

## Learning to Identify while Failing to Discriminate

Jure Sokolić  
University College London  
jure.sokolic.13@ucl.ac.uk

Qiang Qiu  
Duke University  
qiang.qiu@duke.edu

Miguel R. D. Rodrigues  
University College London  
m.rodrigues@ucl.ac.uk

Guillermo Sapiro  
Duke University  
guillermo.sapiro@duke.edu

### Abstract

*Privacy and fairness are critical in computer vision applications, in particular when dealing with human identification. Achieving a universally secure, private, and fair systems is practically impossible as the exploitation of additional data can reveal private information in the original one. Faced with this challenge, we propose a new line of research, where the privacy is learned and used in a closed environment. The goal is to ensure that a given entity, trusted to infer certain information with our data, is blocked from inferring protected information from it. We design a system that learns to succeed on the positive task while simultaneously fail at the negative one, and illustrate this with challenging cases where the positive task (face verification) is harder than the negative one (gender classification). The framework opens the door to privacy and fairness in very important closed scenarios, ranging from private data accumulation companies to law-enforcement and hospitals.*

### 1. Introduction

Advances in machine learning and data science allow to develop products and services that leverage individuals data to provide valuable services. At the same time care must be taken to provide an appropriate amount of protection for users in order to adhere to various legal and ethical constraints.

In this work we envision the following scenario: An entity (e.g., the company or the government) wants to offer a service to its users based on their data, (e.g., an ATM verify the card holder is the actual owner). However, some users want to be able to prevent the entity from inferring certain sensitive information from their data (e.g., the gender of the person withdrawing money from the ATM). The need for this may come from security, fairness, or privacy concerns. For example, a hospital might be allowed to pro-

duce diagnosis on the patient, without being able to infer some other irrelevant condition of the subject. We may allow identity verification for security purposes, but we want to be sure that we can not be discriminated based on gender, race, or age, guaranteeing that such information can not even be inferred from the provided data. This motivates the main question that we try to answer in this work:

*How can we design a (human/face identification/verification) system that can provide a valuable service to a user and offer protection to their sensitive data at the same time?*

The benefits of such a system are twofold. First, the users will be more comfortable because their privacy is respected for the tasks the user wants to block. Second, the system will offer the entity offering the service a principled approach to prove that they are not using/infering users' sensitive information in case of legal/ethical disputes.

Before describing the proposed framework we need to further address two issues that motivate our proposed system. First, can we design a system that is universal, meaning capable to prevent the undesired task by any other system? And second, is the problem that we are trying to solve trivial? The negative answers to these questions are important motivations to the framework here introduced.

Regarding the first question, there are many examples in practice where additional sources of information lead to attacks on the systems that were designed to be private. For example, the authors in [18] showed how to break the anonymity of the publicly released Netflix Prize Dataset. They achieved that by using other public datasets and an appropriate algorithm. It has been shown in [20] that a person in an image can be identified based on the social network graph and the photos of other users even when the faces in the images are obfuscated. The source and power of this additional data may be very hard to anticipate. Another example is the unexpected possibility to infer cardio-vascular

health, a task we might want to protect, from regular credit scores as those commonly provided to get mortgage authorization [14].

The impossibility of universal privacy protection just exemplified has also been studied extensively in the domain of differential privacy [4], where a number of authors have shown that assumptions about the data or the adversary must be made in order to be able to provide utility [6, 15, 16]. Similarly, in the fairness domain the authors in [12] propose a method to remove discrimination from any learned predictor and note that their fairness objective is not universal, but rather domain specific. As a consequence their framework does not aim to certify fairness but rather provide a set of tools which can be used to improve the fairness of a system.

On the other hand, one may argue that users can prevent inference of sensitive variables by simply sharing less data. For example, [2] shows that if the feature vector representing a face image is sufficiently low dimensional, one can infer the gender of the person, but not his/her identity. However, one may also be interested in face verification, which is not possible if the features do not provide sufficient information, and be agnostic to gender at the same time. This task is much harder, meaning the desired task (verification) is harder than the protected task (gender detection), and the appropriate assumptions must be made, as we will be shown in the sequel.

In view of the issues presented above we propose a novel framework that rather than being universal, assumes a closed environment where privacy with respect to particular sensitive variables is desired. The goal is to ensure that a given entity, trusted to infer certain information with our data, is blocked from inferring sensitive information from it. For example, a company might be allowed to use face verification (the positive task), without being able to infer the irrelevant gender of the subject (negative task). Similarly, a company can guarantee they internally are not using the provided data for any undesired task, an important goal that is not contradicting the virtually impossible challenge of blocking everybody from the undesired task.

