α-Mutual Information: A Tunable Privacy Measure for Privacy Protection in Data Sharing

MirHamed Jafarzadeh Asl[1], Mohammadhadi Shateri[2], and Fabrice Labeau[1]

[1]Department of Electrical and Computer Engineering, McGill University, QC, Canada,
Email: mirhamed.jafarzadehasl@mail.mcgill.ca, fabrice.labeau@mcgill.ca
[2]Department of Systems Engineering, École de Technologie Supérieure, QC, Canada,
Email: mohammadhadi.shateri@etsmtl.ca

Abstract—This paper adopts Arimoto's α-Mutual Information as a tunable privacy measure, in a privacy-preserving data release setting that aims to prevent disclosing private data to adversaries. By fine-tuning the privacy metric, we demonstrate that our approach yields superior models that effectively thwart attackers across various performance dimensions. We formulate a general distortion-based mechanism that manipulates the original data to offer privacy protection. The distortion metrics are determined according to the data structure of a specific experiment. We confront the problem expressed in the formulation by employing a general adversarial deep learning framework that consists of a releaser and an adversary, trained with opposite goals. This study conducts empirical experiments on images and time-series data to verify the functionality of α-Mutual Information. We evaluate the privacy-utility trade-off of customized models and compare them to mutual information as the baseline measure. Finally, we analyze the consequence of an attacker's access to side information about private data and witness that adapting the privacy measure results in a more refined model than the state-of-the-art in terms of resiliency against side information.

Index Terms—Tunable privacy measure, Arimoto's α-mutual information, adversarial learning, data sharing, privacy-utility trade-off.

I. INTRODUCTION

Despite technological advancements and increased data generation, the need for data sharing has risen dramatically. However, data sharing always carries the risk of security breaches, with unauthorized entities trying to extract private information from shared data. Notably, the privacy problem in data sharing differs from the data security issue. In data release privacy, any authorized receiver of the data is considered an anticipated invader. Therefore, data security methods are unprofitable in data sharing [1]. As data sharing has progressed with advancements in speed, feasibility, etc., addressing various privacy issues has become more challenging than ever before. For instance, many social media applications require individuals to share private data online [2]. Hence, various privacy-protecting techniques for data sharing have been studied for years. Differential Privacy (DP) has received significant attention in this area, especially due to its low computational overhead [3]. Although DP prioritizes data privacy, it may not be ideal for applications where preserving the utility of shared data is crucial, as it does not specifically address other data properties [4].

A. Related work

Considering the mentioned shortcoming of DP, information-theoretical approaches are widely applied in privacy protection, offering improved privacy-utility trade-offs (PUTs) [5]–[7]. Mutual Information (MI) has been popular in information-theoretical privacy measures. In [6], an MI-based method is designed to prevent leakage of private features in representation learning methods on graphs. Besides, efforts are made to extract the most from the patterns in data to determine convenient metrics. One such example is demonstrated in [7], where Directed Information (DI) is selected as the privacy measure. Nonetheless, in order to achieve flexible PUTs, the necessity of discovering a tunable privacy measure has been perceived. An adjustable metric allows for tailoring the privacy definition to specific use cases, enhancing performance, and demonstrating the capacity of information-theoretical strategies.

Configurable measures of information leakage based on Rényi entropy [8] and Arimoto α-mutual information (α-MI) [9] are designed in the literature. Suggesting tunable metrics in [10], authors introduce α-leakage as a measure of information disclosure that quantifies how much an adversary can infer a specific private attribute of the data. The definitions have been extended in [11]. To the best of our knowledge, the closest study to our work is presented in [12], which employs α-loss (equivalent to using Arimoto α-MI as privacy measure) within an adversarial learning framework for data sharing. However, they formulated the problem as a minimax game with constraints, which has been demonstrated to be unstable with regard to loss in deep learning [13]. Moreover, the influence of the α parameter in such a tunable measure and its impact on improving PUT has not been investigated.

Furthermore, one may assess privacy-preserving data-sharing systems regarding their effectiveness in a scenario where a malicious attacker has access to sort of side information (SI) correlated with private data. The authors in [14] analyze this problem. However, we show that customizing the privacy measure can lead to more reliable models than in [14] in terms of PUT. Notably, the robustness of Maximal α-leakage to arbitrary SI is studied in [15]; however, their conclusion is drawn based on the availability of ground truth private attributes. Although this notion is reasonable in the training phase of a framework, it is unrealistic to imagine that private features are known in the testing stage. Moreover, the assumption of having all attributes of the original data as private features might not be practical in many applications.

