On the Pitfalls of Label Differential Privacy

Róbert Busa-Fekete, Andrés Muñoz Medina, Umar Syed, Sergei Vassilvitskii
Google Research, New York, USA
{busarobi, ammedina, usyed, sergeiv}@google.com

Abstract

We study the privacy limitations of label differential privacy, which has emerged as an intermediate trust model between local and central differential privacy, where only the label of each training example is protected (and the features are assumed to be public). We show that the guarantees provided by label DP are significantly weaker than they appear, as an adversary can "un-noise" the perturbed labels. Formally we show that the privacy loss has a close connection with Jeffreys' divergence of the conditional distribution between positive and negative labels, which allows explicit formulation of the trade-off between utility and privacy in this setting. Our results suggest how to select public features that optimize this trade-off. But we still show that there is no free lunch—instances where label differential privacy guarantees are strong are exactly those where a good classifier does not exist. We complement the negative results with a non-parametric estimator for the true privacy loss, and apply our techniques on large-scale benchmark data to demonstrate how to achieve a desired privacy protection.

1 Introduction

Differential Privacy (DP) has emerged as a de facto standard for anonymous data analysis. Depending on the specific trust model, differential privacy can be further divided into two main types: central and local differential privacy [6]. Central differential privacy assumes the existence of a curator of data who can be trusted to faithfully execute differentially private algorithms on the raw data. On the other hand, in local DP the user obscures their information before providing it to an analyst. The analyst is never able to infer any information about any particular user, but can still make inferences about aggregate statistics.

In this work we focus on a particular variant of local differential privacy, known as *label differential privacy*. In this model every user has a *public* feature vector $\boldsymbol{x} \in \mathcal{X}$ and a *private* binary attribute, known as a label, $y \in \{0, 1\}$. This approach has received a lot of recent attention. Practically, it is natural and captures common data release examples, for instance user surveys and private digital advertising [2]. Specifically, in the former, individuals' demographic information is often treated as non-sensitive, as compared to the target of the survey (e.g. income, political preferences, etc.). In the latter, user attributes are often public, transmitted as part of the ad request, but the specific action of a click or a conversion is protected. Empirically, label differential privacy also gives better performance while still providing differential privacy guarantees, and so has seen new methods developed specifically for this model [1, 3, 7].

We show that guarantees offered by label differential privacy are weaker than they appear, as a user experiences privacy loss due to both the public release of the features, as well as the private release of the label. For an illustration consider the following simple example. Let X be a set of points on the real line, and for each $x \in X$, its corresponding label is $+1$ if $x \geq 0$ and -1 otherwise. Even after flipping the labels with some probability (and thus achieving label DP), a classifier trained on the data will almost surely predict the correct label y. This information can then be used by an attacker to update their belief on the true label for any example.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

What the above example shows is that when the features are sufficiently predictive of the label, obscuring the label is not enough, as a classifier can still be trained on such noisy data. We make this connection between the achieved privacy and the potential utility (in terms of classification quality) formal and show how the privacy guarantees depend on the distribution of $\eta(\boldsymbol{x}) = \mathbf{P}(Y=1|\boldsymbol{X}=\boldsymbol{x})$. We then develop a method for estimatin the the privacy leakage by using a non-parametric approach and establish a connection between the privacy leakage and the Jeffreys divergence. Finally, we validate these findings with exhaustive experiments.

2 Privacy in binary classification

We denote the conditional label distribution function by $\eta(\mathbf{x}) = \mathbf{P}(Y=1|\mathbf{X}=\mathbf{x})$ and the marginal of $\mathbf{P}$ over $\mathcal{X}$ by μ. That is $\mu(A) = \mathbf{P}(\mathbf{X} \in A)$. Throughout the rest of the paper, capital letters will denote random variables and realizations of these will be denoted by lower case letters.

Let $\mathcal{D}_n = \{\boldsymbol{x}_i, y_i\}_{i\in[n]}$ be drawn i.i.d. from $\mathbf{P}$, and denote by $\tilde{\mathcal{D}}_n$ to be the privatized view where each label is flipped independently with probability $\pi \in [0, 1/2)$, i.e $\tilde{\mathcal{D}}_n = \{\boldsymbol{x}_i, \tilde{y}_i\}_{i\in[n]}$, where $\tilde{y}_i = Zy_i + (1-Z)(1-y_i)$ and Z is a Bernoulli random variable with parameter π. We now turn our attention to the privacy definitions we will use throughout the paper.