To achieve this, we assume an environment where the entity declares a set of (training) data and a set of tools (machine learning algorithms) that will be used to provide a certain service (positive task) to users. We then design a data sanitization function whereby users provide data to the entity that prevents inference of sensitive variables (negative tasks) within the constrained environment. This general idea is visualized in Figure 1.

The proposed framework is describe in detail in Section 2. A critical application in face verification is presented in Section 3. Related literature is reviewed and discussed in Section 4. Finally, the paper is concluded in Section 5.

## 2. Proposed Framework

Our framework consists of a *closed environment* with a set of data  $\mathcal{Z}$  and a set of algorithms  $\mathcal{A}$ . In particular, we assume that the training set consist of  $m$  training examples:  $\mathcal{Z} = \{z_i\}_{i=1}^m$ . An element of  $\mathcal{Z}$  is a triplet  $z = (x, t, s)$ , where  $x \in \mathcal{X}$  represents a feature vector in some feature space  $\mathcal{X}$ ,  $t$  is a label associated with utility (positive task) and  $s$  is a label associated with sensitive data (negative task). For the sake of simplicity we assume that  $t$  and  $s$  are binary, i.e.,  $t, s \in \{-1, 1\}$ . The set of algorithms  $\mathcal{A}$  consists of algorithms  $A$  that map the training set to a predictor

$$A : \mathcal{Z} \mapsto q_{A(\mathcal{Z})} \quad A \in \mathcal{A}, \quad (1)$$

where the predictor is of the form

$$q_{A(\mathcal{Z})} : \mathcal{X} \mapsto \{-1, 1\}. \quad (2)$$

The training data and the algorithms are used to provide a service to the users, where the users' data consists of feature vectors  $x$  and it is denoted by  $\mathcal{U} = \{(x)_i\}_{i=1}^n$ . In particular, the entity provides a service to the user via the prediction of the utility variable  $t$ . However, since the feature vector may also be used to predict the sensitive variable  $s$ , care must be taken to provide protection against that. As discussed before, on one hand, a user might not use the service because of the concerns of revealing her sensitive information. On the other hand, the entity wants to be able to prove that they are not misusing users data by predicting the sensitive variable  $s$  against the user's preference, or for example due to fairness. We assume that the training set has no privacy restrictions.

As discussed in the Introduction, giving guarantees about non-identifiability of certain variables in the "open world" where new data or new algorithms are readily available is very likely impossible. Therefore, what we want to do is to design mechanisms that will prevent prediction of sensitive variables within this important closed environment. Such methods are very relevant in practice: i) they create an additional level of trust between the users and the entity; ii) it gives the entity an elegant way to certify that sensitive information is not being abused within their closed environment; and iii) can provide additional value such as fairness.<sup>1</sup>

We require the entity to define a utility metric, a privacy metric, and to design a data sanitization function that will achieve privacy while providing utility. Of particular challenge is the case when inferring  $s$  (private data) is actually easier than inferring the actual authorized information  $t$ . This is here addressed for the first time.

<sup>1</sup>Such a closed environment can be achieved for example by specifying what algorithm has access to what data in a manner similar to the implementation of human access control in digital environments.

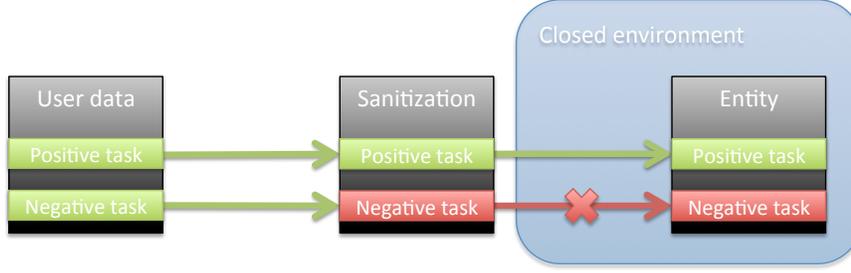


Figure 1. The entity provides a service to its users in a closed environment with a set of data and a set of algorithms. Given a sensitive variable that may be inferred from raw user data, the entity designs a *data sanitization function*, which transforms users' data in such a way that the sensitive variables may not be inferred within the closed environment. The data sanitization function is then either shared with the user who can now share *sanitized data* or it is applied to the data before the users' data enters the entity.

