Calibrating Practical Privacy Risks for Differentially Private Machine Learning

1st Yuechun Gu, Keke Chen
Trustworthy and Intelligent Computing Lab
Computer Science and Electrical Engineering, Cybersecurity Institute
University of Maryland, Baltimore County, Baltimore, USA
{ygu2,kekechen}@umbc.edu

***Abstract*—Differential privacy quantifies privacy through the privacy budget ϵ, yet its practical interpretation is complicated by variations across models and datasets. Recent research on differentially private machine learning and membership inference has highlighted that with the same theoretical ϵ setting, the likelihood-ratio-based membership inference (LiRA) attack success rate (ASR) may vary according to specific datasets and models, which might be a better indicator for evaluating real-world privacy risks. Inspired by this practical privacy measure, we study the positive correlation between the ϵ setting and ASR. We also find that for a specific dataset and a specific task we can lower the attack success rate by modifying the dataset. As a result, we may enable flexible privacy budget settings in model training. One dataset modification strategy is selectively suppressing privacy-sensitive features without significantly damaging application-specific data utility. We use the SHAP (or LIME) model explainer to evaluate features' privacy sensitivity and utility importance and develop an optimized feature-masking algorithm. We have conducted extensive experiments to show (1) the inherent link between ASR and the dataset's privacy risk in terms of a specific modeling task; (2) By carefully selecting features to mask, we can preserve more data utility with equivalent practical privacy protection and relaxed ϵ settings. The implementation details are shared online at https://github.com/RhincodonE/On-sensitive-features-and-empirical-epsilon-lower-bounds.**

***Index Terms*—Differential privacy, Feature sensitivity, Membership inference attack**

I. Introduction

With the fast advancement of deep learning techniques, companies can now leverage a huge amount of user-generated or user-related image and text data to train powerful large models. A main concern is these large models learn much more than what they are supposed to learn [1] – once the models are published, adversaries can use them to infer private information in the training data, e.g., via model inversion [2], [3], membership inference [4], [5], property inference [6], [7], and domain inference [8], [9] attacks. To address the private-information leak from published models, recent developments in privacy-preserving deep learning have incorporated the theory of differential privacy [10], e.g., the well-known Differentially Private Stochastic Gradient Descent (DP-SGD) [11].

A unique feature of differentially private machine learning is that the setting of privacy budget ϵ in (ϵ, δ)-differential privacy [12] is independent of applications and datasets. The smaller the ϵ setting, the better the privacy is preserved. It's considered an advantage since the learned privacy-preserving model will be resilient to attacks equipped with any type of adversarial knowledge. On the other side, it's also well believed that the differential privacy setting is too conservative to preserve data utility [13], [14]. For instance, the commonly accepted setting $\epsilon = 1$ has led to significant utility loss for many applications. In real-world applications, much higher ϵ values are often used to achieve better data utility [15], [16], which has raised concerns among researchers since no clear guidance is available to justify this practice. An intriguing question is whether and how we can relax the ϵ setting for different types of data and applications[1].

Is there a more practical auxiliary measure that can guide us to set the privacy parameters? This measure is likely dataset- and model-specific to complement the application-agnostic nature of differential privacy. The recent development on membership inference attacks (MIA) reminds us that it's possible. However, nobody has tried to apply MIA to this challenging problem.

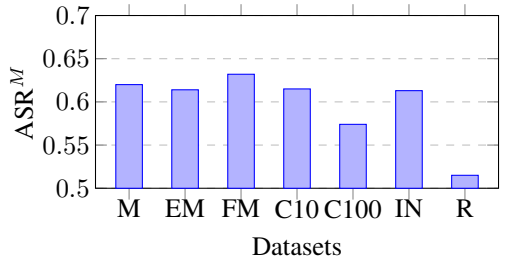
Carlini et al. [17] show that the likelihood- ratio-based membership inference attack (LiRA) can utilize the definition of differential privacy directly to conduct a sample-level membership inference attack. When applied to machine learning models, the essential idea of differential privacy is interpreted as follows: it's difficult (at the $\exp^{\epsilon}$ level) to distinguish whether a model used a sample in training or not. LiRA conducts a hypothesis-testing approach to derive the likelihood of a sample's membership. Based on LiRA, one can also derive the attack success rate (ASR) for each sample [18]. Due to the intrinsic link between LiRA and differential privacy [19], it's possible to use LiRA-based ASR as a measure to guide the selection of ϵ. We define a dataset-level ASR^M to indicate the worst-case identifiability of all samples in terms of a model M. We find this measure can precisely capture the *sensitivity level of dataset* in machine learning. Specifically, ASR^M close to 0.5 indicates that LiRA is close to random guessing and most samples are not sensitive in terms of the model M. Thus, we can use a larger ϵ setting. Otherwise, the

This work is partially supported by the National Science Foundation (Aware # 2232824).

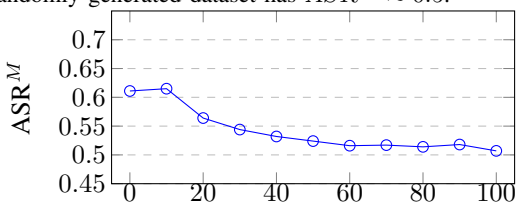
[1]Intuition says yes to the first question: When it comes to tasks that do not contain personally identifiable information, e.g., classifying animal pictures, we may use an arbitrary large ϵ.

dataset is highly sensitive under the model M, and a smaller ϵ value is needed to protect sample privacy. Initial evidence shows that this measure is indeed dataset- and model-specific. Figure 1a shows that ASR^M varies significantly over different datasets and corresponding models and correctly shows ≈ 0.5 for randomly generated datasets.

A simple experiment shows that ASR^M decreases with the increasing number of randomly masked features (see Figure 1b). This gives us an opportunity that *we may modify the dataset to adjust its sensitivity level of a dataset. As a result, we can relax the ϵ setting in DP-SGD.* Such a method is also possibly optimized to achieve good utility and privacy preservation.



(a) ASR^M variation across datasets. The model trained on a randomly generated dataset has $ASR^M \approx 0.5$.



(b) ASR^M decreases with more masked features (MNIST).

Fig. 1: ASR^M variation across different datasets and modified versions of the same dataset. All models are trained using DP-SGD with a theoretical $\epsilon = 8$. Abbreviations: M (MNIST), EM (EMNIST), FM (Fashion-MNIST), C10 (CIFAR-10), C100 (CIFAR-100), IN (ImageNet-1k subset), R (Randomly generated dataset).

Scope of Research. This paper studies two important problems: (1) a practical method for guiding the selection of ϵ value in differentially private machine learning, and (2) methods for modifying a dataset to allow relaxed ϵ settings to achieve a better balance between privacy and utility.

