Membership Inference Attacks *via* Adversarial Examples

Anonymous Author(s)
Affiliation
Address
email

Abstract

The raise of machine learning and deep learning led to significant improvement in several domains. This change is supported by both the dramatic rise in computation power and the collection of large datasets. Such massive datasets often include personal data which can represent a threat to privacy. Membership inference attacks are a novel direction of research which aims at recovering training data used by a learning algorithm. In this paper, we develop a mean to measure the leakage of training data leveraging a quantity appearing as a proxy of the total variation of a trained model near its training samples. We extend our work by providing a novel defense mechanism. Our contributions are supported by empirical evidence through convincing numerical experiments.

1 Introduction

At the forefront of artificial intelligence (AI) the benefits and outstanding performance of modern machine learning cannot be missed. Through the lens of supervised learning, computers can now compete with humans regarding vision [28, 30, 38, 61], object recognition [16] or medical image segmentation [39]. Unsupervised and semi supervised learning schemes led to the impressive development of natural language processing [18, 36, 37, 76] or language understanding [12, 24, 81, 84]. Recently, reinforcement learning [70] reached super-human levels in a broad scope of tasks such as games [54, 67], complex biological tasks [40] or even smart device and smart cities [55]. Most of the aforementioned methods leverage deep learning [43] to reach state-of-the-art performance. However, these methods usually need large amount of data to achieve generalization [2, 22, 45, 75] and nowadays, large datasets can be collected at ease in the era of the so-called *big data* [83].

Information disclosure [44] describes the involuntary leakage of information of a provider (*e.g.* a website database, an API or even a deep learning model). Machine learning models used in production do not necessarily provide fairness [53, 78], safety [4, 11] or privacy considerations [35]. Before one claims that leakage of sensitive information is *not a bug but a feature* [26], privacy of data providers for machine learning models has become necessary. It is indeed required according to recent data regulation (see CCP [1], Villani et al. [77] and Regulation [62], van der Burg et al. [73]). It is a growing direction of research [17, 52].

Recent contributions from deep learning [6, 10, 63] and privacy and security communities [21, 47, 64] tend to show that neural networks and deep models are likely to behave as interpolators in high dimension. Considered as benign for generalization [9, 49, 72], overfitting and thus overconfident (incorrect) predictions can be harmful [4, 64] and uncalibrated [60, 79].

Sharing data and model in a private way is a major barrier to adoption of emerging AI technologies [58]. Therefore, ensuring that data-driven technologies respect major privacy constraints and government policies regulations is at the core of modern AI security [25]. In this way, this paper tries to measure the amount of information disclosure or data leakage that a malicious adversary, exploiting the vulnerabilities of a given model, can potentially recover. Defense mechanism preventing or

Submitted to 2022 Trustworthy and Socially Responsible Machine Learning (TSRML 2022). Do not distribute.

minimizing the information leakage are desired tools to ensure confidentiality of training data. We propose a defense mechanism preventing leakage of training data with membership inference attacks based on adversarial strategies.

Contributions. *(i)* We detail a novel membership inference attack (MIA) framework using adversarial examples with general functionals in the output space; *(ii)* We present a particular method which relies on the total variation of the score function; *(iii)* We propose a defense mechanism against MIA leveraging adversarial examples deferred to the supplementary material.

Outline. Section 2 introduces the relevant background for the analysis. Section 3 provides our novel MIA framework using adversarial examples in output space. Section 4 gathers numerical experiments. Section 5 finally concludes our work with a discussion of further avenue. Proofs and relevant material are detailed in the Appendix.

2 Framework

Statistical Learning & Pattern Recognition. Statistical learning [29] is a subset of machine learning that deals with the statistical inference issue of constructing a prediction function from data. In pattern recognition [14], a discrete random label Y valued in $\mathcal{Y}$, with cardinality arbitrarily set to $K > 1$, is to be predicted based upon observing a random vector X valued in an input space $\mathcal{X}$ using the classification rule minimizing a loss function. To perform pattern recognition, the empirical risk minimization (ERM) learning paradigm [74] suggests to solve

$$g^\star \in \arg\min_{g \in \mathcal{G}} \mathbb{E}\left\{\ell(g(X), Y)\right\},$$

where the minimization is set over a class of classifiers $\mathcal{G}$ and the risk $\mathbb{E}\left\{\ell(g(X), Y)\right\}$ is associated to a loss ℓ measuring the error between the prediction $g(X)$ and the label Y. Since the joint distribution $P_{X,Y}$ is unknown in general, one may rely on a training dataset $\mathcal{D}_{\text{train}} = \{(X_i, Y_i)\}_{i=1}^n$ composed on $n \geq 1$ i.i.d copies of (X, Y) to minimize the empirical risk

$$\widehat{g} \in \arg\min_{g \in \mathcal{G}} \left\{ \widehat{R}(g) = \frac{1}{n} \sum_{i=1}^n \ell(g(X_i), Y_i) \right\}.$$

