DP-UTIL: Comprehensive Utility Analysis of Differential Privacy in Machine Learning

Ismat Jarin
ijarin@umich.edu
University of Michigan, Dearborn
United States

Birhanu Eshete
birhanu@umich.edu
University of Michigan, Dearborn
United States

ABSTRACT
Differential Privacy (DP) has emerged as a rigorous formalism to quantify privacy protection provided by an algorithm that operates on privacy sensitive data. In machine learning (ML), DP has been employed to limit inference/disclosure of training examples. Prior work leveraged DP across the ML pipeline, albeit in isolation, often focusing on mechanisms such as gradient perturbation.

In this paper, we present DP-UTIL, a holistic utility analysis framework of DP across the ML pipeline with focus on input perturbation, objective perturbation, gradient perturbation, output perturbation, and prediction perturbation. Given an ML task on privacy-sensitive data, DP-UTIL enables a ML privacy practitioner to perform holistic comparative analysis on the impact of DP in these five perturbation spots, measured in terms of model utility loss, privacy leakage, and the number of truly revealed training samples.

We evaluate DP-UTIL over classification tasks on vision, medical, and financial datasets, using two representative learning algorithms (logistic regression and deep neural network) against membership inference attack as a case study attack. One of the highlights of our results is that prediction perturbation consistently achieves the lowest utility loss on all models across all datasets. In logistic regression models, objective perturbation results in lowest privacy leakage compared to other perturbation techniques. For deep neural networks, gradient perturbation results in lowest privacy leakage compared to other perturbation techniques. Objective perturbation, gradient perturbation, output perturbation, and prediction perturbation. Given an ML task on privacy-sensitive data, DP-UTIL enables a ML privacy practitioner to perform holistic comparative analysis on the impact of DP in these five perturbation spots, measured in terms of model utility loss, privacy leakage, and the number of truly revealed training samples.

We evaluate DP-UTIL over classification tasks on vision, medical, and financial datasets, using two representative learning algorithms (logistic regression and deep neural network) against membership inference attack as a case study attack. One of the highlights of our results is that prediction perturbation consistently achieves the lowest utility loss on all models across all datasets. In logistic regression models, objective perturbation results in lowest privacy leakage compared to other perturbation techniques. For deep neural networks, gradient perturbation results in lowest privacy leakage compared to other perturbation techniques. Objective perturbation, gradient perturbation, output perturbation, and prediction perturbation.

CCS CONCEPTS
• Security and privacy → Security services; Privacy-preserving protocols;

KEYWORDS
Differential Privacy, Machine Learning, Membership Inference Attack

1 INTRODUCTION
The continued progress in machine learning (ML) has resulted in a line of applications where privacy-sensitive datasets, such as medical records, are increasingly being used to train and deploy ML models. Almost parallel to the ML progress, especially with the impressive performance of deep neural network (DNN) models, privacy attacks against ML models also emerged because of models memorizing sensitive details about the training set. Prior work has shown that ML models trained on privacy-sensitive data are vulnerable to a range of privacy-motivated attacks such as membership inference [30], attribute inference [23], model inversion [10], and model parameters inference [33]. Moreover, unintended memorization of privacy-sensitive details at training time results in inadvertent leakage of private data at prediction time [4, 31]. A practical challenge in the context of ML is that of striking a reasonable balance between the utility (e.g., accuracy) of the ML model and the privacy of subjects from whom training data is obtained.

To counter ML privacy attacks, differential privacy (DP) [8] has emerged as a rigorous notion to formalize and measure privacy guarantee based on a parameter called privacy budget. To an individual that contributes a data point to a training set of a ML model, DP promises: "an adversary learns almost the same information about you whether or not you belong to the training data set". In an attempt to keep this promise, across the ML pipeline, DP has been used to limit inference/disclosure of ML training examples pre-training (via input perturbation [7, 20]), during training (via objective perturbation [6] and gradient perturbation [1]), and post-training (via output perturbation [6] and prediction perturbation [13, 27, 28]).

Motivation: While prior work [6, 7, 13, 14, 20, 27, 28] has leveraged these perturbation mechanisms across the ML pipeline, these applications were done in isolation, often focused on one of the perturbation methods (e.g., gradient perturbation). As a result, given a privacy-sensitive dataset and an ML task (e.g., medical image classification), there is a lack of a holistic assessment methodology as to the utility of DP when it is employed before, during, and after training. More precisely, isolated applications of DP in prior work do not shed light on how the aforementioned perturbation
methods compare in their effectiveness and, more importantly, in their trade-offs. Moreover, the usefulness of these alternative perturbation methods across diverse datasets and ML models remains under-explored.

**Overview:** In this paper, we present DP-UTIL, a framework for holistic utility analysis of DP across the ML pipeline. DP-UTIL enables an ML privacy practitioner to analyze perturbation methods in terms of their impact on model utility, privacy leakage, and actual number of privacy-sensitive samples inferred by an adversary. The benefit of DP-UTIL is twofold. First, it enables an ML privacy practitioner to have an across-the-ML-pipeline view of the impact of DP using standard metrics such as model accuracy and privacy leakage. Second, it enables comparative analysis on the suitability of one DP mechanism (e.g., gradient perturbation) against another DP mechanism (e.g., output perturbation) so that the practitioner makes informed decisions as to where to plug DP in the ML pipeline for a given task and dataset.

We evaluate DP-UTIL over classification tasks on three datasets covering vision, medical, and financial domains. We use membership inference as a case study attack to analyze privacy leakage and actual number of revealed data samples. To shed light on the difference between convex and non-convex optimization formulations used in training ML models, we use Logistic Regression (LR) and Deep Neural Network (DNN), respectively, as representative ML models due to the wide usage of both on privacy-sensitive datasets. For LR models, we evaluate five perturbation methods: input perturbation [11], objective perturbation [6], gradient perturbation [1], output perturbation [6], and prediction perturbation [27, 28]. For DNN models, we compare input perturbation [20], gradient perturbation, and prediction perturbation because, as of now, these three are widely implemented for DNNs.

**Main Findings:** Our findings suggest that perturbation techniques that offer lower utility loss are more vulnerable to inference attacks. Moreover, for a lower privacy budget, perturbation techniques like objective perturbation and output perturbation result in ML models that classify near-random guessing, i.e., produce extreme utility loss that models fail to classify correctly. For binary classifiers, objective perturbation is a better choice compared to gradient perturbation while for multi-class classifiers, objective perturbation offers the highest privacy/utility trade-off. For multi-class classifiers, gradient perturbation performs well in terms of privacy/utility trade-off. Overall model architectures and datasets, prediction perturbation results in the lowest utility loss but at a cost of privacy leakage. True revealed records have almost a linear relationship with privacy leakage. Overall, we observe that as the privacy leakage increases, a model starts to leak more true records. In a nutshell, our detailed evaluations suggest that, to make informed decisions as to which perturbation mechanism to use, an ML privacy practitioner needs to examine the dynamics between optimization techniques (e.g., convex vs. non-convex), number of classes (e.g., binary vs. multi-class), and privacy budget.

**Comparison with Closely Related Work:** DP-UTIL complements prior work in two major ways. First it enables comprehensive DP utility analysis covering five DP perturbation mechanisms for LR and three DP perturbation methods for DNN. Second, it sheds new light on the utility of DP in the ML pipeline.