### Utility Metric, Privacy Metric and Data Sanitization.

First, we will assume that the entity has a test set  $\mathcal{Z}'$  with  $m'$  samples where the samples are drawn from the same distribution as the training examples. We define both the utility metric and the privacy metric based on the testing set. This is needed since a representative test set is exploited to gain confidence about the utility of the machine learning model. The same test can be used to verify that given the training set  $\mathcal{Z}$  and the set of algorithms  $\mathcal{A}$ , no meaningful conclusion about the sensitive variable  $s$  can be drawn from the data.

The entity will then design a data sanitization function

$$f : \mathcal{X} \mapsto \mathcal{X}, \quad (3)$$

such that both utility and privacy will be satisfied. Note that  $f$  maps to the same feature space  $\mathcal{X}$  so that sanitized data  $f(x)$  may be used as the input to a predictor  $q_{A(\mathcal{Z})}$ . We will evaluate the data sanitization function on the sanitized test set which will be denoted as  $\mathcal{Z}'_f$ :

$$\mathcal{Z}'_f = \{(f(x), s, t) : (x, s, t) \in \mathcal{Z}'\}. \quad (4)$$

In particular, the utility metric of the closed system will be denoted by

$$u(\mathcal{A}, \mathcal{Z}, \mathcal{Z}'_f) \in [0, 1], \quad (5)$$

and the privacy metric will be denoted as

$$p(\mathcal{A}, \mathcal{Z}, \mathcal{Z}'_f) \in [0, 1]. \quad (6)$$

The goal is to design such  $f$  that both utility and privacy metrics are as close to 1 as possible.

For example, denote by  $\text{CA}_t(\mathcal{A}, \mathcal{Z}, \mathcal{Z}'_f)$  the classification accuracy of the utility task of the predictor  $q_{A(\mathcal{Z})}$  trained on the training set  $\mathcal{Z}$  and evaluated on the sanitized test set  $\mathcal{Z}'_f$ :

$$\text{CA}_t(\mathcal{A}, \mathcal{Z}, \mathcal{Z}'_f) = \sum_{(f(x), s, t) \in \mathcal{Z}'_f} \frac{(1 - \ell(q_{A(\mathcal{Z})}(f(x)), t))}{|\mathcal{Z}'_f|}, \quad (7)$$

where  $\ell$  is a 0-1 loss. Then we might define the utility metric as the classification accuracy of the best possible predictor given the training set  $\mathcal{Z}$  and the set of algorithms  $\mathcal{A}$ :

$$u(\mathcal{A}, \mathcal{Z}, \mathcal{Z}'_f) = \max_{A \in \mathcal{A}} \text{CA}(A, \mathcal{Z}, \mathcal{Z}'_f). \quad (8)$$

Similarly, denote by  $\text{CA}_s(\mathcal{A}, \mathcal{Z}, \mathcal{Z}'_f)$  the classification accuracy of the sensitive task of the predictor  $q_{A(\mathcal{Z})}$  trained on the training set  $\mathcal{Z}$  and evaluated on the sanitized test set  $\mathcal{Z}'_f$ :

$$\text{CA}_s(\mathcal{A}, \mathcal{Z}, \mathcal{Z}'_f) = \sum_{(f(x), s, t) \in \mathcal{Z}'_f} \frac{(1 - \ell(q_{A(\mathcal{Z})}(f(x)), s))}{|\mathcal{Z}'_f|}. \quad (9)$$

Assume also that the prior probabilities of the sensitive labels are balanced,  $p(s = 1) = p(s = 0) = 0.5$ . Then the privacy metric can be defined as the deviation of the the most accurate predictor given the training set  $\mathcal{Z}$  and the set of algorithms  $\mathcal{A}$  from the "random" classifier:

$$p(\mathcal{A}, \mathcal{Z}, \mathcal{Z}'_f) = \min_{A \in \mathcal{A}} 1 - 2 \cdot |\text{CA}_s(A, \mathcal{Z}, \mathcal{Z}'_f) - 0.5|. \quad (10)$$

Note that the definitions of privacy and the utility are application specific. For example, we may define the utility or privacy metrics via averages over different users. However, we could use more restrictive definitions where utility or privacy are defined per user. The novel framework here introduced is independent of the particular selection. A concrete example is presented in Section 3.