We have adopted the attack success rate (ASR) of LiRA as the basic measure to indicate the practical privacy risk of a dataset under a specific model release due to its tight connection with the definition of differential privacy. We have analyzed why this measure is data- and model-specific based on the definition of the hypothesis testing method (Section IV).

We further investigate possible dataset modification strategies that can lower the sensitivity of the dataset while not significantly damaging its utility. With lowered dataset sensitivity, we can apply relaxed ϵ settings, which can help achieve much better utility. The basic dataset-sensitivity reduction strategy is inspired by feature suppression used by data anonymization [20], i.e., masking sensitive features to protect privacy. Our method incorporates model explanation techniques, such as SHAP [21] and LIME [22], to identify *utility-essential* and *privacy-sensitive* features, respectively, from the target utility task (employed by the application) and an auxiliary identity-related task. For example, distracted driver classification is a critical utility task for smart vehicles, which helps identify tired or distracted drivers and prevent potential car accidents. The training data contains (input: driver image, label: type of distraction) pairs. It's easy to define an auxiliary identity task by replacing the labels with the identities of the drivers. We find that if the top features (e.g., pixels in images) from both types of tasks are not entirely overlapping, we can always extract a subset of features to suppress, which helps reduce the dataset sensitivity while minimizing the loss of data utility. The lowered dataset sensitivity allows us to set a higher ϵ value in DP-SGD.

We show that with our approach, we can achieve the same level of practical privacy protection (i.e., ASR^M) with much better-preserved utility in experiments. We have used well-known datasets, e.g., MNIST, CIFAR10, facial expression datasets, and distracted driver detection datasets [23] in experiments. Facial expression and distracted driver detection datasets are adopted to intuitively show how the identity and utility tasks are defined in our approach. On feature-masked datasets, we observe low ASR^M values (the practical privacy threats) even with theoretical ϵ values 5-10 times larger than the unmasked ones and 22%-41% better model quality from the relaxed ϵ setting.

In summary, we conclude our contributions as follows:

- We are the first team to investigate the data and model-specific nature of the LiRA attack for guiding the selection of theoretical ϵ settings.
- We show that the LiRA-based attack success rate, ASR^M, can be reduced via masking features of the target dataset, which can be optimized to achieve a better balance between utility and privacy. A lowered ASR^M allows a relaxed ϵ setting in DP-SGD.
- We have demonstrated that our methods work as expected in experiments on real datasets and modeling tasks. We have also shared the source code for researchers to reproduce the results.

The remaining sections cover the following topics. We explore related works in Section II, introduce key background information in Section III, provide insights into the core ideas of this paper and the feature masking method in Section IV. Results from our experiments are discussed in Section V.

II. Related Works

Differential privacy [24] is a well-accepted and theoretically justifiable privacy protection method. It has been integrated into deep learning via methods like DP-SGD [11], DP-Adam,

and DP-RMSProp [25]. Despite its theoretical solid guarantees, the application-specific setting of the privacy budget (ϵ) remains elusive, as the theoretical privacy budget ϵ setting is agnostic to datasets and models [26]. Real-world applications often use a relaxed setting, e.g., Apple's reported ϵ values of 2 to 16 for user activity analysis [15] and Google's 2.64 for COVID-19 community mobility reports [16], which raise concerns about the actual protection power.

Carlini et al. [17] proposed a likelihood ratio attack (LiRA) based on a black-box membership inference game using hypothesis testing. In their offline version of LiRA, adversaries aim to calculate the probability of rejecting the hypothesis that a target sample is not part of the training data distribution. According to Ahmed et al. [19], since LiRA operates within a black-box membership inference game, there exists a differential privacy distinguisher that gives the same results as LiRA's.

Some researchers also combine differential privacy with feature selection. Zhang et al. [27] consider the issue of privacy loss when data have a correlation in machine learning tasks and use differential privacy to privately select important features from datasets to avoid compromising privacy in the correlation of features. Pittaluga et al. [28] use differential privacy to privatize the features in the sample and design a feature-level differential privacy to guarantee privacy. However, none of the existing works interpret and guide the ϵ setting from the perspective of feature importance

Feature suppression was used in data anonymization [20] to hide those features that could potentially reveal sensitive information or personally identifiable details. We use it to lower a given dataset's sensitivity level, i.e., the LiRA attacking success rate ASR^M. It explores a possible way of combining traditional data anonymization methods and differential privacy to preserve better utility with solid theoretical privacy guarantees.

III. Prelimineries

A. Differential Privacy

Differential Privacy (DP) ensures that the analytical result on dataset cannot be used to distinguish the inclusion or exclusion of any individual data record. Mathematically, DP is defined as follows:

(ϵ, δ)-Differential Privacy (DP). Let $\mathcal{A}$ be a randomized function. We say that $\mathcal{A}$ provides (ϵ, δ)-differential privacy if for all datasets D_0 and D_1 differing on at most one element, and for all $O \subseteq \text{Range}(\mathcal{A})$,

$$\Pr[\mathcal{A}(D_0) \in O] \leq e^{\epsilon} \times \Pr[\mathcal{A}(D_1) \in O] + \delta$$

where δ is an ignorable small value, often $< 1/N$, and N is the number of samples in the dataset. ϵ is often called the privacy budget. In the context of machine learning, the adversary's ability to distinguish if the model $\mathcal{A}$ is trained on D_0 or D_1 is bounded by e^{ϵ}, and δ is the probability that the bound fails to hold.

Since a differentially private deep learning algorithm, such as DP-SGD [11], involves many steps of calculation and randomization, a specific privacy budget accounting method is used to aggregate step-wise privacy budgets to derive the estimate of the overall privacy budget ϵ, which is also called the *theoretical* ϵ.