Definition 1. *Let $\epsilon > 0$, we say a randomized mechanism $M\colon \mathcal{Y} \to \mathcal{Y}$ for is label-locally differentially private if for any $y \in \mathcal{Y}$ $e^{-\epsilon} \leq \frac{\mathbf{P}(M(Y)=y|Y=0)}{\mathbf{P}(M(Y)=y|Y=1)} \leq e^{\epsilon}$,where the probability is taken over the randomness of M.*

It is not hard to show that the noisy label mechanism $M(Y) := \widetilde{Y}$ previously described is ϵ label-locally differentially private for $\epsilon = \log\frac{1-\pi}{\pi}$. Local differential privacy is well known to provide some of the strongest privacy guarantees as it is impossible for an adversary to know the true label of a user from the output of the mechanism. More precisely, a well known property of local differential privacy — which follows from a simple application of Bayes rule to Definition 1 — is $\frac{\mathbf{P}(Y=y|\tilde{Y}=y)}{\mathbf{P}(Y=1-y|\tilde{Y}=y)} \leq e^{\epsilon}\frac{\mathbf{P}(Y=y)}{\mathbf{P}(Y=1-y)}$. That is, given a noisy label an adversary can only increase their posterior knowledge about the true value of a user's label by a factor of e^{ϵ}. In other words, one can interpret ϵ as the privacy loss incurred by revealing $\widetilde{Y}$. If the labels are independent from the feature vectors, then the privacy bound of label flipping is exactly as described above. However, with knowledge of the conditional label distribution function η and π, an adversary can improve the quality of label inference which in turns degrades the privacy guarantee given by Definition 1. To formalize this, we introduce the notion of instance-based privacy loss.

Definition 2. *A function $\nu\colon \mathcal{Y} \times \mathcal{Y} \times \mathcal{X} \to \mathbb{R}$ is said to be the instance-based privacy loss of the noisy label mechanism if $\frac{\mathbf{P}(Y=y|\tilde{Y}=\tilde{y},\boldsymbol{X}=\boldsymbol{x})}{\mathbf{P}(Y=1-y|\tilde{Y}=\tilde{y},\boldsymbol{X}=\boldsymbol{x})} = e^{\nu(y,\tilde{y},\boldsymbol{x})}\frac{\mathbf{P}(Y=y)}{\mathbf{P}(Y=1-y)}$*

The privacy loss function $\nu(y,\tilde{y},\boldsymbol{x})$ formalizes the drop in uncertainty about the true label of $\boldsymbol{x}$ when the noisy label and the feature vector are revealed, and it reduces to the no information case when only the prior of the labels is known. Using Bayes' theorem, we can rewrite instance-based privacy loss as

$$\nu(y,\tilde{y},\boldsymbol{x}) = \log\frac{\mathbf{P}(\tilde{Y}=\tilde{y}|Y=y,\boldsymbol{X}=\boldsymbol{x})}{\mathbf{P}(\tilde{Y}=\tilde{y}|Y=1-y,\boldsymbol{X}=\boldsymbol{x})} + \log\frac{\mathbf{P}(Y=y|\boldsymbol{X}=\boldsymbol{x})}{\mathbf{P}(Y=1-y|\boldsymbol{X}=\boldsymbol{x})} - \log\frac{\mathbf{P}(Y=y)}{1-\mathbf{P}(Y=y)} \tag{1}$$

The three terms in (1) have clear semantics. The first reflects the uncertainty introduced by using the noisy label mechanism. The second, the uncertainty reduction that can be obtained by knowing the feature vector $\boldsymbol{x}$. Finally, the third term represents the prior knowledge of an adversary. Observe that a key quantity is the conditional dependence of the labels on the feature vectors. If the uncertainty of a label is small given the feature vector, then the noisy label mechanism provides little privacy protection.In an extreme case, when $\eta(\boldsymbol{x}) \in \{0,1\}$, we lose all privacy guarantees because the true label is revealed by the feature vector. In what follows, we investigate the privacy guarantee that can be achieved in this binary classification setup with label noise. We will devise several approaches to analyze $\nu(Y,\tilde{Y},\boldsymbol{X})$ with respect to the data distribution and noise, with the assumption that the data publisher has unlimited access to data without noise.

