > 0$ consider,

$$J_t - J_{t+1} = g_{\mathsf{x}_\bullet(t)\mathsf{x}_\bullet(t)}\left(\boldsymbol{A}_{\mathsf{x}_\bullet(t)}(t), \boldsymbol{A}_{\mathsf{x}_\bullet(t)}(t)\right) \quad (70)$$
$$- g_{\mathsf{x}\mathsf{x}_\bullet(t)}\left(\boldsymbol{A}_{\mathsf{x}_\bullet(t)}(t+1), \boldsymbol{A}_{\mathsf{x}}(t)\right) \quad (71)$$
$$= \tilde{\boldsymbol{v}}_{\mathsf{x}_\bullet \mathsf{x}_\bullet}^\intercal \boldsymbol{A}_{\mathsf{x}_\bullet}^{(t)} \tilde{\boldsymbol{v}}_{\mathsf{x}_\bullet \mathsf{x}_\bullet} - \tilde{\boldsymbol{v}}_{\mathsf{x}\mathsf{x}_\bullet}^\intercal \boldsymbol{A}_{\mathsf{x}_\bullet(t)}(t+1) \tilde{\boldsymbol{v}}_{\mathsf{x}\mathsf{x}_\bullet}$$
$$- \ln|\boldsymbol{A}_{\mathsf{x}_\bullet(t)}(t)| + \ln|\boldsymbol{A}_{\mathsf{x}_\bullet(t)}(t)|$$
$$+ \ln|\boldsymbol{A}_{\mathsf{x}_\bullet(t)}(t+1)| - \ln|\boldsymbol{A}_{\mathsf{x}}(t)| \quad (72)$$

With further simplification and by imposing the condition $J_t - J_{t+1} > 0$, we get

$$\alpha_{\mathsf{x}_\bullet(t)} \leq d_{\mathsf{x}_\bullet(t),\mathsf{x}} \quad (73)$$

- If $J_{t+1} = g_{\mathsf{x}_\bullet(t)\mathsf{x}}$, consider $J_t - J_{t+1}$,

$$J_t - J_{t+1} = g_{\mathsf{x}_\bullet(t)\mathsf{x}_\bullet(t)}\left(\boldsymbol{A}_{\mathsf{x}_\bullet(t)}(t), \boldsymbol{A}_{\mathsf{x}_\bullet(t)}(t)\right)$$
$$- g_{\mathsf{x}_\bullet(t)\mathsf{x}}\left(\boldsymbol{A}_{\mathsf{x}}(t), \boldsymbol{A}_{\mathsf{x}_\bullet(t)}(t+1)\right) \quad (74)$$
$$= \tilde{\boldsymbol{v}}_{\mathsf{x}_\bullet \mathsf{x}_\bullet}^\intercal \boldsymbol{A}_{\mathsf{x}_\bullet}^{(t)} \tilde{\boldsymbol{v}}_{\mathsf{x}_\bullet \mathsf{x}_\bullet} - \tilde{\boldsymbol{v}}_{\mathsf{x}\mathsf{x}_\bullet}^\intercal \boldsymbol{A}_{\mathsf{x}}(t) \tilde{\boldsymbol{v}}_{\mathsf{x}\mathsf{x}_\bullet}$$
$$- \ln|\boldsymbol{A}_{\mathsf{x}_\bullet(t)}(t)| + \ln|\boldsymbol{A}_{\mathsf{x}_\bullet(t)}(t)|$$
$$+ \ln|\boldsymbol{A}_{\mathsf{x}}(t)| - \ln|\boldsymbol{A}_{\mathsf{x}_\bullet(t)}(t+1)| \quad (75)$$

With further simplification and by imposing the condition $J_t - J_{t+1} > 0$, we get:

$$\alpha_{\mathsf{x}_\bullet(t)} \leq b_{\mathsf{x}_\bullet(t),\mathsf{x}} \quad (76)$$

where $b_{\mathsf{x}_\bullet(t)}$ is given in (52).

□

References

[1] C. Dwork, "Differential privacy: A survey of results," in *Theory and Applications of Models of Computation*, M. Agrawal, D. Du, Z. Duan, and A. Li, Eds. Springer Berlin Heidelberg, 2008, pp. 1–19.

[2] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[3] A. D. Sarwate and K. Chaudhuri, "Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data," *IEEE Signal Processing Magazine*, vol. 30, no. 5, pp. 86–94, 2013.

[4] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*. IEEE, 2007, pp. 94–103.

[5] K. Nissim, R. Smorodinsky, and M. Tennenholtz, "Approximately optimal mechanism design via differential privacy," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, 2012, pp. 203–213.