Offline LiRA Attack. Following the definition of differential privacy where D and D/x differ in sample (x, y), where x and y are the feature vector and the label, correspondingly, Carlini et al. [17] present the likelihood-ratio membership inference attack. The online version of LiRA trains multiple shadow models $\{M_i\}$ on random sample datasets $\{D_i\}$, respectively, among which only half of the shadow datasets contain x, called in-domain cases, and the other half call out-domain cases. x in-domain and out-domain should be distinguishable if x is sensitive or vulnerable to membership inference attacks. Carlini et al. found that the distribution of log likelihood-ratio, $\phi(p) = \log(p/(1-p))$ of the output confidence level at the label y, $p = M_i(x)_y$, for the in-domain cases $x \in D_i$ is distinguishable from the out-domain cases, which serves as the basis of the LiRA attack. However, the online version incurs extremely high computational costs to train enough models to reliably derive the distributions of $\phi(p)$ for in-domain and out-domain cases, respectively. In contrast, offline LiRA relies solely on the out-domain samples' $\phi(p)$ distribution, significantly reducing computational costs with slightly reduced accuracy. For efficiency, we use the attack success rate (ASR) of offline LiRA as the indicator of the privacy sensitivity of sample (x, y). In offline LiRA, the out-domain $\phi(p)$ distribution is approximately described as a normal distribution $\mathcal{N}(\mu_{\text{out}}, \sigma^2_{\text{out}})$. To test the membership of (x, y) in the target model M, we measure how far $\phi(M(x)_y)$ is from out-domain $\phi(p)$ distribution by

$$\Lambda((x,y)) = 1 - \Pr[Z > \phi(q)], \text{ where } \quad Z \sim \mathcal{N}(\mu_{\text{out}}, \sigma^2_{\text{out}})$$
$$\phi(q) = \log \frac{q}{1-q}, \text{ where } \quad q = M(x)_y$$

We determine the new sample as a member sample when $\Lambda(x) > 0.5$, a decision rule adopted by [17] and proved effective.

The LiRA attack serves as the tool to estimate each sample's attack success rate. A batch ASR test method can be used to derive each sample's ASR as follows. We start by training n shadow models $M_{1\ldots n}$ on n randomly sampled subsets of the original dataset. Each shadow dataset, D^i_{IN}, is constructed by randomly picking samples from the entire dataset a probability of 0.5, and the remaining unselected samples are the out-domain samples, D^i_{OUT}. D^i_{OUT} samples go through the model M_i and the log-likelihood conversion of their outputs roughly follows a normal distribution, which is used to estimate the distribution parameters μ_{out} and σ^2_{out}. For each sample x_j in the dataset and each M_i, we have the ground-truth label $G(x_j, M_i) = IN$ if $x_j \in D^i_{IN}$; otherwise, $G(x_j, M_i) = OUT$. The LiRA(x_j, M_i) output (IN or OUT) is compared to the ground truth label $G(x_j, M_i)$. We compute

the ASR of sample x_j as follows:

$$ASR(x_j) = \sum_{i=1}^{n} \mathbf{I}(\text{LiRA}(x_j, M_i) == G(x_j, M_i))/n$$

Dataset Sensitivity. Furthermore, following the worst-case scenario of differential privacy, we can also measure the overall sensitivity of a dataset under a model M with $ASR^M = \max ASR(x_j), j = 1..N$, for N samples in the dataset. Intuitively, the highest sample ASR determines the sensitivity level of the dataset, i.e., the worst case of the dataset under the attack.

IV. Reducing Dataset Sensitivity via Feature Masking

In this section, we begin by discussing the rationale behind our method and outlining our threat modeling. Then, we present the main concepts and definitions of feature masking for reducing the empirical lower bound.

A. Motivation and Threat Modeling

Differentially private machine learning aims to train machine learning or deep learning models resilient to privacy attacks, such as membership inference [29]. A common DP learning algorithm, e.g., DP-SGD, allows the model owner to specify the privacy budget ϵ in (ϵ, δ)-differential privacy – the smaller the ϵ, the better the privacy is preserved. However, this privacy setting is independent of the dataset and the trained model, ignoring practical privacy risks the dataset may have. Our purpose is to study empirical privacy risk to derive better data- and model-specific privacy measures. Before diving into the details, we introduce the threat modeling for privacy attacks under the deep learning environment.

Protected assets: Identities of training data examples, i.e., the user who contributed or is related to a sensitive training sample. Examples: samples can be used to directly identify the owner, e.g., a face image; or samples contain unique features that can be used to link other data sources, e.g., tooth images, if linked to a person's dental records, can be used to identify a person.

Involved parties: Trusted model builders and curious model users. Model builders are trusted and may use DP-SGD to protect the trained model. The training process is secure and private, and no information is leaked. The learned model might be exposed via a service interface and used by a curious model user.

Adversarial capability. We assume the learned model might be exposed via a service interface under black-box privacy attacks. A black-box attack can only use the model prediction service as an API, but can obtain the prediction confidence vector for the classes. We also assume attackers know the training data distribution but not individual training data examples. Attackers will try to breach the privacy of individual training data examples, e.g., via membership inference attacks.

B. LiRA Attacking Success Rate is Data and Model Specific

We examine the definition of the offline version of LiRA to show that the LiRA ASR is data and model-specific.

Specifically, the offline LiRA based ASR calculation needs to train models, $M_1 \dots M_n$, each of which may or may not contain the target sample. M_i are trained with the same model architecture as the target model M on a sample set of D as depicted in Section III. Therefore, the testing is inherently tied to the target model and the dataset.

As expected, we have found that ASR^M varies by datasets and models in experiments. Furthermore, we have noticed by modifying the dataset, e.g., randomizing the labels or removing sensitive features, ASR can be significantly reduced (Figure 4).

This observation inspired us to find ways to reduce ASR^M for a specific dataset. We consider a few candidate methods as follows.

1. Lowering ϵ. In addition to the inherent randomness of the dataset, a lower privacy budget ϵ may push down the empirical privacy risk as well, but it also reduces more data utility, which is against our goal. However, it's still important to learn how the measure ASR^M changes with the setting of ϵ. If they are positively correlated, we can use this correlation to guide the setting of the privacy budget for a modified dataset, as we will show in the experiments.

2. Reducing dataset sensitivity. Since differential privacy is designed to protect the privacy of human-related data, intuitively, each dataset is associated with some level of *privacy sensitivity*, depending on how it can be used to explore private information. There are candidate methods for lowering dataset sensitivity. (1) Injecting noise at the instance level to disguise the private information and thus reduce the overall dataset sensitivity. This approach is adopted by locally differential privacy [30], which, however, leads to significantly more utility loss than the global differential privacy approach used by DP-SGD. (2) Identifying and removing privacy-sensitive features (i.e., the feature masking approach). If we model a privacy attack as a learning task – recovering the human-related identity information from the training examples, we might be able to identify the features that help this task most and then remove them. Experiments show that such a procedure indeed can help reduce dataset sensitivity represented with ASR^M.

Approximately equivalent ϵ settings. A key hypothesis is whether different ϵ settings for the original and modified datasets provide equivalent *practical* privacy protection. So far, there is no method for finding the exact practical privacy guarantee for a theoretical ϵ setting. However, we think the attacking success rate of LiRA might be a close one indicating a practical privacy guarantee. We thus define the equivalency as follows.