3 Worst case privacy guarantees

Revealing the noisy label $\tilde{y}$ and feature vector $\boldsymbol{x}$ reduces the uncertainty of the true label y which is expressed by the privacy parameter $\nu(\boldsymbol{x}, y, y')$ defined in (1). We begin by analyzing the worst case privacy loss of the data which we define as $\nu_{\max} = \max_{y,\tilde{y}\in\{0,1\}} \operatorname{ess\,sup}_{\boldsymbol{x}\in\mathcal{X}} \nu(y,\tilde{y},\boldsymbol{x})$ where

$\operatorname{ess\,sup}$ denotes the essential supremum of a function with respect to the marginal measure μ. By definition of the essential supremum, it is not hard to see that if $\mathbf{P}(\eta(\boldsymbol{X}) \in \{0,1\}) > 0$, then $\nu_{\max} = \infty$. That is, the noisy label mechanism provides no privacy in the worst case. It is therefore important for a data publisher to be able to estimate $\nu_{\max}$. Next we show how to estimate $\nu_{\max}$ given a finite sample of the data under a mild assumption which includes Hölder smoothness of $\eta(\boldsymbol{x})$ which we shall refer to as *measure smoothness assumption*.

We will work in the non-parametric regime using k-nearest neighbor estimator which is a *plug-in estimator*, i.e. it estimates $\eta(\boldsymbol{x})$ by using the conditional empirical distribution using the neighbors of $\boldsymbol{x}$. We will denote the estimate of η by $\hat{\eta}(\boldsymbol{x})$. The motivation for using plug-in estimates is that, under mild assumptions, one can show that the L_1 error of the estimator vanishes, therefore we do not have to deal with approximation error. Given $\boldsymbol{x} \in \mathcal{X}$ we let $\{\tau_{n,q}(\boldsymbol{x})\}_{q\in[n]}$ be an enumeration of $[n]$ such that for each $q \in [n-1]$, $\rho\left(\boldsymbol{x}, \boldsymbol{X}_{\tau_{n,q}(\boldsymbol{x})}\right) \leq \rho\left(\boldsymbol{x}, \boldsymbol{X}_{\tau_{n,q+1}(\boldsymbol{x})}\right)$. In words, given $\boldsymbol{x} \in \mathcal{X}$, $\{\tau_{n,i}(\boldsymbol{x})\}_{i\in[n]}$ sorts $\{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n\}$ in increasing order of ρ-distance to $\boldsymbol{x}$.

The k-nearest neighbor regression estimator $\hat{\eta}_{n,k} : \mathcal{X} \to [0,1]$ is given by $\hat{\eta}_{n,k}(\boldsymbol{x}) := \frac{1}{k} \cdot \sum_{i\in[k]} Y_{\tau^{n,i}(\boldsymbol{x})}$.By using $\hat{\eta}_{n,k}$, we define an estimator for supremum and infinimum of the regression function as $\hat{M}_{n,k}(\eta) = \max_{i\in[n]} \{\hat{\eta}_{n,k}(\boldsymbol{X}_i)\}$ and $\hat{m}_{n,k}(\eta) = \min_{i\in[n]} \{\hat{\eta}_{n,k}(\boldsymbol{X}_i)\}$.

Next, we can give a finite-sample, high probability bound on the supremum and infinimum of the conditional label distribution function. This result is based on Theorem 4.1 of [8] and relies on the pointwise estimation error and the continuity of η. The proofs are deferred to the full version.

Theorem 1. *Suppose that the measure-smoothness assumption holds with parameters λ, C_λ. Given $k \in [n]$ with $k \geq 8\log(2/\delta)$, with probability at least $1-\delta$ over $\mathcal{D} = \{\boldsymbol{X}_i\}_{i\in[n]}$ we have*

$$\left|\hat{M}_{n,k}(\eta) - \sup_{\boldsymbol{x}\in\mathcal{X}} \eta(\boldsymbol{x})\right| < \sqrt{\frac{\log\frac{4n}{\delta}}{2k}} + 2C_\lambda\left(\tfrac{2k}{n}\right)^\lambda \text{ and } \left|\hat{m}_{n,k}(\eta) - \inf_{\boldsymbol{x}\in\mathcal{X}} \eta(\boldsymbol{x})\right| < \sqrt{\frac{\log\frac{4n}{\delta}}{2k}} + 2C_\lambda\left(\tfrac{2k}{n}\right)^\lambda$$

4 Expected classification privacy loss

The previous section was concerned with the worst case privacy leakage. Another natural quantity that might be relevant for a data publisher is the expected privacy leakage with respect to $\mathbf{P}$. The instance-based privacy loss ν defined by (1) can be viewed as a random variable $\nu(Y, \widetilde{Y}, \boldsymbol{X})$.