B. Contributions

In this paper, a tunable privacy measure has been adopted on distortion-based privacy-preserving data release models. The main contributions of this work are as follows:

1) To the best of our knowledge, this is the first time that the impacts of the α parameter are practically investigated in α-MI as a measure of privacy in the privacy-preserving data release.
2) The impact of the tunable privacy measure is illustrated in the presence of SI that is correlated with the sensitive information of shareable data.
3) We suggest a framework that uses a stable strategy to address the optimization problem of privacy-preserving data release as opposed to a minimax formulation [12].
4) Our framework is customized for several datasets with different structures to examine the advantages of using an adaptable privacy measure.

Notation and conventions

A sequence of random variables $(X_1, X_2, \ldots, X_T)$ is shown as X^T. A sample batch from X^T is written as $\{x^{(b)T}\}_{b=1}^{B}$. The probability distribution of X_t is p_{X_t}, and the conditional distribution of X_t given Y_t is shown as $p_{X_t|Y_t}$. The conditional distribution X^T given Y^T would be $p_{X^T|Y^T}$. A Markov chain composed of $X, Y,$ and Z is written as $X \multimap Y \multimap Z$. The expectation of a function f with respect to p_X is denoted as $E[f(X)]$. The Kullback-Leibler (KL) divergence between two distributions p_1 and p_2 is represented as $\text{KL}(p_1 \| p_2)$.

II. PROBLEM FORMULATION AND TRAINING OBJECTIVE

Let variables Y^T denote the users' useful data. This data may be metered power consumption of houses over T time slots, or any non-sequence data ($T = 1$) such as patients' health conditions. Private variables X^T represent the sensitive information that a particular user is unwilling to share in public, e.g., people's identities in the data collected by social media. We also define observed variables W^T as the variables that would normally be released or shared. We assume that W^T is not independent of X^T. The private information X^T may be present, together with the Y^T, in W^T, or X^T is correlated with Y^T and W^T is formed of Y^T. Therefore, for a particular task, sensitive information should be eliminated from valuable data before sharing the data publicly. In this scenario, a privacy-preserving system is of interest. This system contains a releaser that creates a new representation of Y^T, denoted as Z^T, generated by distorting Y^T to follow two objectives simultaneously: the releaser aims to hide private data from any possible attacker interested in inferring them from released data; at the same time, it tries to preserve useful data, as much as possible, based on specific criteria. Therefore, measures are needed to quantify the released data's privacy performance and utility achievement (i.e., preserving useful attributes). Moreover, harmful attackers could have access to some supplementary (side) information, S, that can assist them in attaining higher inference performance. To quantify the distortion between Z^T and Y^T, we define a distortion measure as $\mathcal{D}(Z^T, Y^T) \triangleq \mathbb{E}[d(Z^T, Y^T)]$, where $d : \mathbb{R}^T \times \mathbb{R}^T \to \mathbb{R}$ can be any distortion metric on $\mathbb{R}^T$. Here, Arimoto's α-Mutual Information is proposed for the privacy measure in the releaser as $I_\alpha^A(X; Z) = H_\alpha(X) - H_\alpha^A(X|Z)$ [9], where $H_\alpha(X)$ is the Rényi entropy of order $\alpha \in (0, 1) \cup (1, \infty)$ [8] written as:

$$H_\alpha(X) = \frac{\alpha}{1-\alpha} \log \left(\sum_x p_X^\alpha(x) \right)^{\frac{1}{\alpha}} = \frac{\alpha}{1-\alpha} \log \|p_X\|_\alpha, \quad (1)$$

and $H_\alpha^A(X|Z)$ is Arimoto's conditional α-entropy defined as:

$$H_\alpha^A(X|Z) = \frac{\alpha}{1-\alpha} \log \sum_z p_Z(z) \left(\sum_x p_{X|Z}^\alpha(x|z) \right)^{\frac{1}{\alpha}}$$
$$= \frac{\alpha}{1-\alpha} \log \mathbb{E}_Z \left[\|p_{X|Z}\|_\alpha \right]. \quad (2)$$