**DP-UTIL is more comprehensive:** Table 1 summarizes the coverage comparison of DP-UTIL and closely related work [14, 20, 36]. Compared to [14] which is limited to privacy-utility analysis of gradient perturbation for DP, DP-UTIL covers all the 5 perturbation methods for LR and 3 widely used perturbations for DNN. Hence, it is more comprehensive. In addition, while [14] uses image classification datasets, we evaluate DP-UTIL with datasets from medical and finance domains in addition to a benchmark image dataset. With respect to [20] which studies the privacy guarantee offered by input perturbation against objective, gradient, and output perturbation, in DP-UTIL we extend the analysis with prediction perturbation and extend the evaluation metrics with privacy leakage and number of truly revealed training examples over a wider range of privacy budget than [20]. Additionally, [20] does not offer deeper insights on implications of considering different experimental setups (e.g., binary vs. multi-class models, LR vs. DNN, image data vs. numerical data).

Compared to [36] which covers input and gradient perturbation for DNN and input and output perturbation for Naive-Bayes, DP-UTIL extends the coverage by analyzing three more DP perturbations for LR and one more (prediction perturbation) for DNN.

**DP-UTIL offers new insights:** In [14], the main takeaway is that relaxed DP formulations improve model utility for a given privacy budget, yet the lower DP noise results in additional privacy leakage (hence, the utility does not come for free). Compared to [14], we observe in some cases, a perturbation method results in lower utility over other perturbation techniques, hence costs privacy leakage in exchange. Thus, choosing one perturbation method over another for better utility does not come without paying in privacy leakage. For example, prediction perturbation offers the lowest utility loss for both LR and DNN, hence costs more privacy leakage compared to other perturbation mechanisms.

Compared to [20], where theoretical guarantee for input perturbation seems promising, our experimental findings suggest that input perturbation results in rapid privacy leakage with higher privacy budget and this change is usually triggered at \( \epsilon \geq 1 \).

In [36], their findings suggest that the number of classes of a given dataset is unlikely to influence where the privacy/utility trade-off occurs. Our findings rather suggest that number of classes has implications on privacy/utility trade-off. For objective perturbation, for instance, binary classifiers show an overall better privacy/utility trade-off compared to multi-class classifiers for both DNN and LR models. Our evaluations also suggest overall similar findings for gradient and input perturbation. Another major conclusion in [36] is noise added at a later stage (e.g., output) in the ML pipeline results in lower utility loss. However, our findings show that objective/gradient perturbation overall results in lower utility loss compared to output perturbation.

**Contributions:** We make the following contributions:

- We propose DP-UTIL\(^1\), a holistic utility analysis framework for differential privacy across the machine learning pipeline to understand the impact of different perturbation techniques with respect to a given range of privacy budget (Sections 6.1, 6.2, and 6.3). To that end, we analyze input perturbation, objective

\(^1\)DP-UTIL Code is available at: https://github.com/um-dsp/DP-UTIL
perturbation, gradient perturbation, output perturbation, and prediction perturbation.

- Using membership inference as a case study and privacy leakage as a metric, we comparatively analyze the extent to which machine learning models are protected with state-of-art DP perturbation techniques (Section 6.2).
- We perform an extensive analysis of utility loss and privacy leakage over a range of privacy budget values for two model architectures (Logistic Regression and Deep Neural Network), two naturally privacy-sensitive datasets (finance: LendingClub-Loan dataset [18], healthcare: COVID-19 dataset [19]), and a benchmark image classification dataset (CIFAR-10 dataset [22]).

## 2 BACKGROUND

In this section, we briefly highlight machine learning preliminaries and the definition of differential privacy.

### 2.1 Machine Learning Preliminaries

**Typical ML Training.** In this paper, we focus on supervised machine learning models. Given a set of labeled training samples $X_{train} = (X_i, y_i): i \leq n$, where $X_i$ is a training example and $y_i$ is the corresponding label, the objective of training a ML model $\theta$ is to minimize the expected loss over all $(X_i, y_i): J(\theta) = \frac{1}{n} \sum_{i=1}^{n} l(\theta, X_i, y_i)$. In ML models such as LR and DNNs, the loss minimization problem is typically solved using stochastic gradient descent (SGD) by iteratively updating $\theta$ as:

$$\theta = \theta - \epsilon \cdot \Delta \theta \sum_{i=1}^{n} l(\theta, X_i, y_i)$$

where $\Delta \theta$ is the gradient of the loss with respect to the weights $\theta$; $X$ is a randomly selected set (e.g., a mini-batch) of training examples drawn from $X_{train}$, and $\epsilon$ is the learning rate which controls the magnitude of change on $\theta$.

**Typical ML Testing:** Let $X$ be a $d$-dimensional feature space and $Y$ be a $k$-dimensional output space, with underlying probability distribution $Pr(X, Y)$, where $X$ and $Y$ are random variables for the feature vectors and the classes (labels) of data, respectively. The objective of testing a ML model is to perform the mapping $f_\theta: X \rightarrow Y$. The output of $f_\theta$ is a $k$-dimensional vector and each dimension represents the probability of input belonging to the corresponding class.

### 2.2 Differential Privacy

For two neighboring datasets $D_1$ and $D_2$ which differ by just one data point, let the output space of a randomized mechanism $M$ be $S$. Differential privacy (DP) guarantees that a randomized mechanism $M$ does not enable an observer (adversary) to distinguish whether $M$’s output was based on $D_1$ or $D_2$. Dwork et al. [9] formalize $(\epsilon, \delta)$-DP as follows. A mechanism $M$ preserves $(\epsilon, \delta)$-DP if:

$$Pr[M(D_1) \in S] \leq e^\epsilon \times Pr[M(D_2) \in S] + \delta$$

(2)

where $\epsilon$ is the privacy budget and $\delta$ is the mechanism’s failure probability. When $\delta = 0$, we obtain a strict $\epsilon$-DP formulation of (2), also called pure $\epsilon$-DP. The lower the value of $\epsilon$, the stronger the privacy protection and the higher the utility loss.

To achieve $\epsilon$-DP, Laplace distribution is a common choice to sample noise. For $(\epsilon, \delta)$-DP, one can leverage Gaussian distribution for sampling noise. In both $\epsilon$-DP and $(\epsilon, \delta)$-DP, the sampled noise is correlated with the sensitivity of the mechanism $M$. For two neighboring datasets $D_1$ and $D_2$ differing by one record, the sensitivity $\Delta M$ is the maximum change in the output of $M$ over all possible inputs. Computing $\Delta M$ as the maximum of $||M(D_1) - M(D_2)||$ establishes worst-case upper bound on how much the output of $M$ changes when $D_1$ and $D_2$ are identical except for one record, i.e., $||D_1 - D_2||_1 = 1$.

## 3 PERTURBATION MECHANISMS

In this section, we introduce the five privacy noise mechanisms across the ML pipeline. To guide the forthcoming discussion, we use Algorithm 1 as a high-level ML pipeline skeleton and highlight candidate spots where DP perturbations can be plugged. As noted in [14], the type (convex or non-convex) of the optimization problem dictates the specifics of DP perturbation mechanisms.

### 3.1 Input Perturbation

In a pre-training setting, one natural perturbation alternative is to add noise to individual training samples and produce the perturbed version of $X_{train}$ and train the model on it [7, 11, 20]. For a training data $X_{train}$ with dimension $d$, a typical input perturbation on sample $X_i$ is done as:

$$X_i' = X_i + \text{noise}, \forall X_i \in X_{train}$$

(3)
where $\text{noise} = X'_i + \text{Lap}(\frac{S_i}{\epsilon})$ with $S_i$ as the sensitivity (value range) of the $i^{th}$ feature of $X_i$. To keep the perturbed features within valid value range, clipping is applied using the upper- and lower-bounds of each feature value. Note that, for this perturbation to be practical, no or weak inter-feature dependency is assumed.