The entity might now collect users data via the data sanitization function  $f$  into the environment with training data  $\mathcal{Z}$  and with the set of algorithms  $\mathcal{A}$ . The closed environment is certified with the utility  $u(\mathcal{A}, \mathcal{Z}, \mathcal{Z}'_f)$  and the privacy  $p(\mathcal{A}, \mathcal{Z}, \mathcal{Z}'_f)$ , which may be communicated to the user or a possible regulatory entity.

We have presented a framework to protect users privacy based explicitly on the predefined data and algorithms. The guarantees do not extend to algorithms outside  $\mathcal{A}$  and additional data (side information) outside  $\mathcal{Z}$ . Although these

properties would be desirable, as previously described a large body of literature suggests that universal privacy protection is only possible at the expense of utility [15].

### 3. Face verification and gender recognition from face images

We now demonstrate the proposed framework with an application involving face verification<sup>2</sup> and gender classification. In particular, we will consider the face verification as the utility task and the gender classification as the sensitive task. This may be useful in various applications where we want to re-identify users, but want to prevent gender discrimination/information within the system. This of course goes beyond fairness, where the goal is only to make sure that the system’s performance is independent of gender, but not to block the gender detection.

Note that for the opposite case, where gender classification is the utility task and face verification is the sensitive task is also relevant. However, in this case the undesired task can be easily blocked by simply using a sufficiently low dimensional projection, as shown in [2], and is therefore less challenging.

#### 3.1. Dataset and Tools

##### 3.1.1 Dataset

We use the FaceScrub dataset [19], containing approximately 100,000 face images of 530 individuals, with 265 males and 265 females. For each image,  $x$  is obtained by using the features of the last layer of the VGG-face network [21] projected to dimension 1000 using principal component analysis. Associated with each feature is the identity label  $t$  and the gender label  $s$ . Note that our positive task is verification and not identification, therefore, given two examples  $(x, t, s)$  and  $(x', t', s')$  we want to establish either  $t = t'$  or  $t \neq t'$  based on the features  $x$  and  $x'$ . The negative task is gender classification where the features  $x$  are used to predict the gender  $s$ .

We create three subsets from the original dataset:

- Training set  $\mathcal{Z} = \{(x, t, s)_i\}_{i=1}^m$ , which represents data in the closed environment and is used to design the data sanitization function;
- Test set  $\mathcal{Z}' = \{(x, t, s)_i\}_{i=1}^{m'}$ , which is used to evaluate the utility and privacy metrics. The sanitized version of the test set is denoted as  $\mathcal{Z}'_f = \{(f(x), t, s)_i\}_{i=1}^{m'}$ ;
- User set  $\mathcal{U} = \{(x)_i\}_{i=1}^n$ , which represents users’ data.

The three sets  $\mathcal{Z}$ ,  $\mathcal{Z}'$  and  $\mathcal{U}$  have a balanced ratio of males and females and do not share identities.

<sup>2</sup>Note that face verification verifies if two images belong to the same person (consider for example the ID verification at the airport or the entrance to a facility). It does not identify a person and therefore does not reveal gender.

##### 3.1.2 Metrics

We use the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve to measure success of the face verification task and the gender classification accuracy (CA) to measure success of the gender classification task. Therefore, the utility and the privacy metrics for this particular application are defined as

$$u(\mathcal{A}, \mathcal{Z}, \mathcal{Z}'_f) = \text{AUC}(\mathcal{Z}'_f) \quad (11)$$

and

$$p(\mathcal{A}, \mathcal{Z}, \mathcal{Z}'_f) = \min_{A \in \mathcal{A}} 1 - 2 \cdot |\text{CA}_s(A, \mathcal{Z}, \mathcal{Z}'_f) - 0.5|, \quad (12)$$

respectively.

##### 3.1.3 Algorithms

We test using 3 algorithms, helping us to present the proposed framework and not just an instance of it:

- Verification algorithm (VA): it takes two feature vectors and it outputs the cosine similarity between the two vectors;
- Support Vector Machine (SVM) classifier;
- Random Forest (RF) classifier.

**Details of SVM and RF training.** The training procedure of the classifier works as follows: Provided data is randomly split into training and validation sets. The predictor is trained on the training set for various hyper-parameter settings and the performance on the validation set is recorded. The hyper-parameters corresponding to the best performance on the validation set are chosen and the predictor is trained on the entire dataset.

We use the linear SVM and RF implemented in [22]. The constant  $C$  associated with the linear SVM is picked from the set  $\{1e - 3, 1e - 1, 1, 100\}$ . For the RF classifiers we consider number of trees in  $\{25, 50, 100\}$  and tree depth is set to 3, 5 or it is set automatically by the algorithm.