Consequently, $H_\alpha^A(X|Z)$ is generalized to $H_\alpha^A(X^T|Z^T)$ as:

$$H_\alpha^A(X^T|Z^T) = \frac{\alpha}{1-\alpha} \log \sum_{z^T} p_{Z^T}(z^T) \left(\sum_{x^T} p_{X^T|Z^T}^\alpha(x^T|z^T) \right)^{\frac{1}{\alpha}}$$
$$= \frac{\alpha}{1-\alpha} \log \mathbb{E}_{Z^T} \left[\|p_{X^T|Z^T}\|_\alpha \right], \quad (3)$$

where $p_{X^T|Z^T} = \prod_{t=1}^{T} p_{X_t|X^{t-1}, Z^T}$. Finally, the problem of finding the optimal releaser is formulated as follows:

$$\inf_{p_{Z^T|W^T}} I_\alpha^A(X^T; Z^T|S) \quad \text{subject to} \quad \mathcal{D}(Z^T, Y^T) \leq \epsilon, \quad (4)$$

where $\epsilon \geq 0$ is a parameter to force the releaser to control the trade-off between privacy and utility. In addition, the SI term is considered in $I_\alpha^A(X^T; Z^T|S) = H_\alpha^A(X^T|S) - H_\alpha^A(X^T|Z^T, S)$ by substituting all $p_{.|Z^T}$ by $p_{.|Z^T, S}$. Given the fact that $H_\alpha^A(X^T|S)$ cannot be changed by the releaser, i.e., it does not depend on $p_{Z^T|W^T}$, we re-formulate (4) as follows:

$$\inf_{p_{Z^T|W^T}} -\frac{1}{T} H_\alpha^A(X^T|Z^T, S), \quad \text{s.t.} \quad \mathcal{D}(Z^T, Y^T) \leq \epsilon, \quad (5)$$

where the term $\frac{1}{T}$ is included for normalization purposes.

Finding the solution for the optimization problem in (5) is not generally tractable. In addition, tackling this problem requires the availability of $p_{X^T|Z^T, S}$. Hence, the privacy-preserving framework approximates $p_{X^T|Z^T, S}$ by using an estimator network, called adversary. The problem of estimating $p_{X^T|Z^T, S}$ by $p_{\hat{X}^T|Z^T, S}$ can be optimally tackled by minimizing the KL divergence between the distributions written as [16]:

$$\inf_{p_{\hat{X}^T|Z^T, S}} \text{KL}\left(p_{X^T|Z^T, S} \| p_{\hat{X}^T|Z^T, S} \right) = \inf_{p_{\hat{X}^T|Z^T, S}} \mathbb{E}\left[\log \frac{p_{X^T|Z^T, S}}{p_{\hat{X}^T|Z^T, S}} \right], \quad (6)$$

where the expectation is with respect to $p_{X^T, Z^T, S}$. Note that solving (6) is equivalent to minimizing the negative log-likelihood $\mathbb{E}\left[-\log p_{\hat{X}^T|Z^T, S}(X^T|Z^T, S) \right]$. Furthermore, we try to simplify (6) by decomposing the probability distribution $p_{\hat{X}^T|Z^T, S}$, leveraged from the natural characteristics of the defined privacy-preserving problem. We denote the releaser and the adversary as $\mathcal{R}_\theta$ and $\mathcal{A}_\phi$, which are controlled by their parameters θ and ϕ, respectively. For $t \in \{1, 2, \ldots, T\}$, the releaser $\mathcal{R}_\theta$ takes observed variables, W^t, as its input and generates released variables represented as Z_t. Using Z^t, the adversary $\mathcal{A}_\phi$ aims to estimate sensitive information x_t by approximating $p_{X_t|Z^t, S}$ at each time t as $p_{\hat{X}_t|Z^t, S}$ and then solving $\hat{x}_t^* = \underset{\hat{x}_t \in \mathcal{X}}{\arg\max} \, p_{\hat{X}_t|Z^t, S}(\hat{x}_t|z^t, s)$. This means, while the goal of $\mathcal{A}_\phi$ is to estimate X^T as precisely as possible based on Z^T, $\mathcal{R}_\theta$ aims to trade-off two different objectives.

Fig. 1. General privacy-preserving framework based on adversarial learning. Parts shown in red color are included as per the application and availability.

Fig. 2. Privacy-preserving framework for image datasets.