When features have diverse representations and unbounded value ranges, estimating sensitivity is not trivial. In domains such as image classification, where features are homogeneously constituted (e.g., pixel intensity values), estimating sensitivity is relatively easy. Another challenge with input perturbation is that post-perturbation, the utility of the trained model needs to be within acceptable utility loss penalty. Given the feature-level fidelity of input perturbation, achieving an acceptable trade-off on model utility is an optimization challenge. Input perturbation has been recently shown [11] to offer both local and model privacy guarantees compared with other perturbation mechanisms.

Differentially private empirical risk minimization (DP-ERM) with input perturbation ensures both local and model privacy. In [11], it is shown that adding noise to input data depends on privacy parameters $\epsilon$, $\delta$, data size $n$, and constants of the loss function. They have also shown that DP-ERM satisfies $(\epsilon, \delta)$-DP where $\epsilon$ is learning rate. For Equation 3, Gaussian noise can be expressed as $N(0, \sigma^2)$, where $\sigma = \sqrt{2d a_1 + \sqrt{2d a_1^2 + 2(1-2a)\epsilon}}$, where $a = \sqrt{T/\epsilon}$ and $d$ is data dimension.

For deep learning, performing input perturbation assumes that the loss function is not strongly convex though it is $\sigma$-Lipschitz and satisfies Polyak-Lojasiewicz condition. For Equation 3, Gaussian noise can be expressed as $N(0, \sigma^2)$, where $\sigma^2 = \frac{C^2 T \log \frac{1}{\delta}}{n}$ for some constant $c$ with $T$ as the total number of iterations [20].

### 3.2 Objective Perturbation

When training a ML model, one of the DP perturbation alternatives is objective perturbation, which works well with convex optimization problems such as ERM [6]. The method proposed in [6] is a two-stage process: (1) add noise to the objective function itself and (2) reveal the minima of the perturbed objective. For convex optimization problems, suppose we consider logistic regression with $l_2$ regularization penalty. The (convex)objective function $J(\theta)$ with objective perturbation is computed as:

$$J(\theta) = \frac{1}{n} \sum_{i=1}^{n} l(\theta, X_i, y_i) + \lambda R(\theta) + \text{noise}$$

where $R(\cdot)$ is the regularization function such as $l_1$ and $l_2$. To add DP guarantee to the model using objective perturbation, noise is added to $J(\theta)$ and then $\theta = \min J(\theta)$ is computed via iterative gradient update using Equation (1). Chaudhuri et al. [6] prove that if $||X_i||_2 \leq 1$ and $y_i \in \{-1, 1\}$ then noise $= \frac{\epsilon}{\sqrt{n}}$ is added to the objective function which has a sensitivity of $\frac{\epsilon}{\sqrt{n}}$.

### 3.3 Gradient Perturbation

The other commonly used perturbation mechanism during training is gradient perturbation. Again, considering logistic regression with $l_2$ regularization penalty, the gradient of the objective function with gradient perturbation is computed as:

$$\nabla J(\theta) = \frac{1}{n} \sum_{i=1}^{n} \frac{-X_i y_i}{1 + \exp X_i \theta} + \lambda \theta + \text{noise}$$

where the gradient has a sensitivity of $\frac{\epsilon}{n}$. In gradient descent, the gradient value is computed for each iteration of the training process, which requires sampling noise with a scale of $\frac{\epsilon}{\epsilon}$ for each iteration of model update [5]. Now integrated into Google’s TensorFlow framework, Abadi et al. [1] proposed DP-SGD to enable deep learning with privacy via gradient perturbation of non-convex optimizers. They use gradient clipping to limit the sensitivity of the training algorithm. Two modifications have been made to realize DP-SGD. First, the sensitivity of the gradient is bounded by clipping the gradient. Second, a randomly sampled noise is added to the clipped gradient. Given training data $(X_1, X_2, ..., X_n)$ and target labels $(y_1, y_2, ..., y_n)$, and gradient $g(X_i)$, to build a $(\epsilon, \delta)$-DP model, this method computes the gradient for a random subset of examples for each batch lot $L$, clip the $l_2$ norm for each gradient, and add random noise of the distribution of $N(\theta, \sigma^2 C^2)$, where $C$ is the clipping threshold and $\sigma$ can be expressed as:

$$\sigma \geq \frac{q}{\sqrt{T \log (1/\delta)}} \frac{1}{\epsilon}$$

where $q = \frac{\epsilon}{n}, T$ is the step size, and $c_2$ is existing constant.

### 3.4 Output Perturbation

In a post-training setting, output perturbation is used to limit the leakage/inference of true model parameters. As shown in [6] for convex optimization problem (e.g., ERM), noise is added to the model parameters $\theta$ as:

$$\theta = \theta + \text{noise}$$

For logistic regression with $l_2$ regularization, output perturbation typically requires sampling noise with sensitivity $\frac{\epsilon}{\sqrt{n}}$.

### 3.5 Prediction Perturbation

The other alternative perturbation mechanism in a post-training setting is prediction perturbation whereby random noise is added to the prediction result before producing the final prediction. For instance, in MemGuard [17], random noise is added to the confidence vector to mask the prediction confidence against membership inference attacks of the likes of Shokri et al. [30]. In PATE [27, 28], noise is appended to the majority vote count of model prediction results of teacher ensembles, each of which are trained on disjoint subsets of the original training data. For an input sample $x$, the final output is chosen via a noisy vote aggregation of the teachers’ prediction results as:

$$f^\theta(x) = \argmax_j v_j(x) + \text{noise}$$

where $v_j(x)$ is the number of teachers that assigned class $j \in \{1, ..., k\}$ to sample $x$ out of the possible $k$ labels. Given that noise is added on top of a vote count, the prediction is perturbed by sampling noise from Laplace distribution as $\text{Lap}(\frac{\epsilon}{\sqrt{n}})$ with sensitivity $= 1$. 


4 DP-UTIL DESIGN OVERVIEW

In this section, we describe DP-UTIL, an extensible framework aimed at conducting a comprehensive privacy/utility trade-off analysis of DP across the ML pipeline.

DP-UTIL is the first framework to combine the thus-far proposed five DP perturbation methods in a single pipeline while enabling multi-metric privacy/utility trade-off analysis. It is designed to easily add new components or update existing ones. Next, we use Figure 1 to describe DP-UTIL and how it can be used and extended by ML privacy practitioners or researchers. In particular, we focus on the three components: DP Perturbation Plugins, Inference Attacks, and Holistic Trade-off Analysis.

4.1 DP Perturbation Plugins

Across the ML pipeline, prior work has proposed five spots where DP could be plugged to enable privacy-preserving ML (hence the term “DP Perturbation Plugins” in Figure 1). Depending on dataset type (e.g., images vs. numeric), loss function (e.g., convex vs. non-convex), gradient computation method, and model architecture, the privacy guarantee offered by each DP perturbation varies. Currently, across two model architectures (LR and DNN) DP-UTIL supports five perturbation plugins: all five for LR, and input, gradient, and prediction perturbation for DNN. While our current design relies on the peer-reviewed implementations of perturbation mechanisms for LR and DNN, users of DP-UTIL can add future implementations with minimal effort.