Definition 3. *We define the expected conditional instance-based privacy loss function $\overline{\nu} \colon \mathcal{X} \to \mathbb{R}$ to be the expected value of $\overline{\nu}(Y, \widetilde{Y}, \boldsymbol{X})$ conditioned on $\boldsymbol{X} = \boldsymbol{x}$, i.e. $\overline{\nu}(\boldsymbol{x}) := \mathbf{E}[\nu(Y, \widetilde{Y}, \boldsymbol{x})]$*

The expected conditional instance-based privacy loss measures expected privacy leakage of the noisy label mechanism for a user with feature vector $\boldsymbol{x}$. Simple calculation yields that the average instance-based privacy loss conditioned on any $\boldsymbol{x} \in \mathcal{X}$ is $\overline{\nu}(\boldsymbol{x}) = (2\eta(\boldsymbol{x}) - 1)\left[\log\frac{\eta(\boldsymbol{x})}{1-\eta(\boldsymbol{x})} - \log\frac{p_+}{1-p_+}\right] + (2\pi - 1)\log\frac{\pi}{1-\pi}$. We can analyze the random variable $\overline{\nu}(\boldsymbol{X})$ to measure the expected privacy leakage across all users. Interestingly, this expectation is closely related to to the Jeffreys divergence [5]: $\mathcal{I} = \mathbf{E}\left[(2\eta(\boldsymbol{X}) - 1)\left(\log\frac{\eta(\boldsymbol{X})}{1-\eta(\boldsymbol{X})}\right)\right]$. Indeed, using the fact that $\mathbf{E}[\eta(\boldsymbol{X})] = p_+$, a straightforward calculation shows that

$$\mathbf{E}[\overline{\nu}(\boldsymbol{X})] = \mathcal{I} + h(\pi) - h(p_+), \tag{2}$$

where $h(z) = (2z-1)\log\frac{z}{1-z}$. The above expression can be readily used to obtain an upper bound on the probability of having a user with privacy loss higher than a given threshold $\tau > 0$. Using Markov's inequality we have: $\mathbf{P}(\overline{\nu}(\boldsymbol{X}) \geq \tau) \leq \frac{\mathcal{I} + h(p_+) - h(\pi)}{\tau}$. However,the Jeffreys divergence may be infinite, making the bound vacuous. We introduce a different way of estimating $J(\tau) := \mathbf{P}(\overline{\nu}(\boldsymbol{X}) > \tau)$ which remains valid when $\mathcal{I}$ is infinite. This quantity has clear semantics: what is the probability that we observe an instance $\boldsymbol{x}$ which has expected instance-based privacy loss that is higher than τ. Our estimator is based on the nearest neighbor estimator of Section 3 and it is defined as

$$\hat{J}_{n,k}(\tau) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{\hat{\eta}_{n,k}(\boldsymbol{x}_i) - 2c(n) > t_H\} + \mathbb{I}\{\hat{\eta}_{n,k}(\boldsymbol{x}_i) + 2c(n) < t_L\} \tag{3}$$

where $c(n) := c(n,k,\delta) = \sqrt{\frac{2\log 6n/\delta}{k}} - C_\lambda\left(\frac{2k}{n}\right)^\lambda$, $t_L = \frac{1}{e^\kappa+1}$ and $t_H = \frac{1}{e^{-\kappa}+1}$. This estimator is motivated by the the fact the $J(\tau)$ can be upper bounded by the marginal measure of certain set of

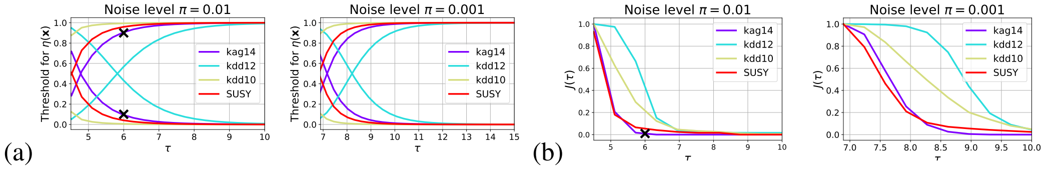

Figure 1: (a) The thresholds for $\eta(\boldsymbol{x})$ for various τ and π to compute $J(\tau)$. Based on Lemma 1, we can upper bound $\mathbf{P}(\nu(\boldsymbol{X}) > \tau) \leq \mu(G_{t_L}) + \mu(G^{t_H})$. We plotted t_H and t_L with respect to τ. (b) High probability upper bound for $J(\tau)$ with $\delta = 0.01$. These graphs show $\hat{J}_{n,k}(\tau)$ defined in (3), computed on the training data.