On the one hand, $\mathcal{R}_\theta$ intends to minimize the amount of information leaked about X^T from Z^T, which will mislead the adversary. On the other hand, $\mathcal{R}_\theta$ tries to keep Z^T as close as possible to Y^T by limiting the distortion between Z^T and Y^T below a designated value. Based on these assumptions about releaser and adversary, we can conclude that the Markov chains $(X^t, Y^t) \rightarrow W^t \rightarrow Z^t \rightarrow \hat{X}^t$ and $\hat{X}^{t-1} \rightarrow Z^t, S \rightarrow \hat{X}^t$ hold for $t \in \{1, 2, \ldots, T\}$. Therefore, $p_{\hat{X}^T|Z^T,S}$ is re-formulated as:

$$p_{\hat{X}^T|Z^T,S}(\hat{x}^T|z^T,s) = \prod_{t=1}^{T} p_{\hat{X}_t|\hat{X}^{t-1},Z^T,S}(\hat{x}_t|\hat{x}^{t-1}, z^T, s)$$

$$= \prod_{t=1}^{T} p_{\hat{X}_t|Z^T,S}(\hat{x}_t|z^T,s) \overset{\text{(i)}}{=} \prod_{t=1}^{T} p_{\hat{X}_t|Z^t,S}(\hat{x}_t|z^t,s). \quad (7)$$

where (i) corresponds to the causality constraints that the problem may have. Hence, The adversary's objective in (6) can be achieved by addressing the optimization problem written as:

$$\inf_{p_{\hat{X}_t|Z^t,S}} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[-\log p_{\hat{X}_t|Z^t,S}(X_t|Z^t,S)\right], \quad (8)$$

and the optimization problem of the releaser, defined in (5), is converted to a practical formulation as:

$$\inf_{p_{Z^T|W^T}} -\frac{1}{T} H_\alpha^A(\hat{X}^T|Z^T,S), \quad \text{s.t.} \quad \mathcal{D}(Z^T,Y^T) \leq \epsilon, \quad (9)$$

where the distribution on (7) is used to compute $H_\alpha^A(\hat{X}^T|Z^T,S)$. This optimization problem can be tackled with the availability of $p_{\hat{X}^T|Z^T,S}$, the adversary's output.

Based on (8), $\mathcal{A}_\phi$ tries to maximize the quantified information between X^T and Z^T by minimizing KL distance between $p_{\hat{X}_t|Z^t}$ and $p_{X_t|Z^t}$. On the other hand, $\mathcal{R}_\theta$ aims to minimize α-MI in (5). Arimoto's α-MI is known to be a generalization for MI to measure the information shared between random variables [9], [17]. This suggests that the adversary's goals and the releaser's are in opposite directions. Thus, addressing (5) and (8) can be done by a stable adversarial training procedure that uses the general modeling framework illustrated in Fig. 1. Two loss functions $\mathcal{L}_\mathcal{R}(.)$ and $\mathcal{L}_\mathcal{A}(.)$ are determined for $\mathcal{R}_\theta$ and $\mathcal{A}_\phi$, respectively. Using (8), $\mathcal{L}_\mathcal{A}(\phi)$ is written as:

$$\mathcal{L}_\mathcal{A}(\phi) := \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[-\log p_{\hat{X}_t|Z^t,S}(X_t|Z^t,S)\right]. \quad (10)$$

As previously mentioned, (10) represents cross-entropy loss which establishes a classifier that generates $p_{\hat{X}^T|Z^T,S}$. The releaser's loss function is derived from (5) as:

$$\mathcal{L}_\mathcal{R}(\theta,\phi,\omega,\alpha,\lambda) := \mathcal{D}(Z^T,Y^T) - \frac{\lambda}{T} H_\alpha^A(\hat{X}^T|Z^T,S). \quad (11)$$

The presence of S in (10) and (11) depends on the availability of SI. Adjusting $\lambda \geq 0$ in (11) is equivalent to changing ϵ in (5).

Considering the extreme cases, $\lambda = 0$ leads the releaser to the full utility regime, meaning that $\mathcal{R}_\theta$ acts independently from $\mathcal{A}_\phi$, hence provides no privacy guarantees. For large λ values, the term $-\frac{\lambda}{T} H_\alpha^A(\hat{X}^T|Z^T,S)$ will be dominant in $\mathcal{L}_\mathcal{R}(.)$. Thus, the releaser tends to achieve full privacy, i.e., random guessing performance, by confusing the adversary totally. Moreover, ω in (11) shows the parameters that the utility network could have. Depending on the application, this network should have a complex structure or should only evaluate the specified distortion measure. Moreover, this network may generate $\hat{C}$, to which, in some applications, the distortion metric compares specific features of the useful data.