##### 3.1.4 Data Sanitization

We consider two data sanitization techniques:

- Data Sanitization with Linear Projection:

$$f(x) = Px, \quad (13)$$

where  $P$  is a linear projection matrix.

- **Data Sanitization with Maximum Mean Discrepancy Transformation:**

$$f(x) = x + n(x), \quad (14)$$

where  $n(x)$  is a “noise” vector that depends non-linearly on  $x$ .

The data sanitization function are designed so that they maximize the privacy and minimally affect the utility.

**Data Sanitization with Linear Projection.** The sanitization function with linear projection  $P$  is defined as

$$f(x) = Px. \quad (15)$$

The linear projection  $P$  is obtained by iterating the following steps, where  $P$  is set to identity initially:

1. The features  $x$  in the training set  $\mathcal{Z}$  and the test set  $\mathcal{Z}'$  are transformed by  $P$ :  $x = Px$ .
2. An SVM classifier is trained on the training set  $\mathcal{Z}$  to predict the sensitive variable  $s$ . SVM training details are provided in the paragraph above.
3. A linear projection  $P'$  is constructed so that the kernel of the projection corresponds to the weight vector of the trained SVM. In this way the projection “collapses” the dimension of the space where the training samples are linearly separable.
4. Projection  $P$  is updated to include the kernel of the projection  $P'$ :  $P = P'P$ .

The process is repeated until we achieve a sufficiently low classification accuracy on the test set (we used 55% in our case).

The trade-off between utility and privacy is achieved by increasing or decreasing the dimension of the kernel of the projection  $P$ .

**Data Sanitization with Maximum Mean Discrepancy Transformation.** We extend the technique proposed in [8], which uses the Maximum Mean Discrepancy (MMD) statistics [10]. It is originally used for image style transfer using convolutional neural networks.

The sanitization function is defined as

$$f(x) = x + n(x), \quad (16)$$

where  $n(x)$  is obtained as follows: First a dictionary  $D$  is constructed from the feature vectors  $x$  in the training set  $\mathcal{Z}$ . We set  $n(x)$  to be a linear combination of the features in the training set:  $n(x) = D\theta(x)$  where  $\theta(x)$  is the vector of coefficients. The central part of the method is the computation of the parameter vector  $\theta(x)$ , which we describe next.

We use the features  $x$  from the training set  $\mathcal{Z}$  to construct two subsets:  $\mathcal{X}_+ = \{x : (x, s, t) \in \mathcal{Z}, s = 1\}$  and  $\mathcal{X}_- = \{x : (x, s, t) \in \mathcal{Z}, s = -1\}$ , which correspond to the two classes associated with the sensitive label. Then for each user’s feature vector  $x$  we determine randomly the target class  $s' \in \{-1, 1\}$  and define the cost function

$$c(\theta(x)) = s' \times \left( \frac{1}{|\mathcal{X}_+|} \sum_{x_+ \in \mathcal{X}_+} k(x_+, x + D\theta(x)) - \frac{1}{|\mathcal{X}_-|} \sum_{x_- \in \mathcal{X}_-} k(x_-, x + D\theta(x)) \right), \quad (17)$$

where  $k(x, x')$  is the kernel function. We choose  $k(x, x')$  to be a Gaussian kernel,  $k(x, x') = \exp(-\|x - x'\|_2^2 / 2\sigma^2)$  with parameter  $\sigma = 0.001$  in our experiments.

The value of  $\theta(x)$  is obtained by maximizing the cost function  $c(\theta(x))$ :

$$\max_{\theta(x)} c(\theta(x)). \quad (18)$$

The optimization problem is solved by gradient descent using step size 0.05 and 10,000 iterations.

Intuitively, a large positive value of

$$\frac{1}{|\mathcal{X}_+|} \sum_{x_+ \in \mathcal{X}_+} k(x_+, x + D\theta(x)) - \frac{1}{|\mathcal{X}_-|} \sum_{x_- \in \mathcal{X}_-} k(x_-, x + D\theta(x)) \quad (19)$$

indicates that  $x + D(\theta)$  resembles the data in  $\mathcal{X}_+$ , and a large negative value of (19) indicates that  $x + D\theta(x)$  resembles the data in  $\mathcal{X}_-$ . Therefore, the optimization problem in (18), where  $s' \in \{-1, 1\}$ , can be interpreted as a non-linear noise model that changes feature  $x$  to be close to either  $|\mathcal{X}_+|$  or  $|\mathcal{X}_-|$ .