In the case of deep neural network, the empirical risk minimization rewrites as the minimization over a set Θ corresponding to the set of weights associated to a given architecture of neural networks $\mathcal{G}$. In full generality, the minimization should occur over $\mathcal{G} \times \Theta$ although for the sake of simplicity, it is assumed that the deep models' architecture is fixed and well-designed. A classifier $g_\theta \in \mathcal{G} \times \Theta$ gives a score distribution on the label set $\mathcal{P}(\mathcal{Y})$ hopefully close to the probability distribution $\mathbb{P}(Y|X = x)$ [33]. Consistently with our notation, let $\widehat{g}_\theta$ denote a minimizer over Θ of the risk $\widehat{R}(g_\theta)$.

Membership Inference Attack. In order to measure the leakage of information of a model, the authors of [65] introduced the paradigm of Membership Inference Attacks (MIA). The goal of MIA is to assess if a given labeled sample $(x, y) \in \mathcal{X} \times \mathcal{Y}$ belongs to the set of training data $\mathcal{D}_{\text{train}}$ of a classifier $\widehat{g}_\theta$. Without loss of generality, most MIA strategies determine whether a sample was used to train a specific model based on decision rules involving indicators functions. Thus, the MIA strategies may be written as

$$x \mapsto \mathbb{1}\{s(x) \geq \tau\}$$

where $s : \mathcal{X} \mapsto \mathbb{R}$ is some score function and $\tau \in \mathbb{R}$ is a threshold. The score function s may rely on the loss value [65], the gradient norm [66, 82] or on a modified entropy measure [68, 69].

Adversarial Examples. Evasion attacks [7, 8, 13, 51] of classifiers are referred to adversarial attacks when dealing with neural networks [13]. As a tool providing further understanding on underlying estimators, adversarial strategies demonstrate increasing interests in several fields of machine learning such as computer vision [3], reinforcement learning [42] and ranking data [31]. Two types of adversarial strategies can be considered: *targeted attacks* and *untargeted attacks*.

For a given sample $x \in \mathcal{X}$ labeled by a classifier $\widehat{g}_\theta$ as $y \in \mathcal{Y}$, a targeted attacker ψ with target label $\widetilde{y} \neq y$ builds a carefully crafted noise $\varepsilon \in \mathbb{R}^d$ such that $x + \varepsilon$ is predicted as $\widetilde{y}$. On the other hand, regarding untargeted attacks, an attacker ψ will produce a noise $\varepsilon \in \mathbb{R}^d$ for a given sample $x \in \mathcal{X}$ such that: $\arg\max \widehat{g}_\theta(x) \neq \arg\max \widehat{g}_\theta(x + \varepsilon)$. Additional constraints on the noise ε are common to provide the smallest perturbation providing the desired results. In the numerical experiments of Section 4, adversarial examples are built using the `FAB` adversarial strategy [19] from Torchattacks library [41]. Such a scheme aims at providing the smallest perturbation ε to fool the target model $\widehat{g}$

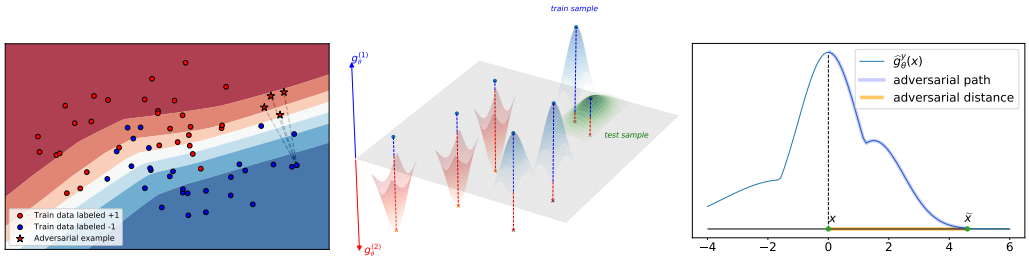
(see Figure 5 in Appendix C.1.)