In terms of support for multiple datasets, currently DP-UTIL supports three datasets from vision (image classification), medical (COVID-19), and finance domains (Loan-Data) for classification tasks. The modular design allows plugging in new datasets and proceed with the rest of the analysis pipeline. The “Pre-processing” component in the ML pipeline block in Figure 1 is meant to offer pre-training data cleaning functionality that a user may customize depending on the dataset at hand.

4.2 Privacy Motivated Inference Attacks

In this component, we assume that multiple inference attacks can be plugged or existing attacks can be replaced with more recent when the state-of-the-art evolves. Among inference attacks are membership inference [16, 25, 30], attribute inference [23], model inversion [10], and model parameter inference/extraction [33]. In its current version, DP-UTIL supports the popular attack of membership inference attack [30], which we introduce next.

In membership inference, an adversary observes a model’s prediction confidence on members of its training set versus non-members to determine whether a target sample is part of the model’s training data. To uniformly analyze the utility of DP across the perturbation mechanisms, we use the membership inference attack introduced by Shokri et al [30]. For this attack, multiple shadow models trained on auxiliary data of the same distribution as the target model’s training data are used to train an attack model. We keep our shadow models’ architecture the same as the target model’s architecture. The attack model is a binary classifier that predicts whether a particular sample is a member of the target model’s training set or not. We use 10 shadow models to train our Random Forest attack model. When we attack the target model, we assume black-box access to each model, i.e., the attacker submits an input sample to a prediction API which returns prediction output of confidence score vector and a label.

Our choice of membership inference attack is informed by its conceptual connection to the primary goal of DP, which is to make the presence/absence of a data sample indistinguishable in the eyes of an adversary. Membership inference essentially aims to achieve the opposite goal: determine, with high confidence, whether a given data sample is present or absent in a training set of a target model. This antagonistic setup between the two makes membership inference a natural fit to showcase DP-UTIL.
4.3 Holistic Trade-off Analysis

As in the other components of DP-UTIL, here we envision a growing list of alternative privacy/utility trade-off analysis metrics used to evaluate model utility (e.g., via accuracy), privacy leakage, actual number of records/attributes inferred, and other relevant metrics such as performance overhead of the analysis scheme and fairness of the model predictions to a sub-population of training data (e.g., minority groups). In its current version, DP-UTIL supports three established metrics: utility loss, privacy leakage, and true revealed data, which together offer a holistic assessment of the utility of DP in limiting privacy motivated attacks such as membership inference.

Utility Loss. Model utility or accuracy is calculated based on percentage rate of correctly predicted labels. We calculate utility loss or label loss as the utility difference between the non-private and the differentially private model. When utility loss is 0, it implies that the private model achieves same utility as non-private model. Formally, utility loss is calculated as: $1 - \frac{\text{accuracy}_{\text{private-model}}}{\text{accuracy}_{\text{non-private-model}}}$

Privacy Leakage. This metric [35] estimates the model’s susceptibility to inference attack. It quantifies the difference between true positive rate and false positive rate of the adversary’s inference attack, and its value lies in the range [0, 1]. A privacy leakage of 0 entails no data leakage induced by the inference attack, while a privacy leakage value of 1 implies absolute inference success. For some of our results, we may observe negative values for privacy leakage. In those cases, the inference attack’s false positive rate is greater than true positive rates, which implies that the attack model is likely to detect more non-members as members.

True Revealed Data. To quantify and observe the impact of non-members falsely inferred as members, we use the true revealed data to estimate the actual number of members whose data is in danger of disclosure when membership inference attack succeeds.

5 DATASETS AND ANALYSIS SETUP

In this section, we describe the setup for our instance of DP-UTIL in Figure 1. Before we describe our setup, to guide our analysis, we provide context on assumptions and scope.

Assumptions and Scope. We assume the correctness of the implementations of the different perturbation mechanisms we analyze. We directly use the original implementations released with the published papers. Following prior work [14], we conduct our analysis for two classes of ML models: logistic regression (convex optimization) and deep neural networks (non-convex optimization). For logistic regression, we analyze input perturbation, objective perturbation, gradient perturbation, output perturbation, and prediction perturbation. For DNNs, we again rely on insights from prior work [14] that noted the suitability of gradient perturbation for non-convex optimization techniques. In addition, we extend prior evaluations of only gradient perturbation mechanisms by introducing input perturbation and prediction perturbation. As of this writing, we have not come across reproducible methods for objective and output perturbation for DNN. When peer-reviewed and reproducible implementations of these missing perturbation methods are made available, the modular design of DP-UTIL allows plugging them into our holistic analysis framework to extend it to a wider range of perturbation types and their variations.

5.1 Datasets

We use three datasets, two of which focus on practical privacy-sensitive domains: healthcare and finance. For financial privacy analysis, we use the LendingClub-Loan dataset [18] from Kaggle, while for medical privacy analysis, we use the COVID-19 dataset [19]. Finally, among benchmark datasets used by prior work, we use the CIFAR-10 [22] dataset. Next, we briefly describe each dataset.

LendingClub-Loan [18]. Lending club is a US peer-to-peer lending company that offers loans in the range $1,000 $40,000. Investors view the loan book on LendingClub website and complete their own analysis to determine the quality of the book based on the information supplied about the borrower, amount of loan, loan grade, and loan purpose. The dataset contains sensitive features about borrowers which include Zip code, employment length, loan amount, home-ownership etc. In the accepted loan data, there is a column name grade, which shows the value from ‘A’ to ‘Z’, where ‘A’ is the highest grade and ‘G’ is the lowest grade. The goal is to build a classifier that given the other features, classifies accepted loan into ‘A’ to ‘G’ grade. The grade is formulated using risk and volatility which adjusts final interest rates. The total number of samples in the dataset is 100K with 166 features. We use 50% of the dataset as training set and the remaining 50% as test set.

COVID-19 [19]. This dataset is COVID-19 related and contains sensitive information about patients as to whether a patient has underlying health conditions such as diabetics, asthma, cardiovascular, or chronic diseases. In addition, among other features, it also includes age, gender, and whether the patient uses tobacco. The task is a binary classification task, i.e., to predict if the patient is COVID-19 positive or negative. The dataset contains 40K samples with 18 features. Similarly, we use 50% of the data as training set and the remaining 50% as test set.

CIFAR-10 [22]. This dataset consists of 60K color images of 10 classes. Each image has a dimension of $32 \times 32 \times 3$. The target classes include 10 object images (e.g., airplane, bus, truck, automobile, dog, bird, frog, deer, horse, ship) that are completely mutually exclusive. We split the 60K samples into equal number of training and test images for our experiments.

5.2 Models and Hyperparameters

Datasets Split. For each dataset, we first split the dataset into two: 50% each. We further split the first 50% into training and testing the model, while the remaining 50% is also split into training and testing for the membership inference attack model. For instance, LendingClub-Loan dataset has total 100K samples. To train our differentially private models, we use 25K samples for training and 25K samples to test the model performance. Similarly, we use the rest of the 50K samples to train and test attack models, 25K each.

Logistic Regression Model. We train the model with $l_2$ regularization, where regularization parameters $\lambda = 10^{-4}$ with 100 epochs. For this setting, we vary our privacy budget $\epsilon$ from $10^{-2}$ to $10^0$. For COVID-19 dataset $\delta = 10^{-4}$ while for CIFAR-10 and LendingClub Loan datasets $\delta = 10^{-5}$. Note that inline with prior work, we keep $\delta$ smaller than the inverse of each training set: 5K, 15K, and 25K. Our learning rate across all datasets is 0.01 and batch size is 250. We use the Adam optimizer.
Deep Neural Network Model. For DNN, our model has an input layer, two hidden layers, and an output layer. The input layer has the dimension of the input, the hidden layers have 64 neurons and ReLU activation function is used. We use softmax activation function for the last layer.