Let $M_{\epsilon,D}$ be a model trained with (ϵ, δ) differential privacy on D, and $M_{\epsilon,D'}$ trained with (ϵ', δ) differential privacy on a modified dataset of D. The ASRs are $ASR^{M_{\epsilon,D}}$ and $ASR^{M_{\epsilon',D'}}$, correspondingly. Assuming a small δ (e.g., $\delta =$

10^{-5}) is ignorable. We say the settings ϵ for D and ϵ' for D' are approximately equivalent, if

$$ASR^{M_{\epsilon,D}} \approx ASR^{M_{\epsilon',D'}}$$

In the following sections, we will explore the idea of feature masking to reduce ASR and then optimize the masking methods to preserve utility.

C. Optimized Feature Masking

Identity and utility tasks. Instead of examining a real privacy attack on the target model or dataset to identify the feature sensitivity level, we use *an identity task* as the surrogate to understand the feature sensitivity of the dataset. In contrast, we name the original modeling task tied to the specific application as *the utility task*. It's best to understand the two tasks in terms of real applications. For example, a facial expression recognition task is a utility task. Since the training data is collected from several persons' expressions, the data can also be used to learn who might be the contributors, which is defined as the identity task. In another example, a self-driving dataset is originally used to identify all kinds of objects from the captured scene: roads, trees, sidewalks, pedestrians, etc., which is the utility task. The identity task can be whether the scene contains persons, which can be used to rank the person-related features that are potentially linked to privacy protection.

One may argue that such an identity task may not be easy to define in some applications. The role of identity task represents the model builder's *best* knowledge about privacy-sensitive information. It provides some hints to our approach that masking certain features might reduce the ASR^M value. With the ASR^M calculation procedure, the model builder can always verify whether the masking helps privacy protection.

Feature privacy sensitivity. We consider a sensitive dataset D comprising N samples $\{x_i\}, i = 1..N$, each with K features $\{F_j\}, j = 1..K$. The first crucial step in our method is to ascertain the sensitivity level of each feature in D. This sensitivity level, which we term the *Feature Privacy Sensitivity* (s_j) for each feature F_j, is a pivotal factor in our feature masking strategy. With a surrogate identity model $I(D, Z) : z = g(x), z \in Z$, where Z is a set of identities, we design a method to quantify the feature-level sensitivity. Intuitively, this model, trained on a labeled dataset $\{(x_i, z_i)\}$, with crafted identity-related labels $z_i \in Z$, can output identity-related information, e.g., whether an image x contains a person z. The feature importance, as it tells the feature's contribution to the identity task, can be used to define the feature privacy sensitivity s_j, which can be captured by a model explainer, like SHAP [21] or LIME [22]. The effect of different explainers has been evaluated in experiments. For simplicity, assume we use SHAP values of the identity task as the feature sensitivity. We only consider positive SHAP values as they indicate the features' positive contribution to the identity task:

$$s_j = \begin{cases} s_j & \text{if } s_j > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Upon determining the feature privacy sensitivity, our next step is to mask features based on their sensitivity levels and their relations to the utility task. We consider both privacy loss and data utility and try to achieve an optimized balance between the two as follows

Utility-optimized masking. Similar to the definition of feature privacy sensitivity, e.g., with SHAP values, we can also derive feature utility sensitivity in terms of the utility task and a utility model $M(D, Y)$, where Y is a set of utility labels, e.g., facial expressions in facial expression datasets. We define the feature F_j's *utility sensitivity*, u_j, as follows:

$$u_j = \begin{cases} u_j, & \text{if } u_j > 0 \\ 0, & \text{otherwise} \end{cases}$$

Let the dot-product of the normalized utility-sensitivity vector u and privacy-sensitivity vector s represent the extent of shared features between the identification and utility models. Ideally, when u and s are orthogonal, i.e., $u^T s = 0$, there is no overlap between identity-related and utility-related features. This would make it straightforward for the data owner to eliminate identity-related features without affecting utility.

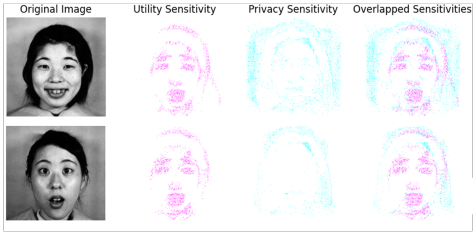


Fig. 2: An example of feature utility sensitivity and privacy sensitivity

However, as shown in Figure 2, it's likely to have features with both high utility and privacy sensitivity levels, e.g., some critical features in face images shared by both the expression classification and identity tasks. Thus, achieving $u^T s = 0$ in real-world scenarios might be impossible. Instead, we can optimize the masking mechanism to maximize the utility with a desired amount of privacy preserved. Specifically, we formulate an optimization problem as follows to find a masking vector m:

$$\begin{aligned} \underset{m}{\text{argmax}} \quad & m^T u \\ \text{s.t.} \quad & m^T s < (1-\alpha) \sum_{j=1}^{K} s_j, \end{aligned}$$

where α is the model owner's desired privacy preservation level and we use $\sum s_j$ to approximately represent the total amount of privacy information the features bring. The optimization will try to pick up the optimal mask to maximize the amount of utility, i.e., $m^T u$, with acceptable privacy loss. This linear optimization problem is often solvable with standard techniques.

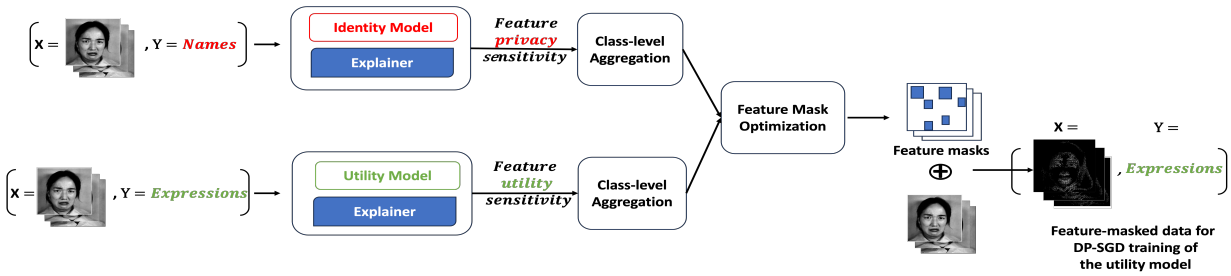


Fig. 3: Pipeline of feature masking

Implementation details. As shown in Algorithm 1 and Figure 3, our implementation uses class-wise feature privacy sensitivity generation to preserve utility best and reduce privacy information during feature masking. Specifically, the data owner uses the model explainer to generate a privacy-sensitivity vector $s_{i,z}$ for a sample $x_{i,z}$ and utility-sensitivity vector $u_{i,y}$ for a sample $x_{i,y}$, where the same sample x_i showing up in the two different tasks and labeled as class z and y from label set Z and Y, respectively. By summing up the feature sensitivity vectors for all samples in one class, we get a class-wise feature privacy-sensitivity vector S_z and utility-sensitivity vector U_y. In case the optimization problem is not solvable[2], we use the top-k% method instead. Specifically, we mask the top-k% privacy-sensitivity and define the mask for each class.