[6] B. Balle and Y.-X. Wang, "Improving the Gaussian mechanism for Differential Privacy: Analytical Calibration and Optimal Denoising," in *International Conference on Machine Learning*. PMLR, 2018, pp. 394–403.

[7] D. Kifer and A. Machanavajjhala, "Pufferfish: A framework for mathematical privacy definitions," *ACM Trans. Database Syst.*, vol. 39, no. 1, Jan. 2014.

[8] X. He, A. Machanavajjhala, and B. Ding, "Blowfish privacy: Tuning privacy-utility trade-offs using policies," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, 2014, pp. 1447–1458.

[9] Y. Xiao, L. Xiong, S. Zhang, and Y. Cao, "Loclok: Location cloaking with differential privacy via hidden markov model," *Proceedings of the VLDB Endowment*, vol. 10, no. 12, pp. 1901–1904, 2017.

[10] Y. Xiao, Y. Shen, J. Liu, L. Xiong, H. Jin, and X. Xu, "DPHMM: Customizable Data Release with Differential Privacy via Hidden Markov Model," *arXiv preprint arXiv:1609.09172*, 2016.

[11] Y. Xiao and L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1298–1309.

[12] S. Song, Y. Wang, and K. Chaudhuri, "Pufferfish privacy mechanisms for correlated data," in *Proceedings of the 2017 ACM International Conference on Management of Data*, 2017, pp. 1291–1306.

[13] F. Fioretto and P. Van Hentenryck, "Optstream: releasing time series privately," *Journal of Artificial Intelligence Research*, vol. 65, pp. 423–456, 2019.

[14] J. Le Ny, *Differential Privacy for Dynamic Data*. Springer, 2020.

[15] O. Sheffet, "Old techniques in differentially private linear regression," in *Algorithmic Learning Theory*. PMLR, 2019, pp. 789–827.

[16] G. Bernstein and D. R. Sheldon, "Differentially Private Bayesian Linear Regression," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[17] A. Honkela, "Gaussian processes with differential privacy," *arXiv preprint arXiv:2106.00474*, 2021.

[18] H. Wang, J. Zhang, C. Lu, and C. Wu, "Privacy preserving in non-intrusive load monitoring: A differential privacy perspective," *IEEE Transactions on Smart Grid*, vol. 12, no. 3, pp. 2529–2543, 2020.

[19] J. Zhao, T. Jung, Y. Wang, and X. Li, "Achieving Differential Privacy of Data Disclosure in the Smart Grid," in *INFOCOM, IEEE Computer and Communications Societies, IEEE Annual Joint Conference*, 2014.

[20] M. B. Gough, S. F. Santos, T. AlSkaif, M. S. Javadi, R. Castro, and J. P. Catalão, "Preserving privacy of smart meter data in a smart grid environment," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 1, pp. 707–718, 2021.

[21] K. Parker, M. Hale, and P. Barooah, "Spectral differential privacy: Application to smart meter data," *IEEE Internet of Things Journal*, vol. 9, no. 7, pp. 4987–4996, 2021.

[22] L. Ou, Z. Qin, S. Liao, T. Li, and D. Zhang, "Singular spectrum analysis for local differential privacy of classifications in the smart grid," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5246–5255, 2020.

[23] F. Fioretto, T. W. Mak, and P. Van Hentenryck, "Differential privacy for power grid obfuscation," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1356–1366, 2019.

[24] V. Dvorkin, P. Van Hentenryck, J. Kazempour, and P. Pinson, "Differentially private distributed optimal power flow," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 2092–2097.

[25] V. Dvorkin, F. Fioretto, P. Van Hentenryck, P. Pinson, and J. Kazempour, "Differentially private optimal power flow for distribution grids," *IEEE Transactions on Power Systems*, vol. 36, no. 3, pp. 2186–2196, 2020.

[26] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber, "Privacy: Theory meets practice on the map," in *2008 IEEE 24th International Conference on Data Engineering*, 2008, pp. 277–286.

[27] D. McClure, "Relaxations of Differential Privacy and Risk/Utility Evaluations of Synthetic Data and Fidelity Measures," Ph.D. dissertation, Duke University, 2015.

[28] S. Meiser, "Approximate and probabilistic differential privacy definitions," *Cryptology ePrint Archive*, 2018.

[29] Y.-X. Wang, S. Fienberg, and A. Smola, "Privacy for Free: Posterior Sampling and Stochastic Gradient Monte Carlo," in *International Conference on Machine Learning*. PMLR, 2015, pp. 2493–2502.

[30] N. Ravi, A. Scaglione, S. Kadam, R. Gentz, S. Peisert, B. Lunghino, E. Levijarvi, and A. Shumavon, "Differentially private k-means clustering applied to meter data analysis and synthesis," *IEEE Transactions on Smart Grid*, 2022.