5.3 Perturbations Setup

Next, we describe the specific setup we use for running the five perturbation mechanisms used in our analysis.

Input Perturbation. For LR models, we use techniques from [11]. For DNN, we implement it based on [20]. The method for DNN is distinct compared to LR as we assume DNN does not follow strong convexity considering practical cases.

Objective Perturbation. For LR models, we use the Diffprivlib v0.4 library by IBM [12]. Their Objective perturbation technique is built based on the work of Chaudhury et al. [6] and they integrate their technique with Scikit-Learn library under some restrictions, i.e., their logistic regression function supports only $l_2$ regularization.

Gradient Perturbation. For both LR and DNN models, we use TensorFlow privacy [2] based on the moments accountant theory proposed by Papernot et al. [27]. We calibrated the moments accountant equation by only changing the noise multiplier parameter.

Output Perturbation. With LR models, we add a Gaussian noise after model parameters with the sensitivity of $\frac{2}{n^2}$, where $n$ is number of samples in each dataset, and we use $\lambda = 10^{-4}$.

Prediction Perturbation. We implement PATE proposed by Papernot et al. [27]. More precisely, we divide the LendingClub-Loan dataset, COVID-19 dataset, and CIFAR10 dataset into 40, 30, and 40 number of disjoint datasets, respectively, and train teacher models for each dataset. Each teacher model is trained using similar model architectures discussed earlier for both LR and DNN. To add random noise to the vote count of each label, we sample Laplace noise with privacy budget $\epsilon$ in the range $[10^{-2}, 10^4]$.

6 ANALYSIS RESULTS

We evaluate DP-UTIL by answering the following questions:

- **RQ1**: Among the five perturbation methods in DP-UTIL, is there a particular method that offers minimal utility loss with minimal privacy leakage?
- **RQ2**: What is the impact of number of classes (binary vs. multi-class) on utility/privacy trade-off across perturbation mechanisms?
- **RQ3**: What is the impact of dataset types (image vs. numerical) on utility loss and privacy leakage across perturbation mechanisms?
- **RQ4**: What is the impact of model architecture (shallow learning vs. deep learning) on utility loss and privacy leakage across perturbation mechanisms?

We now present our findings across the three datasets (CIFAR-10, COVID-19, and LendingClub-Loan), two model types (LR and DNN), and five perturbation mechanisms (input, objective, gradient, output, and prediction). We analyze utility loss (Section 6.1), privacy leakage (Section 6.2), and true revealed data (Section 6.3).

6.1 Utility Loss Analysis

Figures 2 and 3 show utility loss (y-axis) for LR and DNN models, respectively, against privacy budget ($x$-axis) over $\epsilon = [10^{-2}, 10^4]$.

**Logistic Regression (LR) Utility Loss.** Figures 2 (a), 2 (b), and 2 (c) show utility loss for LR on CIFAR-10, COVID-19, and LendingClub-Loan, respectively. Among the five perturbation mechanisms, prediction perturbation consistently results in the lowest utility loss for all datasets. In fact, it incurs zero utility loss after $\epsilon = 1$ for all datasets. Next, we examine results for each dataset.

- **CIFAR-10**: The non-private baseline LR model achieves test accuracy of 0.375. From Figure 2 (a), utility loss of output perturbation is the maximum for $\epsilon < 10^3$, and accuracy is in the range $0.01016 \sim 0.1896$. Prediction perturbation achieves utility loss $\sim 0$ from $\epsilon = 1$ onward. Objective perturbation shows utility loss of $0.25 \sim 0.23$ for $\epsilon$ in [0, 10] and the utility loss is $\sim 0.18$ for $\epsilon > 10$, which is lower compared to output perturbation. Besides, for $\epsilon > 0.1$, input and gradient perturbation show lower utility loss for small $\epsilon$ compared to objective, and output perturbation.

- **COVID-19**: The non-private LR model accuracy is 0.661. As can be noticed from Figure 2 (b), utility loss for output perturbation is maximum for $\epsilon < 10$. For instance, at $\epsilon = 1$, output perturbation has utility loss of 0.201 which is larger compared to objective perturbation (0.145), gradient perturbation (0.129), input perturbation ($\sim 0$), and prediction perturbation ($\sim 0$). For $\epsilon \geq 10$, utility loss is negligible $\sim 0.04$, although input, objective, output, and prediction perturbation techniques show slightly lower (almost negligible) utility loss compared to gradient perturbation.

- **LendingClub-Loan**: From Figure 2 (c), for smaller $\epsilon$ value ($\epsilon < 1$), output perturbation produces maximum utility loss, which is 0.665. For all $\epsilon$ in general, we notice that prediction perturbation produces lower utility loss compared to gradient perturbation, input perturbation and objective perturbation. For example, when privacy budget $\epsilon = 1$, utility loss for prediction, gradient, objective, and input perturbation is $\sim 0.36, 0.368$, and 0.54, respectively. Input and gradient perturbation produce lower utility loss, $\leq 0.1$, for $\epsilon \geq 10^3$. Note that when we compare utility loss of objective perturbation with gradient and input perturbation, gradient and input perturbation incur less loss.

**Deep Neural Network Utility Loss.** Figures 3 (a), 3 (b), and 3 (c) show utility loss for CIFAR-10, COVID-19, and LendingClub-Loan for a DNN model on CIFAR-10, COVID-19, and LendingClub-Loan, respectively. Like our observation for LR, prediction perturbation incurs the lowest utility loss not only across the three datasets, but also over the whole range of privacy budget values.

- **CIFAR-10**: The non-private model utility is 0.39. For prediction perturbation, utility loss is $\epsilon \in (0.13, 0.0553)$ where $\epsilon \leq 1$, while for higher $\epsilon$ values, utility loss is $\sim 0$. For gradient and input perturbation, utility loss is higher compared to prediction perturbation. For example, at lower epsilon ($\epsilon < 1$), gradient and input perturbation produce $\sim 0.15$ utility than prediction perturbation.

- **COVID-19**: The non-private model utility is 0.6676. From Figure 3 (b), utility loss in prediction perturbation is $\sim 0$ for $\epsilon > 0.1$. For input perturbation, utility loss is $\sim 0$ for $\epsilon > 1$. For gradient perturbation, utility loss is higher compared to prediction and input perturbation. For $\epsilon = 0.1$ and 1, utility loss for gradient perturbation is 0.0972 and 0.0698 respectively, and for $\epsilon > 1$, utility loss is $\sim 0.03$. 


Observation 1: With regards to RQ1, prediction perturbation achieves the lowest utility loss across all datasets. This is intuitive as prediction perturbation requires less random noise because the noise is added to aggregated results of teachers’ votes. For both LR and DNN, prediction perturbation reaches 0 utility loss at $\varepsilon \geq 0.01$ for most cases. For lower privacy budget values ($\varepsilon \leq 10^2$ or $\varepsilon \leq 10^3$), output perturbation results in the highest utility loss in contrast to gradient or objective perturbation. This result is again intuitive as objective perturbation adds noise to the objective function and afterwards minimizes the loss while output perturbation adds noise to the model parameters. Concerning RQ2, in LR, objective perturbation incurs more utility loss for multi-class classifiers compared to the binary-class classifier for any $\varepsilon \leq [10^{-2}, 10^4]$. Utility loss for gradient perturbation at $\varepsilon < 10$ shows a larger loss for multi-class classifiers compared to binary classifiers. In response to RQ3, for both LR and DNN, gradient perturbation shows lower utility loss from $\varepsilon > 1$ for image data compared to numerical data. Concerning RQ4, for gradient and prediction perturbation, we observe negligible utility loss difference between LR and DNN. Hence, for utility loss, perturbation techniques turn out to be model-independent.