### 3.2. Results

We now explore various scenarios where we prevent gender classification without compromising verification within a closed environment. In all the cases we will assume that the training set is part of the closed environment. We will then consider the different examples of data sanitization functions and the different algorithms allowed in the closed environment.

**Baseline.** First, we consider a baseline scenario, where the closed environment uses the training set  $\mathcal{Z}$  and all three algorithms:  $\mathcal{A} = \{\text{VA}, \text{SVM}, \text{RF}\}$  and does not do any data sanitization. We report the values of the utility metric and the privacy metric as measured on the test set  $\mathcal{Z}'$ . We also

report the gender classification accuracy of the SVM classifier, the gender classification accuracy of the RF classifier and the verification AUC on the user data. The results are reported in Table 1.

Note that the utility metric is close to 1 and the privacy metric is close to zero. This is expected as no data sanitization is used. The gender classification accuracies on the data in the user set are high, which implies low privacy, and the verification AUC on the user data is also high, which implies high utility.

**Case 1: Linear sanitization and SVMs.** Here we test the closed environment to include the training set  $\mathcal{Z}$ , algorithms:  $\mathcal{A} = \{\text{VA}, \text{SVM}\}$ , and the linear sanitization function for the users’ data. The values of the utility metric and the privacy metric as measured on the test set  $\mathcal{Z}'_f$  are reported in Table 2. The gender classification accuracy of the SVM classifier and the verification AUC on the user data are reported as well. Note that the gender classification accuracy of the RF classifier is not reported because RF classifier is not in the closed environment.

The utility metric in this case is very close to the baseline. Therefore, the utility is preserved by the data sanitization function. The privacy metric, on the other hand, is much higher than in the baseline example, which means that gender recognition does not perform well on the users’ data. These observations are also supported by a low gender classification accuracy of the SVM classifier and the verification AUC on the user dataset, which dropped for only 0.02% compared to the baseline. Therefore, we have simultaneously achieved privacy (and gender fairness as a consequence) and utility within the specified environment.

**Case 2: MMD sanitization, SVMs and RFs.** Now we extend the closed environment from case 1 to include the stronger RF classifiers:  $\mathcal{A} = \{\text{VA}, \text{SVM}, \text{RF}\}$ . The linear sanitization function is not effective when used with the RF classifiers, therefore we use the MMD data sanitization. The results following the format of the baseline and case 1 are presented in Table 3.

Note that the utility metric and the privacy metric are slightly lower than in the case 1. Nevertheless, the verification AUC on the user dataset dropped by 1.03% compared to the baseline, whereas the SVM and RF gender classifiers trained in the closed environment achieved accuracies 50.27% and 53.84%, respectively. We can claim that the data sanitization is therefore effective.

### 3.3. Necessity of the Closed Environment

Finally, we show experimentally that the closed environment assumptions is necessary in the presented scenario.

In particular, we create an additional adversary set  $\mathcal{Z}_f^A = \{(f(x), t, s)_i\}_{i=1}^m$ , which is sanitized by one of the sanitiza-

tion functions and does not share identities with the training set, test set or the user set. We then violate the closed environment assumption and train various classifier on the gender classification task using the adversary set as the training set. The gender classification accuracy results on the user set are reported in Table 4 for different scenarios (baseline, linear sanitization and MMD sanitization).

Note that in the case of linear sanitization and in the case of MMD sanitization the gender classification accuracy is significantly higher than 50% for both the SVM and the RF classifier. The sensitive information can therefore be exposed by leveraging outside data (side information). The reason for this is that data sanitization “removes” information relevant for gender classification that is present in the training set. The adversary set contains information about gender that is complimentary to the information in the training set and is not captured by the sanitization function. From this perspective a closed environment is very well motivated and our work addresses this for the first time.

## 4. Related Work

While the concept of closed environment privacy as here defined is new, and the formulation presented in the previous section should be considered as one possible realization of it, it is related to other contributions in the literature. We give a review of some of these works next.

**Privacy.** The privacy problem has been widely studied in the data mining and database communities, where differential privacy [4] is one of the most popular approaches.

Differentially private data release mechanisms ensure that an adversary may not establish a presence or absence of an individual in a database [24]. An important property, emphasized in the differential privacy literature, is the fact that there is no need for assumptions about the side information of an adversary. However, such universality comes at an expense of utility, as shown in [6, 15, 16]. Our proposed framework differs from the differential privacy framework by assuming a closed environment where the data and tools are predefined/pre-trained. A consequence is that we can maintain utility, while still offering privacy within the closed environment.