III. FRAMEWORK AND IMPLEMENTATION

A. Privacy-preserving framework for image data

Convolutional neural networks (CNNs) excel in various machine learning tasks, particularly with image datasets. Hence, in this application, we decided to build the networks shown in Fig. 1 by using CNN modules and well-known structures related to each network's task. As illustrated in Fig. 2, Y^T is considered as the releaser's input (i.e., $W^T = Y^T$). An encoder-decoder approach has been employed to design $\mathcal{R}_\theta$, while the adversary and utility network are image classifiers. In this work, we choose a dataset of hand-written digits where the digits' thickness is considered as private information. Thus, $\mathcal{A}_\phi$ tries to determine whether an image shows a thick or a thin digit. On the other hand, $\mathcal{R}_\theta$ aims to generate an image with the same dimensions as Y^T while minimizing the distortion between the generated and original image.

The distortion measure typically quantifies the difference between the network's input and output, either on an element-wise basis or through a higher-level approach. For example, while the thickness of digits is the sensitive information that we try to hide, the ability to classify the digits is of interest. Here, an element-wise measure cannot guarantee digit classification. We consider that the distortion measure consists of two parts: (i) a p-norm metric that quantifies the distortion happened to the input variables, written as $d_p(Z^T,Y^T) \triangleq \frac{1}{T}\|Z^T - Y^T\|_p$ for $p \geq 1$; (ii) the loss function of the utility network, which is a categorical cross-entropy loss for an image classifier that recognizes digits. The first part of the distortion measure ensures that the released image will have element-wise similarity with the input, while the second part promotes the similarity in terms of the results of image classification. In this application, We consider $p = 1$ as the first part of the distortion measure, and the second part comes from the utility network, $\mathcal{C}_\omega$, written as:

$$d_\mathcal{C}(Z^T,Y^T) \triangleq \mathcal{L}_\mathcal{C}(\omega) = \mathbb{E}\left[-\log p_{\hat{C}|Z^T}(C|Z^T,S)\right], \quad (12)$$

where C represents particular utility features (e.g., the labels of hand-written digits images), and $\hat{C}$ is the utility network's output. Finally, the distortion measure is derived as:

$$d_{\text{IMG}}(Z^T, Y^T) = d_{\mathcal{C}}(Z^T, Y^T) + \frac{1}{T}\|Z^T - Y^T\|_1 \quad (13)$$

For the model shown in Fig. 2, $\mathcal{A}_\phi$ has a cross-entropy loss, defined in (10), and the loss function of $\mathcal{R}_\theta$ is formulated as:

$$\mathcal{L}_{\mathcal{R}}(\theta, \phi, \omega, \alpha, \lambda) := \mathbb{E}\left\{d_{\text{IMG}}(Z^T, Y^T)\right\} - \frac{\lambda}{T} H_\alpha^A(\hat{X}^T | Z^T, S). \quad (14)$$

The training process for the data releaser model of this work has multiple stages. At every training iteration, $\mathcal{A}_\phi$ is trained k times, while $\mathcal{R}_\theta$ is only trained once per iteration. The choice of k is crucial as it affects the adversary's strength [13]. Algorithm 1 provides a detailed training procedure. After the training phase, a distinct network, called attacker, is considered for the test phase. This network is trained with the released data and will test the privacy achieved by the model. This network plays the role of a real-world attacker, which has an approximately similar structure to $\mathcal{A}_\phi$ and tries to infer sensitive information from released data.

Algorithm 1 Training of privacy-preserving framework.
Hyperparameters: Batch size B, Adversary training steps k.

1: **for** number of iterations **do**
2:      **for** k steps **do**
3:          Sample $\{y^{(b)T}, x^{(b)T}\}_{b=1}^B$ to create $\{w^{(b)T}\}_{b=1}^B$.
4:          Generate $\{z^{(b)T}\}_{b=1}^B$ by using $\{w^{(b)T}\}_{b=1}^B$ and $\mathcal{R}_\theta$.
5:          Compute gradient of $\mathcal{L}_{\mathcal{A}}(\phi)$, approximated with $\{z^{(b)T}\}_{b=1}^B$, or $\{z^{(b)T}, s^{(b)T}\}_{b=1}^B$ when SI is available.
6:          Update ϕ based on the gradient of $\mathcal{L}_{\mathcal{A}}(\phi)$.
7:          If available, compute gradient of the utility network's loss and update ω based on the gradient.
8:      **end for**
9:      Sample $\{y^{(b)T}, x^{(b)T}\}_{b=1}^B$ to create $\{w^{(b)T}\}_{b=1}^B$.
10:     Compute gradient of $\mathcal{L}_{\mathcal{R}}(\theta)$, approximated with $\{w^{(b)T}\}_{b=1}^B$, and update θ based on the gradient.
11: **end for**