With a class-wise binary mask m_z in place, we can process images from class z to mask their features accordingly: $x'_{i,z} = x^T_{i,z} m_z$.

Algorithm 1 Feature-masking mechanism

Require: Identity Model $I(D, Z)$, Image set D of size N_d with classes $1, 2, \ldots, Z$, size N_z of class z, Explainer $E()$
Ensure: Masked Image set D'

function FEATUREMASKING($M(D, Y)$, D)
  for $i \leftarrow 1, N_d$ **do**
    $s_{i,z} \leftarrow E(I(D, Z), x_{i,z})$ ▷ Compute individual privacy-sensitivity vector
    $u_{i,y} \leftarrow E(M(D, Y), x_{i,y})$ ▷ Compute individual utility-sensitivity vector
  end for
  $S_z \leftarrow \frac{1}{N_z} \sum_{i=1}^{N_z} s_{i,z}$
  $U_y \leftarrow \frac{1}{N_y} \sum_{i=1}^{N_y} s_{i,y}$ ▷ Aggregate to class-wise
  $m_z \leftarrow \begin{cases} \text{Linear optimization} & \text{if solvable} \\ \text{Top-k\% method} & \text{otherwise} \end{cases}$
  for $i \leftarrow 1, N_d$ **do**
    $x'_{i,z} \leftarrow x^T_{i,z} m_z$ ▷ Mask the images
    Add $x'_{i,z}$ to D'
  end for
  return D'
end function

[2]This occurs with the probability of 3% in our experiments.

V. EXPERIMENTS

We have conducted comprehensive experiments to answer the critical research questions in our study. 1) How does ASR^M change over different datasets or versions of modified datasets? 2) With different versions of feature-masked data, how do the theoretical ϵ values differ at the same ASR^M of utility models? 3) How do different feature masking strategies affect data utility on the utility models? 4) How does the choice of model explainer affect performance? And 5) What are the costs of the proposed methods?

A. Setup

Datasets: In addition to widely-used datasets such as MNIST, CIFAR10, and ImageNet-1K, we employ three facial expression datasets: RaFD [31], JAFFE [32], and TFEID [33], along with a distracted driver activity recognition dataset, 100-Driver [23], to demonstrate the performance of feature masking in practical scenarios. These datasets were selected because they clearly contain sensitive information, and they intuitively show how two distinct sets of labels, utility labels and identity labels (i.e., the identities of the contributors), can be defined on the same dataset, corresponding to the utility and identity tasks. The utility labels represent 7 types of facial expressions in the facial expression datasets and 22 categories of distracted drivers' activities in the driver activity recognition dataset. Each dataset has also clearly defined which human subject each image belongs to, certainly without real identity information. In training models, we split each dataset into training and testing sets using an 80:20 ratio.

Necessary preprocessing steps have been performed on these datasets. The RaFD dataset contains images of 67 persons, each of which exhibits 8 distinct emotional expressions. The JAFFE dataset includes 210 images of 10 female subjects, each of which has 7 expressions and each expression per person has three instances. The TFEID dataset consists of images from 40 subjects, each of which has 8 different facial expressions. Since the number of classes can degrade the quality of model explainers due to the small dataset size. To mitigate this, we randomly selected 10 identities from the RaFD and TFEID datasets, resulting in 80 images per dataset. The 100-Driver dataset includes 100 identities, each performing 22 distinct activity types. For each person, each activity type contains 16 instances of different settings,

e.g., with or without wearing glasses, multiple cameras with different angles in one vehicle, and different vehicles. In total, there are 470K images. Due to computational limitations, we randomly sample the same number of images for both identity and distraction labels, to generate a subset with 35K images.

Model training. When using Differentially Private Stochastic Gradient Descent (DP-SGD) in training models, we adopted the following parameter settings. We use ResNet-101 for ImageNet-1K and 100-Driver[23] datasets for both the utility and identity models due to the higher complexity of these datasets, and ResNet-18 for other smaller datasets. A learning rate of 0.001 and epochs of 150 were set. We used a batch size of 5. We adopted an early termination mechanism to stop training to prevent overfitting when no improvement was seen on the validation set for a certain number of epochs.

Dataset	Identity	Utility	Identities	Size
RaFD	0.737 (+/-0.031)	0.293 (+/-0.027)	10	80
JAFFE	0.832 (+/-0.025)	0.327 (+/-0.011)		210
TFEID	0.719 (+/-0.028)	0.286 (+/- 0.024)		80
100-Driver	0.913 (+/-0.012)	0.721 (+/- 0.016)		35208

TABLE I: Baseline models on the three datasets.

Model Explainers. To evaluate the impact of model explainers, we selected two of the most widely used methods: SHAP[3] and LIME[4]. Since SHAP is more efficient for batch processing and performs better than LIME, we used it for all the experiments. We also show how SHAP is better than LIME in one set of experiments.

Evaluation metrics. The ASR^M values are generated with the evaluation method described in Section III-A. ASR^M is the maximum sample ASR over all samples in a dataset to be consistent with the worst-case definition of differential privacy. $ASR^{M,\epsilon}$ indicates ASR^M on DP-SGD trained models for a specific ϵ setting. We use model accuracy to evaluate the quality of utility models.

Deriving ASR^M. To conduct the LiRA attack, we train 1000 shadow models[5] – for each shadow model we flip coins for each sample to split the dataset into in-domain and out-domain samples, which are used to train the shadow model. Then, we apply the offline version of LiRA [17] to test all samples on the shadow model and derive ASR^M. The smaller the ASR^M, the more difficult it is to distinguish samples, and thus the privacy is better preserved (or the dataset is less sensitive, if no DP-SGD is applied).