instances for which the conditional distribution is close to either 0 or 1 as this is presented by the next lemma.

Lemma 1. *For any $\tau > 0$, it holds that $J(\tau) = \mathbf{P}(\nu(\boldsymbol{X}) > \tau) \leq \mu(G_{t_L}) + \mu(G^{t_H})$ with $G_\alpha = \{\boldsymbol{x} \in \mathcal{X} : \eta(\boldsymbol{x}) \leq \alpha\}$ and $G^\alpha = \{\boldsymbol{x} \in \mathcal{X} : \eta(\boldsymbol{x}) \geq 1 - \alpha\}$ where $\kappa = \kappa_{\tau,\pi} := \left(\frac{\sqrt{e}+1}{\sqrt{e}-1}\left(\tau - h(\pi)\right)\right)^{2/3} + \log \frac{p_+}{1-p_+}$.*

Next we compute a high probability error bound for the estimator of $J(\tau)$.

Theorem 2. *Suppose that the measure-smoothness assumption holds with parameters λ, C_λ. Then given $k \in [n]$ with $k \geq 8\log(2/\delta)$ and $\delta > 0$, it holds with probability at least $1 - \delta$ over $\mathcal{D}_n$ that $J(\tau) \leq \hat{J}_n(\tau) + \sqrt{1/2n \log(6/\delta)} + \delta/3$.*

5 Experiments

We present examples that show that the finite-sample estimator can help asses privacy violations and tune the flipping probability π. We use four large scale binary classification datasets, as described in Appendix A. For each dataset we computed an approximate k-nearest neighbor graph under the L_2 distance, using $k = 1000$ for SUSY and $k = 10000$ for the rest.

In the first set of experiments we estimate the worst case privacy loss for various datasets. Table 2 in Appendix shows the estimated extreme values of the regression function and log ratios and their confidence bounds based on Theorem 1. The last two columns of the table compute the worst case privacy loss for these datasets for $\pi = 0.01$ and 0.001. In the absence of feature vectors, these values of π provide ϵ differential privacy for $\epsilon = 4.59$ and $\epsilon = 6.9$ respectively. Our experiments show that the true worst case privacy leakage when releasing feature vectors is approximately twice that for the kag14 dataset and 5-7 times larger for the kdd12 dataset. This increase can be explained by the fact that the regression function is close to 0 or 1 for these datasets. We also note how close the confidence intervals of the minimum and maximum of the regression function are to each other. This validates our believe that the techniques we developed can provide a data owner with the confidence of understanding the risks of releasing noisy labels. Finally, notice that for kdd10 and SUSY, the privacy leakage is infinite. That is due to the fact that there are feature vectors that can predict with certainty the value of a label irrespective of the amount of noise we add to it.

Tuning π based on $J(\tau)$. As we have seen, the extreme values of the regression function can be close to 0 and 1, which implies that in practice, controlling the worst case instance-based privacy loss may not be achievable. On the other hand, if our privacy requirements allow us to release a small subset of data with potentially high privacy loss, while giving stronger guarantees for the rest, then we can tune the noise level by controlling the tail distribution of conditional Jeffreys' divergence, $J(\tau)$, as Theorem 2 suggests.

Figure 1 (a) shows the thresholds t_H and t_L for different τ values, as computed based on Lemma 1 and Figure 1 (b) shows the corresponding upper bounds of $J(\tau)$ according to Theorem 2. To interpret the figures, consider the kag14 dataset with $\pi = 0.01$. If we want to know what is the probability of observing an $\boldsymbol{X}$ for which $P(\nu(\boldsymbol{X}) > 6)$, then $\mu(G_{t_L}) + \mu(G^{t_H})$ needs to be estimated with $t_L = 0.1$ and $t_H = 0.9$, the corresponding points on the graphs are indicated by black crosses. We conclude that even though the worst case privacy guarantee for $\pi = 0.01$ on the SUSY dataset was infinite, for more than 95% of users the true privacy leakage is approximately 6 which is very close to the protection a user would get if features were not available (Figure 1 (b)). This same effect can be observed for the kdd12 and kdd10 datasets. What these experiments show is that by using $J(\tau)$ to measure the privacy leakage, a data owner can still protect the majority of its users while only adding noise to the sensitive labels. In the kdd12 dataset the prior is a good predictor, which explains the low privacy loss.