Learning with noisy labels. The authors of [56] introduced the formal framework of learning with noisy labels. They studied and provided surrogate objective regarding the problem of binary classification in the presence of random classification noise. Instead of seeing the true label, a learner sees a sample with a label that has independently been flipped. To the best of our knowledge, there exists no contribution relying on injecting voluntary noise in the labels to prevent deep models leakage with peaks of confidence on training data samples.

3 MIA Methodology

This section provides insights about the methodology behind MIA with adversarial examples. We first motivate the strategy by showing on synthetic data that trained classifiers present local *peaks of confidence* around the training samples. Such peaks are then exploited to perform MIA with the use of general functionals in the output space.

3.1 Motivations



(a) (b) (c)

Figure 1: (1a) Illustration of adversarial examples (red stars) resulting from train data (blue circle). The background color represents the probability output of a binary classifier. (1b) Illustration of local confidence peaks on the landscape of a binary classifier. The vertical axis in blue (*resp.* red) represents the score for class 1 (*resp.* 2). (1c) Illustrative MIA measures (adversarial path and adversarial distance) with a classifier $\widehat{g}_\theta$ where $\widehat{g}_\theta^y(x)$ denotes $\widehat{g}_\theta$'s score for the class y of a univariate sample x labeled y. $\widetilde{x}$ represents an adversarial example for the sample x regarding $\widehat{g}_\theta$.

Excess of confidence. The starting point of the method is based on the following conjecture: *"trained classifiers are over-confident on their training datasets as they present local peaks of confidence around the training samples"*. More precisely, we argue that the shape of the landscape of a trained binary classifier is made of high hills and valleys around the training samples and flatter curves for new unseen data. While Figure 1a represents the landscape of a binary classifier in a sectional view, its general shape takes the form described in Figure 1b where scores are sharp around samples from the train set and flat around new unseen data.

To highlight such behavior with *peaks of confidence*, we place ourselves in the classification framework with synthethic data. The goal here is to assess the following statement: given a trained classifier, it is enough to have a small noise to modify the distribution of the score function on the training data; whereas greater noise is needed to modify the distribution of the score function on new unseen data. Intuitively, close to a training sample, more noise should be required to fool a classifier because the sample needs to step further away from the local excess of confidence.

Synthetic framework. Consider the multiclass classification task with a dataset $\mathcal{D}$ consisting of $n \geq 1$ observations $X_1, \ldots, X_n \in \mathbb{R}^d$ along with their associated labels $y_1, \ldots, y_n \in \{1, \ldots, K\}$. The data is split into a training set $\mathcal{D}_{\text{train}}$ and a testing set $\mathcal{D}_{\text{test}}$. For ease of notation, denote by $\mathcal{D}_a$ with $a \in \{\text{train}, \text{test}\}$ the different datasets. For each class $k \in \{1, \ldots, K\}$, denote by $\mathcal{D}_a(k) = \{(X, y) \in \mathcal{D} : X \in \mathcal{D}_a, y = k\}$ the elements of $\mathcal{D}_a$ which are labeled k and $n_a(k) = \sum_{(X,y)\in\mathcal{D}_a} \mathbb{1}\{y = k\}$ its cardinal. For $k \in \{1, \ldots, K\}$, denote by $\bar{X}_a(k) = \sum_{X\in\mathcal{D}_a(k)} X/n_a(k)$ the barycenter of class k and consider the average pairwise intra-cluster distance $\delta_a(k)$ defined by $\delta_a(k) = \sum_{i,j=1}^{n_a(k)} \|X_i - X_j\|_2 / \left(2(n_a(k) - 1)n_a(k)\right)$.