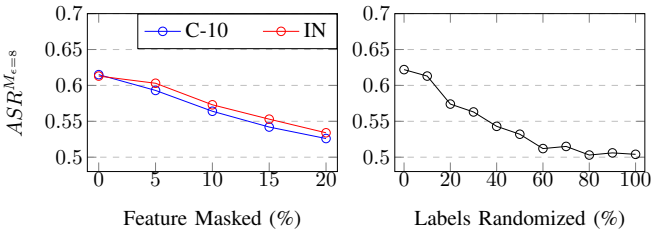
B. Result Analysis: ASR^M for Modified Datasets

The first goal of our approach is to identify a data- and model-specific measure that can evaluate the practical privacy risk. We have shown some examples earlier in Figure 1b that ASR^M can serve this purpose well. Here, we show more detailed experiments to see how different dataset qualities may affect ASR^M. In Figure 4, we take the well-known datasets

[3]https://github.com/shap/shap

[4]https://github.com/marcotcr/lime

[5]Training 1000 ResNet-18 models on an NVIDIA TITAN V100 averagely spend 19 hours, and ResNet-101 spend 42 hours



(a) ASR^M drops with more removed features.(C-10,IN). $\epsilon = 8$

(b) ASR^M drops with more labels are randomized(MNIST).

Fig. 4: Figure (a) shows the decreasing trends of ASR^M when top-k important features are masked for CIFAR-10 and ImageNet-1K. Figure (b) demonstrates that with the increasing percentage of randomized labels (representing data quality reduction), ASR^M decreases for MNIST.

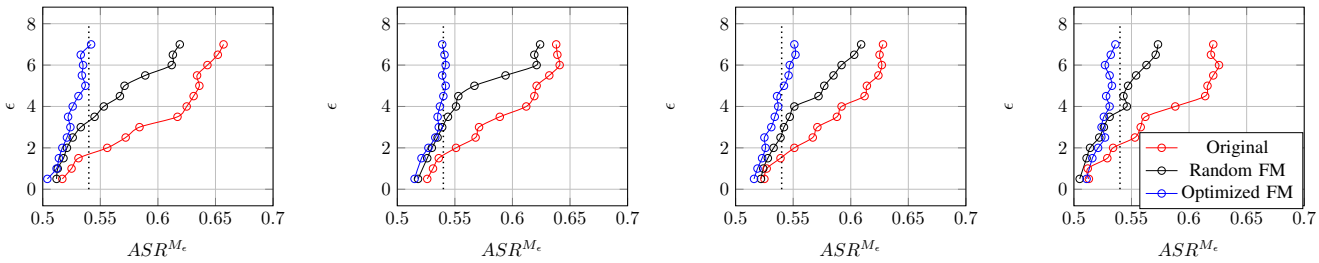
CIFAR-10 and ImageNet-1K and progressively mask the top-ranked features (pixels), and their ASR^M drops correspondingly. To understand how label quality may affect ASR^M, we also randomize a portion of MNIST's label, and we have observed a similar decreasing trend. These results motivated us to explore the feature masking approach to change the dataset sensitivity to achieve better utility and privacy balances.

C. Result Analysis: $ASR^{M,\epsilon}$ vs theoretical ϵ

To understand the correlation between ϵ and $ASR^{M,\epsilon}$, we take the three versions of datasets for experiments: the original (Original), the dataset with 30% randomly masked features (Random FM), and that with optimized feature masking (Optimal FM). Each point in Figures 5 represents a set of experiments: we apply DP-SGD with the specific theoretical ϵ setting to train a set of identity models and then use the LiRA to derive the corresponding ASR^M. The result in Figure 5 shows that (1) ASR^M is positively correlated with the theoretical ϵ. However, the correlation is stronger for the original data. Due to the reduced privacy sensitivity, the FM methods have a narrower range of ASR^M. (2) The optimized FM has advantages in significantly lower ASR^M, which implies reduced practical privacy risk. (3) For the same level of ASR^M, we can probably use a much higher theoretical ϵ for DP-SGD. For example, for JAFFE, $\epsilon = 1$ gives ASR^M around 0.525 (the 2nd red dot from left to right on the DP-SGD curve in Figure 7a. With the same ASR^M, the ϵ for the optimized feature-masked dataset can be relaxed to around 4. The relaxed ϵ will allow for preserving more useful information for the utility models, which will be examined next.

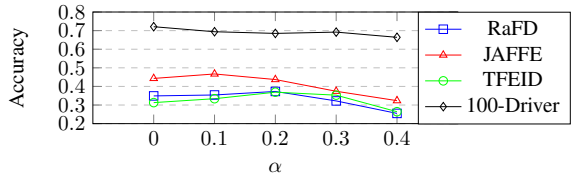
D. Result Analysis: Optimizing Feature Masking

As discussed in Section IV-C, selecting an appropriate α for generating feature masks is critical. Here, α represents the extent of dataset utility sensitivity to be reduced. Figure 6a illustrates how the performance of the utility model changes with different α settings. Interestingly, increasing α initially enhances the utility model's performance, but beyond a certain

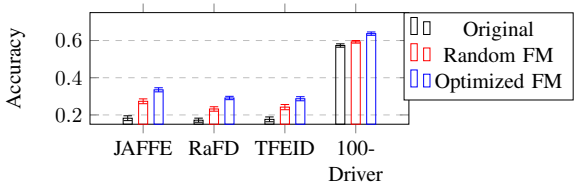


(a) JAFFE (SHAP) (b) RaFD (SHAP) (c) TFEID (SHAP) (d) 100-Driver(SHAP)

Fig. 5: The relationship between $ASR^{M,\epsilon}$ and theoretical ϵ over utility models for optimized masked, random masked, and original dataset. The α parameter setting for optimized feature masking: α at 0.1 for JAFFE, 0.2 for both RaFD and TFEID, and 0.3 for 100-Driver as suggested later in Figure 6a. For the random feature masking approach, we omit 30% of the features at random, i.e., the same number of masked features in the optimized feature masking setting.



(a) Accuracy of utility models without DP-SGD across different α values.



(b) Performance of the utility model with the DP-SGD and ϵ values at $ASR^M = 0.54$, detailed in Table II

Fig. 6: FM optimization algorithm helps preserve better data utility

threshold, it begins to decline for the facial expression datasets. Specifically, we observe that the utility model achieves the best accuracy at $\alpha = 0.2$ for RaFD and TFEID, and at $\alpha = 0.1$ for JAFFE. For the 100-Driver dataset, $\alpha = 0.1$ and $\alpha = 0.3$ both result in comparable performance, but for better privacy protection, we select $\alpha = 0.3$.

To explore the impact of optimization on utility models trained with DP-SGD, we compare random feature masking and optimized feature masking. We assess how these methods affect the performance of utility models when training with DP-SGD at ϵ values that correspond to $ASR^M \approx 0.54$ (marked by the black dotted line in Figure 5). Figure 6b and Table II demonstrate that applying optimized feature masking to datasets results in significantly improved ϵ and performance of utility models when $ASR^M \approx 0.54$, compared to solely using DP-SGD. We conclude that utility models trained on datasets processed by optimized feature masking offer comparable privacy protection while allowing for much larger ϵ values during DP-SGD training. This increased ϵ enables the utility models to achieve a better privacy-utility tradeoff compared to models trained on the original datasets or those using random feature omission.