References

[1] Amos Beimel, Kobbi Nissim, and Uri Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. *CoRR*, abs/1407.2674, 2014.

[2] Charlie Harrison. Github post: Conversion measurement api.

[3] Kamalika Chaudhuri and Daniel J. Hsu. Sample complexity bounds for differentially private learning. In Sham M. Kakade and Ulrike von Luxburg, editors, *COLT 2011 - The 24th Annual Conference on Learning Theory, June 9-11, 2011, Budapest, Hungary*, volume 19 of *JMLR Proceedings*, pages 155–186. JMLR.org, 2011.

[4] Criteo. `http://labs.criteo.com/2014/02/kaggle-display-advertising-challenge-dataset/`, 2014.

[5] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, New York, 1996.

[6] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, August 2014.

[7] Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, and Chiyuan Zhang. On deep learning with label differential privacy. *CoRR*, abs/2102.06062, 2021.

[8] Henry Reeve and Ata Kaban. Fast rates for a kNN classifier robust to unknown asymmetric label noise. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5401–5409, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[9] Niculescu-Mizil A. Ritter S. Gordon G.J. Stamper, J. and K.R. Koedinger. Algebra i 2008-2009. challenge data set from kdd cup 2010 educational data mining challenge. find it at `http://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp`, 2010.

A Main statistics of benchmark datasets

Name	#Train	#Test	#Feat.	p_+	Name	#Train	#Test	#Feat.	p_+
kag14	40M	5.8M	1M	0.256	kdd12	118M	29.9M	54.6M	0.044
kdd10	19.2M	0.7M	29.8M	0.861	SUSY	4.5M	0.5M	18	0.457

Table 1: The main parameters of the benchmark datasets. Here $p_+ = \mathbf{P}(Y = 1)$, i.e. the prior probability of observing a positive label. Kag14 dataset used in Kaggle Display Advertising Challenge and it is released by Criteo [4]. kdd12 dataset is the official dataset of KDD Cup 2012 Track 1and released by Tencent Inc. kdd10 dataset is the official dataset of KDD Cup 2010 [9]. SUSY is taken from UCI repository.

B Worst case privacy loss

Name	$[\hat{m}_L, \hat{m}_H]$		$[\hat{M}_L, \hat{M}_H]$		$\log \frac{1-\hat{m}_L}{\hat{m}_H}$		$\log \frac{\hat{M}_H}{1-\hat{M}_L}$		$\pi = 0.01$		$\pi = 0.001$	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
kag14	[0.01,0.02]	[0.01,0.02]	[0.96,0.98]	[0.95,0.98]	5.28	5.26	3.20	3.08	8.9	8.8	11.2	11.1
kdd12	[0.00,0.00]	[0.00,0.00]	[0.62,0.67]	[0.62,0.67]	27.63	27.63	0.57	0.57	29.1	29.1	31.5	31.5
kdd10	[0.24,0.29]	[0.25,0.29]	[1.00,1.00]	[1.00,1.00]	1.06	1.05	inf	inf	inf	inf	inf	inf
SUSY	[0.01,0.06]	[0.01,0.06]	[0.98,1.00]	[0.98,1.00]	5.17	5.17	inf	inf	inf	inf	inf	inf

Table 2: Extreme values of conditional label distribution function and lower bounds for their log likelihood ratio. We denote the lower and upper confidence bound of the estimate of $m(\eta) = \inf_{\boldsymbol{x}} \eta(\boldsymbol{x})$ by $\hat{m}_L$ and $\hat{m}_H$, respectively. Similarly, for $M(\eta) = \sup_{\boldsymbol{x}} \eta(\boldsymbol{x})$, we denoted the confidence interval by $[\hat{M}_L, \hat{M}_H]$. The worst case privacy loss $\nu_{\max}$ for $\pi \in \{0.01, 0.001\}$ is reported in the last two columns, respectively.