6.2 Privacy Leakage Analysis

Figures 4 and 5 show the privacy leakage plots for LR and DNN models, respectively, over a range of $\varepsilon$ values.

Logistic Regression Privacy Leakage. Figures 4 (a), 4 (b), and 4 (c) show the privacy leakage for LR model for CIFAR-10, COVID-19, and LendingClub-Loan, respectively. In the following, we analyze privacy leakage for each dataset.

CIFAR-10: In the context of Figure 4 (a), the non-private DNN model reaches accuracy of 0.71. Comparatively, as depicted in Figure 3 (c), prediction and input perturbation produce lower utility loss ($\sim 0$) for $\varepsilon > 1$ compared to gradient perturbation. On the contrary, input perturbation shows worst performance compared to other perturbation mechanisms at lower privacy budget. For $\varepsilon \leq 1$, input and gradient perturbation produce $\sim 0.75$ and $\sim 0.4$ higher utility loss, respectively, compared to prediction perturbation.

LendingClub-Loan: The non-private DNN model reaches accuracy of 0.71. Comparatively, as depicted in Figure 3 (c), prediction and input perturbation produce lower utility loss ($\sim 0$) for $\varepsilon > 1$ compared to gradient perturbation. On the contrary, input perturbation shows worst performance compared to other perturbation mechanisms at lower privacy budget. For $\varepsilon \leq 1$, input and gradient perturbation produce $\sim 0.75$ and $\sim 0.4$ higher utility loss, respectively, compared to prediction perturbation.

Observation 1: With regards to RQ1, prediction perturbation achieves the lowest utility loss across all datasets. This is intuitive as prediction perturbation requires less random noise because the noise is added to aggregated results of teachers’ votes. For both LR and DNN, prediction perturbation reaches 0 utility loss at $\varepsilon \geq 0.01$ for most cases. For lower privacy budget values ($\varepsilon \leq 10^2$ or $\varepsilon \leq 10^3$), output perturbation results in the highest utility loss in contrast to gradient or objective perturbation. This result is again intuitive as objective perturbation adds noise to the objective function and afterwards minimizes the loss while output perturbation adds noise to the model parameters. Concerning RQ2, in LR, objective perturbation incurs more utility loss for multi-class classifiers compared to the binary-class classifier for any $\varepsilon \leq [10^{-2}, 10^4]$. Utility loss for gradient perturbation at $\varepsilon < 10$ shows a larger loss for multi-class classifiers compared to binary classifiers. In response to RQ3, for both LR and DNN, gradient perturbation shows lower utility loss from $\varepsilon > 1$ for image data compared to numerical data. Concerning RQ4, for gradient and prediction perturbation, we observe negligible utility loss difference between LR and DNN. Hence, for utility loss, perturbation techniques turn out to be model-independent.
LendingClub-Loan: For $\epsilon \in (0, 1)$, input perturbation shows highest leakage ($0.1 \pm 0.05$) while its leakage is 0 otherwise. Objective perturbation shows comparatively small privacy leakage over all $\epsilon$ values, which is $0.006 \pm 0.003$. For gradient perturbation, privacy leakage value is $0.003 \pm 0.003$ for $\epsilon \leq 1$. For $\epsilon \geq 1$, this value reaches $0.014 \pm 0.007$. For output perturbation, privacy leakage does not follow a pattern for all $\epsilon$ and it varies from $0.007 \pm 0.0052$. For prediction perturbation technique, privacy leakage is 0 for $\epsilon = 0.01$, and 0.00792 for $\epsilon \in (0.1, 10^3)$.

**Deep Neural Network Privacy Leakage.** Figures 5 (a), 5 (b), and 5 (c) show the privacy leakage of a DNN model for CIFAR-10, COVID-19, and LendingClub-Loan, respectively.

**CIFAR-10:** As can be seen from Figure 5 (a), input perturbation shows higher leakage compared to gradient and prediction perturbation. Privacy leakage is $\epsilon (0.01, 0.15)$ for $\epsilon < 10^2$ and $\sim 0.38$ for higher privacy budget values. Gradient perturbation shows comparatively lower privacy leakage ($< 0.01$ for $\epsilon \leq 10$). For $\epsilon > 10$, privacy leakage is incremental. For example, at $\epsilon = 10$, privacy leakage is 0.0251 which reaches 0.0785 at $\epsilon = 10^4$. Also note that, for gradient perturbation, privacy leakage value drops compared to prediction perturbation at $\epsilon \leq 10$, but from $\epsilon > 10$ it is slightly lower for prediction perturbation. At $\epsilon = 1$ privacy leakage is 0.0097 for gradient perturbation while for prediction perturbation this value reaches 0.032. But for higher $\epsilon$ (i.e., $\epsilon = 10^3$), privacy leakage is 0.0602 for gradient perturbation while for prediction perturbation this value reaches 0.0321.

**COVID-19:** From Figure 5 (b), prediction perturbation results in more privacy leakage compared to gradient and input perturbation ($\sim 0$ and $\epsilon (0.03 \pm 0.02)$, respectively). For prediction perturbation, privacy leakage is incremental with respect to increasing value of $\epsilon$. For instance, at $\epsilon = 0.1$, privacy leakage is 0.003239 while for $\epsilon = 10^4$, privacy leakage reaches 0.0431.

**LendingClub-Loan:** As can be seen from Figure 5 (c), gradient perturbation shows privacy leakage $\sim 0$ over all the $\epsilon$ values. For instance, for $\epsilon = 10^3$, privacy leakage is 0.00164. For prediction perturbation, privacy leakage is $\sim 0$ though slightly larger for several $\epsilon$ values. For $\epsilon = 10^3$, privacy leakage of prediction perturbation is $\sim 0.001$ while gradient perturbation reaches $\sim 0$. Input perturbation shows highest privacy leakage, which is $\sim 0.04$ for almost all values of $\epsilon$.

**Observation 2:** In response to RQ1, for LR, objective perturbation shows the lowest privacy leakage compared to other perturbation techniques, which is no leakage for almost all $\epsilon$ choices. On the contrary, for DNN models, gradient perturbation is the best for a privacy practitioner while considering privacy leakage, as leakage seems negligible for different $\epsilon$ choices. Concerning RQ2, LR for the binary classifier shows almost 0 privacy leakage at $\epsilon < 10^3$, which seems promising as it shows almost no privacy leakage with lower utility loss in contrast to multi-class classifiers. In response to RQ4, input perturbation for DNN shows more leakage in contrast to LR for higher privacy budget values (i.e., $\epsilon \sim 10$), which is expected since we use two different input perturbation mechanisms (input perturbation for LR follows more relaxed bounds).

### 6.3 True Revealed Records

Figures 6 and 7 show true revealed record/true positive value with respect to privacy budget $\epsilon$ for CIFAR-10, COVID-19, and LendingClub-Loan, respectively.