B. Privacy-preserving framework for time-series data

Our second example deals with time-series data. The most important feature of time series is the correlation of data points over time. In order to extract this feature, we use Long Short-Term Memory (LSTM) modules to build releaser and adversary networks of the general model shown in Fig. 1. We form W^T by concatenating Y^T and X^T. This study focuses on time-series applications where utility is defined as the similarity between released data and actual observations, such as smart grid applications [5]. Hence, a p-norm distortion is sufficient to compare the input and output of the releaser. Therefore, we choose $d_{\text{TS}}(Z^T, Y^T) = \frac{1}{T}\|Z^T - Y^T\|_{p=2}$ as the distortion measure in this application, and there is no need to have a complex utility network. Finally, the loss function for $\mathcal{A}_\phi$ is the same as (10), and, for releaser $\mathcal{R}_\theta$, it becomes:

$$\mathcal{L}_{\mathcal{R}}(\theta, \phi, \alpha, \lambda) := \mathbb{E}\left\{d_{\text{TS}}(Z^T, Y^T)\right\} - \frac{\lambda}{T} H_\alpha^A(\hat{X}^T | Z^T, S). \quad (15)$$

The training procedure of this framework is available by adjusting Algorithm 1 based on time-series properties. Similar to section III-A, a distinct attacker evaluates the privacy attained by the model.

IV. RESULTS AND DISCUSSION

A. Datasets description

1) Annotated MNIST (AMNIST) dataset: We use the well-known MNIST dataset [18] and modify it by adding a label of thickness level to images using the method provided in [19]. In [19], authors have defined mathematical formulas with different parameters for each digit. Therefore, the digit thickness in a particular sample image can be classified into thick, normal, or thin. We customized the provided code in [19] to label all training and testing images, and we excluded those images with a normal thickness for computational simplicity. We ended up with 28,568 training and 4,681 testing samples.

2) ECO dataset: The Electricity Consumption and Occupancy (ECO) dataset [20] contains power consumption data of 6 households and their ground truth occupancy information. Since, in this work, the consumption data and occupancy labels are re-sampled at every hour, ECO would be considered a time-series dataset with $T = 24$. Here, the power consumption represents the utility feature Y_t, while the household occupancy is the private information X_t. We partitioned data into 8980 training and 2245 testing time-series sequences. Moreover, week's day and month are possible SI available in ECO that can be concatenated to training and testing samples.

B. Metrics

We choose Normalized Error (NE), i.e., Normalized Mean Squared Error (NMSE), to evaluate the distortion between Y^T and Z^T. We employ balanced accuracy to compare models' performance. This metric is used instead of accuracy to mitigate the unbalanced data effects. Henceforward, we use the word *accuracy* to refer to *balanced accuracy*, for brevity.

C. Tunable privacy measure for AMNIST dataset

The effects of the proposed tunable privacy measure are evaluated by performing an experiment using the modified AMNIST dataset and the proposed framework for image data. We selected $\alpha = 1$ (equivalent to MI) and explored the intervals $(0,1)$ and $(1,\infty)$ to examine the model's performance by varying α. The outcomes revealed that models with $\alpha < 1$ exhibit analogous behavior, with only insignificant differences. The same phenomenon holds for $\alpha > 1$. Thus, the following values are considered for the experiments in this work: $\alpha = 0.9, 1, 3$.

The details of the layers used in the framework are demonstrated in Fig. 2. The hyperparameters in Algorithm 1 are set to $B = 256$ and $k = 3$. Here, full privacy is achieved when the attacker cannot guess better than 50% since we consider that the thickness has two possible values. The attacker's structure is similar to the adversary model described in Fig. 2.

In Fig. 3, the PUT for digits' thickness inference is shown. Note that by using the original images, Y^T, a model can classify the digits and predict their thicknesses with 97.25% and 91.50% accuracy, respectively. As illustrated in Fig. 3,

Fig. 3. Privacy-utility trade-off for digits' thickness inference in models with different privacy measures (tuned by changing α). The fitted curves are exponential functions and are shown only for illustration purposes.

for all models, the classification accuracy is almost preserved where the attacker's accuracy is around 60%. Moreover, the classification accuracy is significantly high around the first point in the full privacy region (FPR). This result ensures achieving the essential utility goal, which is the ability to classify the released digits with high accuracy. The behavior around edge cases is almost the same for all models, except that the model with $\alpha = 0.9$ reaches the FPR with lower classification accuracy than others. The result shows the power of $\alpha = 3$ while transitioning from full utility region (FTR) to the middle of the curve by reducing attacker's accuracy the most, with a slight change in digit classification. However, in the (FTR), $\alpha = 1$ suggest better classification accuracy. Notably, the model with $\alpha = 0.9$ is very sensitive to small changes of λ in (11), which is necessary for generating points of the PUT curve. Due to this sensitivity, finding a point in the middle of the curve requires more effort than other α values.