Adversarial examples. Each sample X is gradually modified by a random noise $\xi(t)$ and the dis-

tribution of the scores $\rho_X : t \mapsto \|\widehat{g}_\theta(X + \xi(t))\|_\infty$ are to be compared. The random noise is a Gaussian anisotropic noise directed towards the closest barycenter of X that is differently labeled. More precisely, given a sample $X \in \mathcal{D}_a(k)$, consider the barycenter $\bar{X}_a(l)$ which is the closest to X and has a different label, *i.e.*, $l = \arg\min_{j \neq k} \|X - \bar{X}_a(j)\|_2$. The underlying unit direction is $v = (X - \bar{X}_a(l))/\|X - \bar{X}_a(l)\|_2$ and the sample X is modified as

$$X_t = X + \xi(t), \quad \xi(t) = |\varepsilon_t|.v$$

where $\varepsilon_t \sim \mathcal{N}(0, t\delta_a(k))$ controls the level of noise required to disturb the sample. For each value of t, denote by $\mathcal{D}^t_{\text{train}}$ and $\mathcal{D}^t_{\text{test}}$ the corresponding datasets composed of the modified samples X_t.

For different values of t, univariate Kolmogorov-Smirnov (KS) tests are performed to compare the distributions of the scores on the initial data $\rho_X(0) = \|\widehat{g}(X)\|_\infty$ and their noisy counterparts $\rho_X(t) = \|\widehat{g}_\theta(X + \xi(t))\|_\infty$ for both the train and test sets. Furthermore, we also perform bivariate two sample KS tests (see [27]) directly on the data from the same pairs $\left(\mathcal{D}_{\text{train}}, \mathcal{D}^t_{\text{train}}\right)$ and $\left(\mathcal{D}_{\text{test}}, \mathcal{D}^t_{\text{test}}\right)$ to ensure that we do not disturb the data too much. The p-values are averaged over the total number of classes in their respective pairs. It is straightforward that as the value of t increases, the classifier's score on the initial data and the score on the noisy data are getting increasingly different, hence the p-value of the involved tests decreases. Although, in the idea that trained classifiers present local confidence peaks, it is expected that the p-values for the KS test on $(\rho_X(0), \rho_X(t))$ for $\mathcal{D}_{\text{train}}$ decrease faster than the respective ones for $\mathcal{D}_{\text{test}}$.

Results. Figure 2 (Left) depicts the evolution of p-values from the KS tests performed on $\rho_X(0)$ and $\rho_X(t)$ where the number of classes is equal to 4. As expected, as more noise is added, the distributional gap is more remarkable between the original training data $\mathcal{D}_{\text{train}}$ and their noisy counterpart $\mathcal{D}^t_{\text{train}}$ than compared to the test data $\mathcal{D}_{\text{test}}$ and their noisy counterpart $\mathcal{D}^t_{\text{test}}$, as suggested by the lower median p-values. Boxplots on Figure 2 (Right) provide the evolution of the p-values from a bivariate KS test directly computed on the input data to guarantee that the procedure does not disrupt the data excessively. Such a test describes the evolution of the distributional influence of the noise level t. Note that the decay rate of p-values with t is similar between $(\mathcal{D}_{\text{train}}, \mathcal{D}^t_{\text{train}})$ and $(\mathcal{D}_{\text{test}}, \mathcal{D}^t_{\text{test}})$. This behavior along with the left boxplots confirm that the changes in the output space do not depend on the perturbations made in the input data but rather on confidence peaks on the training data.

As a consequence of this first motivational experiment with some *simplistic* noise as adversarial strategy, local excess of confidence of a classifier transcribes the underlying presence of training data. As a result, the output manifold of the classifier can be scrutinized to perform membership inference attacks.

3.2 Arc length & Total Variation

In order to measure what happens on the manifold of predicted values, wee consider curve lengths along adversarial paths. For any real-valued continuous function f, the total variation is a measure of the one-dimensional arc length of the curve with parametric equation $t \mapsto f(t)$.

Definition 1 (Total variation for univariate function $\mathbb{R}$) *Consider a real-valued function f defined on an interval $[a, b] \subset \mathbb{R}$ and denote by Π the set of all partitions $\Pi = \{\pi = (x_0, \ldots, x_n) | a = x_0 < \ldots < x_n = b\}$ of the given interval. The total variation of f on $[a, b]$ is given by $V_{[a,b]}(f) = \sup_{\pi \in \Pi} \sum_{k=1}^n |f(x_k) - f(x_{k-1})|$. If f is differentiable, we have $V_{[a,b]}(f) = \int_a^b |f'(t)| \mathrm{d}t$.*