**Logistic Regression True Revealed Records.** Figures 6 (a), 6 (b), and 6 (c) show the true revealed records of a LR model over
CIFAR-10, COVID-19, and LendingClub-Loan, respectively. Next, we examine results for each dataset.

**CIFAR-10:** From Figure 6 (a), prediction perturbation shows highest true positive data leakage (10,368) for overall ε values. Output perturbation shows lowest revealed true positive records, surprisingly (0), for ε < 10^3. For ε ≥ 10^3, this value is ~ 3,000. For gradient perturbation, this value reaches 7,000 ~ 8,000 after ε > 1. Objective and input perturbation also reveal lower number of true positive values compared to gradient and prediction perturbation which is ~ 6,000 ± 1000 over the privacy budget range.

**COVID-19:** According to Figure 6 (b), while total number of training data is 5,000 and ε < 10, output perturbation shows lowest values (≤ 750) compared to other perturbation techniques, hence for ε ≥ 10, true positive value increases eventually. For instance, for ε = 100, true revealed value is ~ 2,000 which is 40% of total number of training data. On the other hand, for prediction perturbation, true revealed value is ~ 1850 when ε > 0.01. Objective, gradient, and input perturbation show close numbers of revealed members which is higher than prediction perturbation. For example, gradient perturbation reveals ~ 25,000 true positive values from ε > 0.01.

**LendingClub-Loan:** This result is shown in Figure 7 (c). When total number of training data is 25,000, for ε ∈ (0.01, 10^4), for output perturbation this value increases from 10,848 to 14,101. For instance, for ε = 100, true revealed value is 14,113 which is 56% of total number of training data. On the other hand, for prediction perturbation, it is 14,715 when ε > 0.01, which is the highest. In this context, input perturbation shows better performance compared to output and prediction perturbation. For objective perturbation, true positive samples or true revealed samples are ~ 8,000 for ε ≤ 100, while for ε > 100, total number of true revealed value is ~ 17,000.

For gradient perturbation, true positive value is ~ 6,000 ± 2000 from ε > 0.01, which is the lowest and almost constant across ε values.

**Deep Neural Network True Revealed Records.** Figures 7 (a), 7 (b), and 7 (c) show revealed true members of training dataset or true positive values for a DNN model over CIFAR-10, COVID-19, and LendingClub-Loans, respectively.

**CIFAR-10:** From Figure 7 (a), gradient perturbation reveals lower true positive value than prediction and input perturbation for ε ≤ 10^2, while for ε > 10^2, the numbers are nearly equal for both gradient and prediction perturbation, while input perturbation reveals larger number of members. For example, at ε = 1, gradient perturbation leaks 8,126 true positive values while prediction perturbation and input perturbation reveal ~ 9,200 and ~ 9,300 values, respectively.

**COVID-19:** We observe from Figure 7 (b) that prediction perturbation revealed more true positive values compared to gradient perturbation and input perturbation for all ε. For example, at ε = 100, true revealed value is 3,831 for prediction perturbation while gradient and input perturbation reach 2,148 and 2,552, respectively.

**LendingClub-Loan:** Figure 7 (c) shows that true positive value of input and prediction perturbation is higher than gradient perturbation. For example, at ε = 10^3, prediction and input perturbation reaches true positive value of 14,733 and 17,000 while gradient perturbation reaches 9,034.

**Observation 3:** True revealed records has almost a linear relationship with privacy leakage. Over all the results, we observe that a model starts to leak more true records when the privacy leakage is higher.
6.4 Summary of Observations

From our extensive analysis, we make the following overall observations with respect to RQ1 – RQ4:

- With regards to RQ1, considering both utility and privacy, there is no obvious optimal DP technique that fits well for LR. On the other hand, for deep learning models, gradient perturbation stands out as a choice for practical utility/privacy trade-off. It is also noticeable that, for gradient perturbation, privacy budget \( \epsilon \in [1, 10^2] \) provides reasonable privacy utility trade-offs. We note that our results so far do not point to a reality where one perturbation technique offers better/acceptable utility at no cost (compromise on privacy is inevitable). For instance, prediction perturbation provides better utility compared to other perturbation techniques, but it costs the highest privacy leakage in exchange.

- With respect to RQ2 and analyzing Observations 1 and 2, for a privacy practitioner who wants to work with a binary classifier and LR model, objective perturbation seems a reasonable choice. However, other perturbation techniques (for instance, gradient perturbation), seem better choices for multi-class classifiers compared to objective perturbation as the utility/privacy trade-off is within a tolerable range for gradient perturbation.

- In response to RQ3, analyzing observations 1 and 2, we conclude the overall better performance of gradient perturbation on image datasets compared to numerical datasets.

- Regarding RQ4, we do not observe fluctuations for different perturbation techniques considering different model architectures.

7 RELATED WORK

While previous studies evaluated privacy-accuracy trade-off in terms of privacy budget for different perturbation techniques, they do them in isolation, for example, performing studies for only gradient perturbation. In this context, the absence of a comprehensive picture of privacy/accuracy trade-off for widely adopted perturbation techniques over the ML pipeline is what makes our work broadly orthogonal to prior work [14, 20, 36]. In the following, we highlight the most relevant related works.

Early usages of differential privacy for privacy-preserving ML include ERM [6, 20, 34] and designing differentially private deep learning algorithms [1, 26]. This class of differentially private algorithms add noise at different stages of the ML pipeline via input perturbation [7, 11, 20], objective perturbation [6], gradient perturbation [1], output perturbation [6], and prediction perturbation [17, 27, 28].

In [11], a DP technique for input perturbation is proposed. In this work, they inject random noise into the input data in a manner that satisfies \((\alpha, \delta)\) local differential privacy to the database and \((\epsilon, \delta)\) global differential privacy to the model parameters. In [20], they expanded previous works limited to strong convex loss functions using the Polyak-Łojasiewicz [21] condition to accommodate state-of-the-art models such as DNNs.

Objective perturbation for ERM by Chaudhury et al. [6] assumes strictly convex function and normalized input data for to design privacy preserving LR and Support Vector Machines. In [34], a differentially private ERM is studied for strongly convex loss function with or without non-smooth regularization. Their contribution is that they improve gradient complexity as \(O(n \log n)\). Besides, for high dimensional data, they reduced the gradient complexity to \(O(n^2)\), which is more general and faster than previous works. In a recent work, [15], gradient perturbation on collaborative learning is introduced where multiple data owners add noise to their respective gradient locally after each iteration.

In MemGuard [17], a membership inference defense is proposed which adds a carefully crafted label-preserving noise vector to a prediction confidence vector of a model. The noise is determined via adversarial perturbations. In PATE [27], Scalable-PATE [28], an ensemble of teacher models is trained using disjoint datasets. A separate model, called the student model, is trained-based data labeled with output obtained from a noisy aggregation of the prediction results of teacher ensembles. While PATE [27] uses Laplace noise Scalable PATE [28] uses Gaussian noise. Besides, the Scalable PATE is more selective, i.e., in case of more disagreements among the teachers, the model may simply choose to abstain. In PRICURE [13], a similar strategy to PATE is used to add noise to an aggregation of predictions from multiple models in a collaborative setting to limit the success of membership inference attack.

In [14], gradient perturbation is analyzed on relaxed notions of DP mechanisms for ML focusing on DP with advanced composition [9], zero-concentrated DP [3], and Rényi DP [24]. This work explores utility/privacy trade-off through leakage measurement and concludes that existing DP-ML methods rarely offer acceptable privacy-utility trade-offs for complex models.