Fig. 4 shows examples of the released images for selected models. For each sub-figure, we select a point in the middle of the PUT and the first point in the FPR. The results corresponding to middle of the PUT illustrate that by losing a small quantity of digit classification accuracy, the attacker's accuracy is dropped by about 30%. Interestingly, each model's distortion has occurred differently in the full privacy examples. These results indicate no best value of α for all desired operating points on the PUT. Therefore, α gives a degree of freedom to find a model that works best in a desired region.

We design another attacker which has gained access to the algorithm of [19]. We refer to it as the "Thickness-Computing Attacker (TCA)." Using the algorithm, TCA can label digits as thick, normal, or thin. Since we excluded digits with normal thickness from the experiment's data, the attacker has three options for labeling digits for which the algorithm predicts normal thickness: to assign 1) random, 2) thin, or 3) thick labels. We considered all cases for each model and reported their maximum accuracy. Some results of TCA are compared with the deep attacker (DA) in Table I. DA is stronger than TCA around the middle of the PUT; however, TCA achieves

Fig. 4. Samples of the released images of the privacy-preserving framework for the AMNIST dataset with $\alpha = 3$, 1, and 0.9.

TABLE I
THICKNESS INFERENCE RESULTS OF DIFFERENT ATTACKERS

Model Parameters	DA's Accuracy	TCA's Accuracy	Labeling
$\alpha=3$, $\lambda=0.1036$	62.47%	55.82%	Thin
$\alpha=3$, $\lambda=0.131$	50.56%	57.10%	Thin
$\alpha=1$, $\lambda=0.08$	63.60%	62.15%	Thin
$\alpha=1$, $\lambda=0.13$	50.17%	56.65%	Thin
$\alpha=0.9$, $\lambda=0.065544$	61.10%	60.28%	Thin
$\alpha=0.9$, $\lambda=0.065546$	50.00%	50.60%	Thick

better accuracy near the FPR. Interestingly, this large gap happens for models with $\alpha = 3$ and 1 when the attackers decide to convert normal labels to thin. However, for $\alpha = 0.9$, converting to thick labels is the selected approach. Since the gap is negligible in this case, we conclude that the model with $\alpha = 0.9$ is more robust against different attackers than others.

D. Tunable privacy measure for ECO dataset

Moving forward with ECO dataset, $\alpha = 3, 1, 0.9$ are selected based on the discussed reason in section IV-C. The general framework illustrated in Fig. 1 is customized based on section III-B. In addition, an independent uniformly distributed (over $[0, 1]$) noise U^T is integrated into W^T beside Y^T and X^T to randomize Z^T. It is seen to be helpful in practical applications where an adversarial framework's input consists of noise [21]. The releaser network consists of 4 LSTM layers, each with 64 cells, and the adversary network is formed of 3 LSTM layers, each with 32 cells. The distinct attacker has the same structure as the adversary. The hyperparameters indicated in Algorithm 1 are set to $B = 128$, $k = 4$. As discussed in section IV-A2, household occupancy is private information. Thus, the corresponding attack accuracy of the FPR is 50%. Notably, an attacker can predict household occupancy from the actual power consumption with more than 90% accuracy.

The PUT for house occupancy inference is available in Fig. 5a. In [5], a similar experiment is investigated where MI is the privacy measure. Here, around FTR and FPR, all models

Fig. 5. Privacy-utility trade-off for house occupancy inference in models with different privacy measures. (a) without SI, (b) SI is available to the attacker.

Fig. 6. Samples of the released power consumption modified by privacy-preserving framework for time-series datasets with $\alpha = 3, 1$, and 0.9.

accomplish almost the same trade-off. However, the model with $\alpha = 1$ performs best in the middle of the PUT. Fig. 6 shows 7-day-long samples from modified power consumption signals. In this figure, two operating points are selected for each α. The models corresponding to the left side of Fig. 6 preserve most of the original data (NE is less than 0.26 in the worst case), while the attacker's accuracy is dropped by more than 26%. In addition, different distortion patterns can be realized on the right side of the figure for different α values.