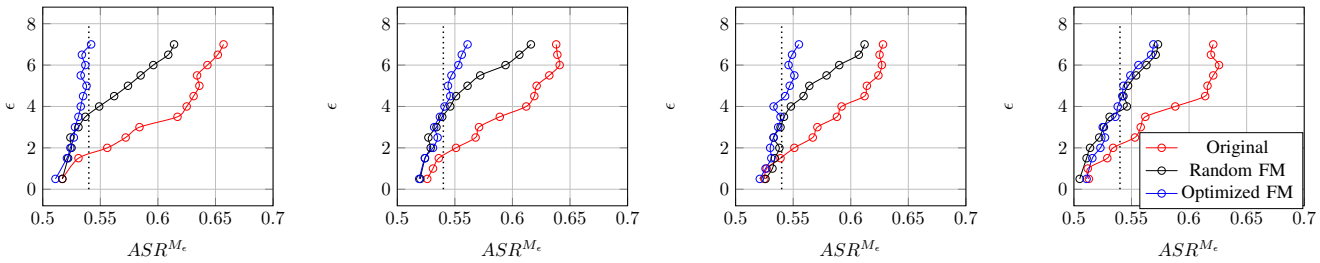
Dataset	Orig. DP-SGD		FM+DP-SGD	
	ϵ	Acc	ϵ	Acc
JAFFE	1.73	0.1824	6.87	0.335
RaFD	1.65	0.1717	6.54	0.291
TFEID	1.78	0.1762	4.33	0.287
100-Driver	2.13	0.5731	7.14	0.637

TABLE II: ϵ and Accuracy of utility models when $ASR^M \approx 0.54$ (the black dotted line in Figure 7).

E. Result Analysis: Impact of Explainer

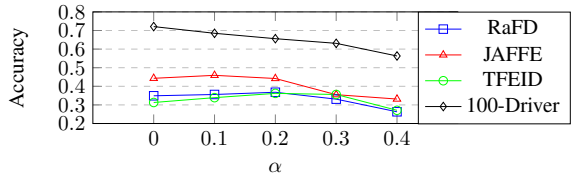
In the previous experiments, we have used SHAP to implement the feature masking algorithm. It's also interesting to understand whether the method is explainer-specific. In this section, we replace SHAP with LIME and reproduce the results. As shown in Figure 7 and Figure 8, LIME does not change the patterns much for both privacy protection and utility preservation on the facial expression datasets. However, on the 100-Driver dataset, the optimized FM curve overlaps with the random FM curve, even with a smaller $\alpha = 0.1$ (Figure 7d), as opposed to a larger $\alpha = 0.3$ for SHAP (Figure 5d). This suggests that LIME-based optimized FM may not perform as well as SHAP on more complex model architectures. We also observe that models using LIME exhibit larger standard deviations in accuracy (Figure 8b) compared to Figure 6b. Figure 9 looks into the stability of SHAP and LIME when applied with optimized FM at $\epsilon = 7$. SHAP consistently provides better stability across all datasets. We attribute this to LIME being a localized interpretation method [22], which is more appropriate for simpler models where localized interpretability suffices. Moreover, LIME's reliance on random sampling may introduce additional variance compared to SHAP.

In conclusion, when deploying optimized FM in practice, SHAP appears to be the more suitable backbone explainer than LIME.

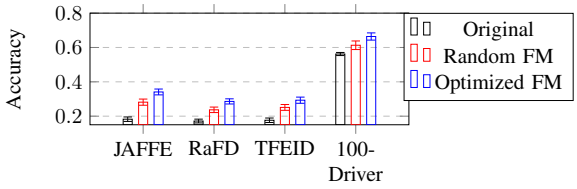


(a) JAFFE (LIME) (b) RaFD (LIME) (c) TFEID (LIME) (d) 100-Driver(LIME)

Fig. 7: We use LIME to reproduce the results. The relationship between ASR^{M_ϵ} and theoretical ϵ over utility models for optimized masked, random masked, and original datasets show a similar trend with using SHAP (Figure 5). Parameter setting for optimized feature masking: α at 0.1 for both 100-Driver and JAFFE, 0.2 for both RaFD and TFEID, as suggested later in Figure 8a. For the random feature masking approach, we omit 30% of the features at random, which approximates the number of masked features by optimized feature masking.



(a) Accuracy of utility models without DP-SGD across different α values.



(b) Performance of the utility model at $ASR \approx 0.54$, depicted as a black dotted line in Figure 7.

Fig. 8: LIME-based FM optimization algorithm also helps preserve better data utility

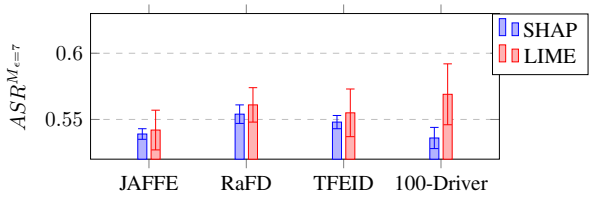
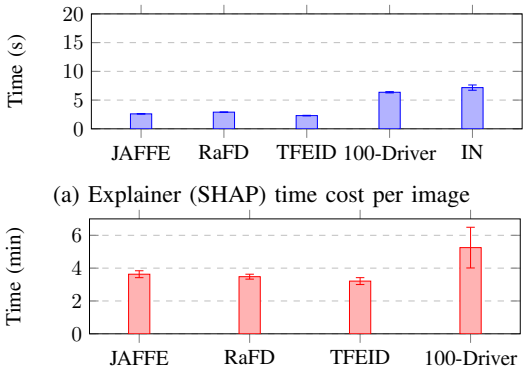


Fig. 9: ASR^M when training utility model with $\epsilon = 7$ and applying optimized FM. SHAP is more stable than LIME.

F. Result Analysis: Time Cost

In this section, we evaluate the time cost of the optimized feature masking (FM) method. This approach involves two main steps: (1) generating feature privacy and utility sensitivity levels for each image with an explainer, and (2) performing class-wise optimization of the feature masks. Figure 10 presents the average time cost for three facial expression datasets using the ResNet-18 model and for the 100-Driver dataset using the ResNet-101 model with SHAP as the explainer. To assess the potential time cost for larger, real-world datasets, we also include the time cost for the ImageNet-1K (IN) dataset on the ResNet-101 model. Although the per-image time cost remains in the seconds range (Figure 10a), thanks to SHAP's batch-processing capability, the total time for large datasets like 100-Driver and ImageNet-1K can accumulate to hours or even days. Methods like sampling can be applied to reduce the first-stage cost. In contrast, the optimization of feature masks takes only a few minutes for the facial expression datasets, 100-Driver, and ImageNet-1K (Figure 10b).