Another evaluation of DP over healthcare dataset was studied by Vinith et al. [32]. They studied DP-SGD models in clinical prediction tasks such as X-ray classification and mortality prediction. Their work concludes that DP-SGD loses salient information about minority classes while it preserves data privacy. In [4], they evaluate the effect of differential privacy memorization attack, though does not provide privacy leakage evaluation. In [29], they perform measurement studies on the effectiveness of DP against MIA and this evaluation was limited to DP-SGD [1].

In [36], a similar analysis pipeline as ours is presented. They choose Naive Bayes for input and output perturbation and NN for input and gradient perturbations to evaluate utility/privacy trade-offs on CIFAR, Purchase, and Netflix datasets. The main difference of this work with ours is we evaluated five perturbations for LR and the currently popular three perturbations for DNN, hence DP-UTIL enables more comprehensive analysis.

8 CONCLUSION

We introduced DP-UTIL, a holistic utility analysis framework of differential privacy across the machine learning pipeline. DP-UTIL supports five DP perturbations for logistic regression and three DP perturbations for deep neural networks through a multi-privacy/utility trade-off analysis.

Our evaluations suggest that some perturbation mechanisms outperform others in terms of utility yet cost privacy leakage in exchange. Besides, our results also offer new insights into how DP-techniques compare across different datasets and classifiers. We also report negligible differences in terms of utility and privacy over diverse model architectures. For example, for deep neural networks, gradient perturbation offers a reasonable utility/privacy trade-off.
over other perturbation methods, while for binary classifiers and LR, objective perturbation provides reasonable utility/privacy trade-off compared to multi-class classifiers.

We hope our holistic analysis framework will enable machine learning privacy practitioners to make informed decisions as to which perturbation mechanism to pick based on thorough comparative analysis of the dynamics between optimization techniques in machine learning, number of classes, and privacy budget.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016. ACM, 308–318.

[2] Galen Andrew, Steve Chien, , and Nicolas Papernot. 2020. TensorFlow Privacy. https://github.com/tensorflow/privacy abs/2002.08570 (2020). arXiv:2002.08570

[3] Mark Bun and Thomas Steinke. 2016. Concentrated Differential Privacy: Simplications, Extensions, and Lower Bounds. In Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 9885). 635–658.

[4] Nicholas Carlini, Liang Chiu, Ulfa Erlingson, John Ker, and Dawn Song. 2019. The Secret Shaper: Evaluating and Testing Unintended Memorization in Neural Networks. In 28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019. USENIX Association, 267–284.

[5] Kamalika Chaudhuri and Claire Monteleoni. 2008. Privacy-preserving logistic regression. In Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008. Curran Associates, Inc., 289–296.

[6] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. 2011. Differentially Private Empirical Risk Minimization. J. Mach. Learn. Res. 12 (2011), 1069–1109.

[7] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. 2013. Local Privacy and Statistical Minimax Rates. In 54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October 2013, Berkeley, CA, USA. IEEE Computer Society, 429–438.

[8] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings (Lecture Notes in Computer Science, Vol. 3876). 265–284.

[9] Cynthia Dwork and Aaron Roth. 2014. The Algorithmic Foundations of Differential Privacy. Found. Trends Theor. Comput. Sci. 9, 3–4 (2014), 211–420.

[10] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015. ACM, 587–601.

[11] Kazuto Fukushima, Khang Q. Tran, and Jun Sakuma. 2017. Differentially Private Empirical Risk Minimization with Input Perturbation. In Discovery Science - 20th International Conference, DS 2017, Kyoto, Japan, October 15-17, 2017, Proceedings (Lecture Notes in Computer Science, Vol. 10558). 133–148.

[12] Naoise Holohan, Stefano Braghin, Pól Mac Aonghusa, and Killian Levacher. 2019. Inference in a Multi-Party Setting. In 52nd Annual IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019. IEEE, 691–706.

[13] Ilya Mironov. 2017. Rényi Differential Privacy. In 30th IEEE Computer Security Foundations Symposium, CSF 2017, Santa Barbara, CA, USA, August 21-25, 2017. IEEE Computer Society, 263–275.

[14] Milan Nase, Reza Shokri, and Amir Houmansadr. 2019. Improving Deep Learning with Differential Privacy using Gradient Encoding and Denoising. CoRR abs/2007.11524 (2020).

[15] Nicolas Papernot, Martin Abadi, Ulfa Erlingson, Ian J. Goodfellow, and Kunal Talwar. 2017. Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. In 5th International Conference on Learning Representations, ICLR 2017. OpenReview.net.

[16] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Ulfa Erlingson. 2018. Scalable Private Learning with PATIE. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018. Conference Track Proceedings. OpenReview.net.

[17] Md. Atiqur Rahman, Tafara Rahman, Robert Laganiere, and Noman Mohammed. 2018. Membership Inference Attack against Differentially Private Deep Learning Models. Trans. Data Priv. 11, 1 (2018), 61–79.

[18] Reza Shokri, Marco Stronati, and Vitaly Shmatikov. 2016. Membership Inference Attacks against Machine Learning Models. CoRR abs/1610.05502 (2016).

[19] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. 2017. Machine Learning Models that Remember Too Much. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS, ACM, 587–601.

[20] Vinith M. Suryakumara, Nicolas Papernot, Anna Goldenberg, and Marzeyeh Ghassemi. 2021. Chasing Your Long Tails: Differentially Private Prediction in Health Care Settings. In PACCT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency. 723–734.

[21] Florian Tramer, Fan Zhang, Artjus, Michael K. Reiter, and Thomas Ristenpart. 2016. Stealing Machine Learning Models via Prediction API’s. In 25th USENIX Security Symposium, USENIX Security 16, 601–618.

[22] Di Wang, Minwei Ye, and Junhui Xu. 2017. Differentially Private Empirical Risk Minimization Revisited: Faster and More General. In Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. 7227–2317.

[23] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. In 1st IEEE Computer Security Foundations Symposium, IEEE Computer Society, 268–282.

[24] Benjamin Zi Hao Zhao, Mohamed Ali Kâafar, and Nicolas Courtelle. 2020. Not one but many Tradeoffs: Privacy Vs. Utility in Differentially Private Machine Learning. (2020), 15–26.

[25] Kaggle. 2020. COVID-19 patient pre-condition dataset. https://www.kaggle.com/ tanmoyx/covid19-patient-pre-condition-data set

[26] Yulin Kang, Yong Liu, Ren Niu, Xinyi Tong, Likun Zhang, and Weiping Wang. 2020. Input Perturbation: A New Paradigm between Central and Differential Privacy. CoRR abs/2002.08570 (2020). arXiv:2002.08570

[27] Hamid Kamire, Julie Nutini, and Mark Schmidt. 2016. Linear Convergence of Gradient and Proximal-Gradient Methods Under the Polyak-Łojasiewicz Condition. In Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 9515). Springer, 795–811.

[28] Keras. [n.d.]. The CIFAR-10 dataset. https://keras.io/api/datasets/cifar10/1

[29] Luca Melis, Congzheng Song, Emilio De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting Unintended Feature Leakage in Collaborative Learning. In 2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019. IEEE, 691–706.

[30] Ilya Mironov. 2017. Rényi Differential Privacy. In 30th IEEE Computer Security Foundations Symposium, CSF 2017, Santa Barbara, CA, USA, August 21-25, 2017. IEEE Computer Society, 263–275.

[31] Kaggle. 2020. All Lending Club loan data. https://www.kaggle.com/ wordsforthewise/lending-club

52