Differential privacy has also been applied to training of machine learning models where the goal is to keep the training data private while learning the parameters of the model, e.g., [23, 1]. A successful attack using generative adversarial networks on differentially private training mechanisms has been proposed in [13]. Our framework differs from these models in the sense that we propose data sanitization mechanisms and privacy for user’s data and not for training data.

The Pufferfish framework [16] defines privacy with respect to a set of *potential secrets* and with respect to a set of

Table 1. Baseline. Setup:  $f(x) = x$ ,  $\mathcal{Z}, \mathcal{A} = \{\text{VA, SVM, RF}\}$ .

Utility metric	Privacy metric	User data		
		Gender acc. (SVM)	Gender acc. (RF)	Verification AUC
0.9607	0.0672	96.73 %	95.37 %	96.10 %

Table 2. Case 1. Setup:  $f(x) = Px$ ,  $\mathcal{Z}, \mathcal{A} = \{\text{VA, SVM}\}$ .

Utility metric	Privacy metric	User data		
		Gender acc. (SVM)	Gender acc. (RF)	Verification AUC
0.9600	0.8892	45.81 %	/	96.08 %

Table 3. Case 2. Setup:  $f(x) = x + n(x)$ ,  $\mathcal{Z}, \mathcal{A} = \{\text{VA, SVM, RF}\}$ .

Utility metric	Privacy metric	User data		
		Gender acc. (SVM)	Gender acc. (RF)	Verification AUC
0.9493	0.8713	50.27 %	53.84 %	95.07 %

Table 4. Gender classification accuracy on the user set using classifiers trained on the adversary dataset.

	SVM	RF
Baseline	94.13 %	94.90 %
Linear sanitization	81.52 %	86.99 %
MMD sanitization	78.24 %	79.90 %

assumptions about the knowledge of a potential adversary. Similarly to our framework, this framework constrains the problem to a particular secret and assumes an adversary has a limited prior knowledge. In our framework we explicitly constraint a potential adversary by defining the training data and the set of algorithms. Moreover, we also propose two techniques for data sanitization that are based on data and achieve privacy and utility within a closed environment.

A deep learning architecture for privacy-preserving mobile analytics has been proposed in [2]. Their main tools is dimensionality reduction of shared features, which allows them to prevent the success of hard tasks (person identification), where a higher feature dimensionality is required. However, such methods do not work if we want to disable an easier task (e.g. gender classification).

The authors in [25] are concerned with image filters that reduce discriminability of a particular classification task while preserving discriminability of another. Their methods differ from ours as they focus on discriminability of images for people and not algorithms. Similarly, the work in [11] proposes minimax filters that preserve privacy while providing utility and apply their framework on the application that allows face expression classification and prevents gender classification. However, they do not consider the case of adversarial data set that allows to train a relatively accurate gender classifier and is the main motivation for the closed environment, as demonstrated in Section 3.3.

**Fairness.** The goal of fairness in machine learning is to ensure that predictions are not biased or discriminatory [5]. To achieve fairness we may regularize the feature representations [27, 7, 17] or design predictors in a way that they obey the fairness constraints [12, 26].

A fair representation with respect to a certain variable does not necessarily imply privacy or non-identifiability of this variable. On the other hand, fairness can automatically result from the here proposed framework. Certain approaches such as the variational fair auto-encoder [17] or a censored representation trained using an adversary [7] also achieve non-identifiability. These methods can potentially be used in our proposed framework for closed privacy. However, our focus is not on a particular method or algorithm but rather on a wider framework that may encompass various algorithms.

**Other.** An orthogonal direction that offers privacy is homomorphic encryption [9], where models are trained and tested using encrypted data, and the prediction results are also returned in an encrypted form. Homomorphic encryption has been applied to neural networks in [3]. Such homomorphic encryption schemes are extremely computationally inefficient. Our proposed framework is as efficient as the desired utility, and again its goal is to block the entity from extracting protected information and not to block others from accessing data (encryption).

## 5. Conclusions

In this work we have introduced a framework that provides privacy to users within a closed environment and demonstrated its usage in an application where the face verification is possible and the gender classification is blocked. We also demonstrated that without the closed environment

assumption the privacy can not be guaranteed. There are many possible extension to the presented framework. In particular, it may be interesting to design methods that would allow sharing of users' data between various closed environments in a way consistent with privacy guarantees. Another relevant extension is to adaptive environments where users' data is used internally to improve the models.