(a) Explainer (SHAP) time cost per image

(b) Optimized FM generation time cost per class

Fig. 10: Time cost of sensitivity generation by SHAP and optimizing feature masks across datasets.

G. Discussion

We have experimented with a variety of datasets to observe the data- and model-specific characteristics of ASR^M. Among these datasets, we have used facial expression and 100-driver datasets for identity-related feature masking experiments. For datasets with no clear identity elements, like those used in self-driving car technology featuring pedestrians, buildings, and address plates, defining identity tasks can be trickier, probably tightly related to the analysis of application-specific sensitivity. Nevertheless, our work has established a framework to understand the inherent sensitivity of the dataset and guide the

setting of privacy parameters for differential privacy methods. For a specific application and dataset, one can always try different identity-related tasks and evaluate them within our framework.

VI. Conclusion

The parameter setting of differentially private machine learning is detached from specific applications, datasets, and models, which is considered a unique feature. However, to better understand the tradeoff between privacy and utility, e.g., finding a justifiable relaxed ϵ setting, we have to look into more data- and model-specific auxiliary measures. Recent studies on likelihood-ratio-based membership inference attacks, LiRA, have given us an effective tool to tackle this challenging problem. We have shown that the LiRA-based attack success rate ASR^M can serve as a well-justified data- and model-specific measure for evaluating practical privacy risks for models trained with or without DP-SGD. We explore the factors affecting ASR^M to identify a better way to set the theoretical differential privacy budget. With the proposed optimized feature masking methods, we demonstrated in experiments that models trained on datasets with masked features and relaxed ϵ settings can still achieve equivalent practical privacy protection (i.e., similar ASR^M levels) compared to original DP-SGD. Meanwhile, the optimized masking method preserves better model quality. Our approach offers a promising framework for fine-tuning the privacy budget setting in terms of specific data and models to achieve better privacy and utility balances.

References

[1] A. Zytek, I. Arnaldo, D. Liu, L. Berti-Equille, and K. Veeramachaneni, "The need for interpretable features: Motivation and taxonomy," *SIGKDD Explor. Newsl.*, vol. 24, no. 1, pp. 1–13, jun 2022. [Online]. Available: https://doi.org/10.1145/3544903.3544905

[2] M. Fredrikson et al., "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1322–1333.

[3] Y. Zhang and et al., "The secret revealer: Generative model-inversion attacks against deep neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 253–261.

[4] R. Shokri et al., "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 3–18.

[5] C. A. Choquette-Choo et al., "Label-only membership inference attacks," in *International conference on machine learning*. PMLR, 2021, pp. 1964–1974.

[6] K. Ganju et al., "Property inference attacks on fully connected neural networks using permutation invariant representations," in *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, 2018, pp. 619–633.

[7] M. Parisot et al., "Property inference attacks on convolutional neural networks: Influence and implications of target model's complexity," in *18th International Conference on Security and Cryptography, SECRYPT 2021*. SciTePress, 2021, pp. 715–721.

[8] Y. Gu and K. Chen, "Gan-based domain inference attack," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 12, 2023, pp. 14 214–14 222.

[9] Y. G. et al., "Adaptive domain inference attack," in *https://arxiv.org/abs/2312.15088*, 2023.

[10] M. Gong et al., "A survey on differentially private machine learning," *IEEE computational intelligence magazine*, vol. 15, no. 2, pp. 49–64, 2020.

[11] M. Abadi et al., "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[12] C. Dwork, "Differential privacy: A survey of results," in *International conference on theory and applications of models of computation*. Springer, 2008, pp. 1–19.

[13] Z. Jorgensen et al., "Conservative or liberal? personalized differential privacy," in *2015 IEEE 31St international conference on data engineering*. IEEE, 2015, pp. 1023–1034.

[14] A. El Ouadrhiri and A. Abdelhadi, "Differential privacy for deep and federated learning: A survey," *IEEE access*, vol. 10, pp. 22 359–22 380, 2022.

[15] A. Differential Privacy Team, "Learning with privacy at scale," https://machinelearning.apple.com/research/learning-with-privacy-at-scale, 12 2017.

[16] A. A. et.al., "Google covid-19 community mobility reports: Anonymization process description (version 1.1)," 2020.

[17] N. Carlini et al., "Membership inference attacks from first principles," in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 1897–1914.

[18] N. Carlini, M. Jagielski, C. Zhang, N. Papernot, A. Terzis, and F. Tramer, "The privacy onion effect: Memorization is relative," *Advances in Neural Information Processing Systems*, vol. 35, pp. 13 263–13 276, 2022.

[19] A. Salem et al., "Sok: Let the privacy games begin! a unified treatment of data inference privacy in machine learning," in *2023 IEEE Symposium on Security and Privacy (SP)*, 2023, pp. 327–345.

[20] B. C. M. Fung et al., "Privacy-preserving data publishing: A survey of recent developments," *ACM Computing Survey*, vol. 42, pp. 14:1–14:53, June 2010.

[21] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[22] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, pp. 1135–1144. [Online]. Available: https://doi.org/10.1145/2939672.2939778

[23] J. Wang et al., "100-driver: A large-scale, diverse dataset for distracted driver classification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 7, pp. 7061–7072, 2023.

[24] C. Dwork, "Differential privacy," in *International colloquium on automata, languages, and programming*. Springer, 2006, pp. 1–12.

[25] R. Gylberth et al., "Differentially private optimization algorithms for deep neural networks," in *2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2017, pp. 387–394.

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

[27] T. Zhang et al., "Correlated differential privacy: Feature selection in machine learning," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2115–2124, 2020.

[28] F. Pittaluga and B. Zhuang, "Ldp-feat: Image features with local differential privacy," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 17 580–17 590.

[29] H. Hu et al., "Membership inference attacks on machine learning: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 11s, pp. 1–37, 2022.

[30] M. Yang et al., "Local differential privacy and its applications: A comprehensive survey," *Computer Standards & Interfaces*, p. 103827, 2023.

[31] O. Langner et al., "Presentation and validation of the radboud faces database," *Cognition and emotion*, vol. 24, no. 8, pp. 1377–1388, 2010.

[32] M. J. Lyons, "" excavating ai" re-excavated: debunking a fallacious account of the jaffe dataset," *arXiv preprint arXiv:2107.13998*, 2021.

[33] L. Chen and Y. Yen, "Taiwanese facial expression image database," Brain Mapping Laboratory, Institute of Brain Science, National Yang-Ming University, Taipei, Taiwan, Tech. Rep., 2007.