Another experiment is designed with ECO for a situation where the SI discussed in section IV-A2 is available to an attacker. Fig. 5b shows the PUT for selected α values. Similar work is conducted in [14], where MI is the privacy measure. In [14], an attacker trained and tested with only SI achieves an accuracy of 57.8%, concluding that the attacker is not completely confused even by signals with large distortion. In Fig. 5b, the model with $\alpha = 1$ attains the attacker's accuracy of 57.8% on large NE, while surprisingly, the model with $\alpha = 0.9$ maintains the accuracy of 55.2%. In addition, The baseline for the model with $\alpha = 3$ is 56.8%. These results suggest better performance than [14] in preserving sensitive information of a highly distorted signal when SI is available to the attacker.

V. CONCLUSION

This research proposes a general privacy-preserving data-sharing model that allows for tunable privacy measures, particularly leveraging α-Mutual Information. A key finding of the research is the influential role of the α parameter, which can be adjusted to balance privacy and utility in various scenarios. Experimental tests, using an image dataset of handwritten digits and a time-series sequence of power consumption measurements, revealed that tuning α allows for tailored data-sharing frameworks, with signals released per specific features of interest. The research also considered scenarios where attackers have access to correlated SI. The results indicated that fine-tuning of the privacy measure should consider not just

the PUT, but also the model's resilience against SI. Lastly, in addition to the generic attacker of the framework, an arithmetic attacker was considered in relation to the AMNIST dataset used in this work. The results highlighted that certain models (with different privacy metrics) may be more or less successful at concealing sensitive information, depending on whether the attacker knows private information's pattern in the actual data.

REFERENCES

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

[2] S. Ray, T. Palanivel, N. Herman, and Y. Li, "Dynamics in data privacy and sharing economics," *IEEE Transactions on Technology and Society*, vol. 2, no. 3, pp. 114–115, 2021.

[3] Y. Zhao and J. Chen, "A survey on differential privacy for unstructured data content," *ACM Comput. Surv.*, vol. 54, sep 2022.

[4] R. Mendes and J. P. Vilela, "Privacy-preserving data mining: Methods, metrics, and applications," *IEEE Access*, vol. 5, pp. 10562–10582, 2017.

[5] M. Shateri, F. Messina, P. Piantanida, and F. Labeau, "Real-time privacy-preserving data release for smart meters," *IEEE Transactions on Smart Grid*, vol. 11, no. 6, pp. 5174–5183, 2020.

[6] B. Wang, J. Guo, A. Li, Y. Chen, and H. Li, "Privacy-preserving representation learning on graphs: A mutual information perspective," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, p. 1667–1676, 2021.

[7] M. Shateri, F. Messina, P. Piantanida, and F. Labeau, "Deep directed information-based learning for privacy-preserving smart meter data release," in *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids*, pp. 1–7, 2019.

[8] A. Rényi, "On measures of entropy and information," in *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, vol. 4, pp. 547–562, University of California Press, 1961.

[9] S. Arimoto, "Information measures and capacity of order α for discrete memoryless channels," *Topics in Information Theory*, 1977.

[10] J. Liao, O. Kosut, L. Sankar, and F. P. Calmon, "A tunable measure for information leakage," in *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 701–705, 2018.

[11] A. Gilani, G. R. Kurri, O. Kosut, and L. Sankar, "(α, β)-leakage: A unified privacy leakage measure," 2023.

[12] P. Kairouz, J. Liao, C. Huang, M. Vyas, M. Welfert, and L. Sankar, "Generating fair universal representations using adversarial models," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1970–1985, 2022.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds.), vol. 27, Curran Associates, Inc., 2014.

[14] M. Shateri, F. Messina, P. Piantanida, and F. Labeau, "On the impact of side information on smart meter privacy-preserving methods," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids*, pp. 1–6, 2020.

[15] J. Liao, L. Sankar, O. Kosut, and F. P. Calmon, "Robustness of maximal α-leakage to side information," in *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 642–646, 2019.

[16] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. 2006.

[17] S. Verdú, "α-mutual information," in *2015 Information Theory and Applications Workshop (ITA)*, pp. 1–6, 2015.

[18] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[19] B. Kim, "Annotated MNIST: Thickness and skew labeler for MNIST handwritten digit dataset." GitHub, Aug. 1, 2017 [Online]. Available: https://github.com/1202kbs/Annotated_MNIST.

[20] W. Kleiminger, C. Beckel, and S. Santini, "Household occupancy monitoring using electricity meters," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2015.

[21] A. Tripathy, Y. Wang, and P. Ishwar, "Privacy-preserving adversarial networks," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 495–505, 2019.