## Acknowledgements

The work of G. Sapiro was supported in part by NSF, ONR, ARO, and NGA.

## References

- [1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. *ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, July 2016.
- [2] S. Ali Ossia, A. Shahin Shamsabadi, A. Taheri, H. R. Rabiee, N. D. Lane, and H. Haddadi. A hybrid deep learning architecture for privacy-preserving mobile analytics. *arXiv:1703.02952*.
- [3] N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig, and J. Wensing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. *International Conference on Machine Learning (ICML)*, pages 201–210, 2016.
- [4] C. Dwork. Differential privacy: A survey of results. *International Conference on Theory and Applications of Models of Computation*, pages 1–19, 2008.
- [5] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. *ACM Innovations in Theoretical Computer Science Conference*, pages 214–226, 2012.
- [6] C. Dwork and M. Naor. On the difficulties of disclosure prevention in statistical databases or the case for differential privacy. *Journal of Privacy and Confidentiality*, 2(1):93–107, 2010.
- [7] H. Edwards and A. Storkey. Censoring representations with an adversary. *International Conference on Learning Representations (ICLR)*, pages 1–14, Nov. 2016.
- [8] J. R. Gardner, P. Upchurch, M. J. Kusner, Y. Li, K. Q. Weinberger, K. Bala, and J. E. Hopcroft. Deep manifold traversal: Changing labels with convolutional features. *arXiv:1511.06421*, 2015.
- [9] T. Graepel, K. Lauter, and M. Naehrig. ML confidential: Machine learning on encrypted data. *International Conference on Information Security and Cryptology*, pages 1–21, 2012.
- [10] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems (NIPS)*, pages 513–520, 2007.
- [11] J. Hamm. Preserving privacy of continuous high-dimensional data with minimax filters. *Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 324–332, 2015.
- [12] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. *Advances in Neural Information Processing Systems (NIPS)*, pages 3315–3323, Oct. 2016.
- [13] B. Hitaj, G. Ateniese, and F. Perez-Cruz. Deep models under the GAN: Information leakage from collaborative deep learning. *arXiv:1702.07464*, 2017.
- [14] S. Israel, A. Caspi, D. W. Belsky, H. Harrington, S. Hogan, R. Houts, S. Ramrakha, S. Sanders, R. Poulton, and T. E. Moffitt. Credit scores, cardiovascular disease risk, and human capital. *Proceedings of the National Academy of Sciences*, 111(48):17087–17092, 2014.
- [15] D. Kifer and A. Machanavajjhala. No free lunch in data privacy. *ACM SIGMOD International Conference on Management of data*, pages 193–204, 2011.
- [16] D. Kifer and A. Machanavajjhala. Pufferfish: A framework for mathematical privacy definitions. *ACM Transactions on Database Systems*, 39(1):3:1–3:36, 2014.
- [17] C. Louizos, K. Swersky, Y. Li, M. Welling, and R. Zemel. The variational fair autoencoder. *International Conference on Learning Representations (ICLR)*, pages 1–11, Nov. 2016.
- [18] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. *IEEE Symposium on Security and Privacy*, pages 111–125, 2008.
- [19] H. Ng and S. Winkler. A data-driven approach to cleaning large face datasets. *IEEE International Conference on Image Processing (ICIP)*, pages 343–347, 2014.
- [20] S. J. Oh, R. Benenson, M. Fritz, and B. Schiele. Faceless person recognition; Privacy implications in social media. *European Conference on Computer Vision (ECCV)*, pages 19–35, 2016.
- [21] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. *British Machine Vision Conference*, pages 1–12, 2015.
- [22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [23] M. J. Wainwright, J. I. Michael, and J. C. Duchi. Privacy aware learning. *Advances in Neural Information Processing Systems (NIPS)*, pages 1430–1438, 2012.
- [24] L. Wasserman and S. Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389, 2010.
- [25] J. Whitehill and J. Movellan. Discriminately decreasing discriminability with learned image filters. *Computer Vision and Pattern Recognition (CVPR)*, pages 2488–2495, 2012.
- [26] M. B. Zafar, M. Gomez Rodriguez, and K. P. Gummadi. Fairness constraints : A mechanism for fair classification. *arXiv:1507.05259*, 2017.
- [27] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork. Learning fair representations. *International Conference on Machine Learning (ICML)*, pages 325–333, 2013.