Output space functionals. Given a classifier $g \in \mathcal{G}$ and a sample $x \in \mathcal{X}$, standard MIA strategies determine whether x is part of the training set or not using $\mathbb{1}\{s(x) \geq \tau\}$ where $s : \mathcal{X} \mapsto \mathbb{R}$ is a score function. This scheme can be extended to account for the information brought in by adversarial examples. For any sample $x \in \mathcal{X}$ with associated adversarial example $\widetilde{x} \in \mathcal{X}$, a general MIA strategy may be written as

$$x \mapsto \mathbb{1}\{\varphi(g, x, \widetilde{x}) \geq \tau\}, \tag{1}$$

where $\varphi : \mathcal{G} \times \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a functional to be specified. When using ℓ_p-norms in the input space as $\varphi(\widehat{g}_\theta, x, \widetilde{x}) = \|\widetilde{x} - x\|_p$, the rule of Equation (1) recovers the framework of [21]. Such distances denoted as *adversarial distances* by [21] are only concerned by the input space $\mathcal{X}$ and do not directly take into account the behavior of the classifier $\widehat{g}_\theta$ and its predicted values. Instead, we advocate for functionals $\varphi : \mathcal{G} \times \mathcal{P}(\mathcal{Y}) \times \mathcal{P}(\mathcal{Y})$ that focus on distances in the output space.

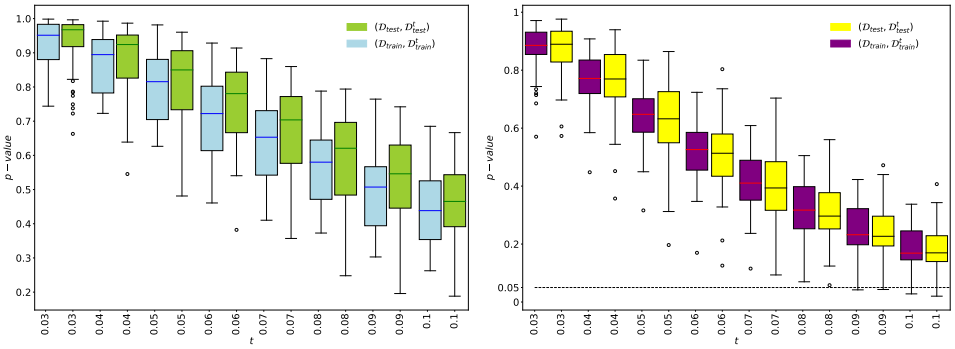


Figure 2: (Left) Boxplot of univariate two sample Kolmogorov-Smirnov tests average p-values between max scores provided by $\widehat{g}_\theta$ computed on the original train data $\mathcal{D}_{\text{train}}$, (*respectively* test data $\mathcal{D}_{\text{test}}$) and the noisy train counterpart $\mathcal{D}^t_{\text{train}}$ (*respectively* noisy test counterpart $\mathcal{D}^t_{\text{test}}$) with varying noise level t. (Right) Boxplot of bivariate two sample Kolmogorov-Smirnov tests average p-values between the original train data $\mathcal{D}_{\text{train}}$ (*respectively* test data $\mathcal{D}_{\text{test}}$) and the noisy train counterpart $\mathcal{D}^t_{\text{train}}$ (*respectively* noisy test counterpart $\mathcal{D}^t_{\text{test}}$) with varying noise level t. The boxplots display the results for 50 different (random) iterations of the experiment. The number of classes is equal to 4.

Motivated by the illustration of Figure 1b, we argue that more attention is needed on the manifold of predicted values $g(\mathcal{X})$ rather than the aforementioned *adversarial distances*. Indeed, the landscape is more likely to capture the high confidence levels associated to the samples used during the training phase. Such behavior is depicted in Figure 1c below where the arc length of the adversarial path better transcribes the local excess of confidence than the adversarial distance.

Consider $x \in \mathcal{X}$ and its associated minimal adversarial counterpart $\widetilde{x} = x + \varepsilon$. The adversarial path is the path connecting the max scores of x and $\widetilde{x}$ along the manifold of predicted values. These paths are characterized by the following family of parameterizations.

Definition 2 (Adversarial paths) *The family of adversarial paths is parameterized by* $(\gamma_x)_{x\in\mathcal{X}}$ *where for all* $x \in \mathcal{X}, \gamma_x : [0,1] \to [0,1]$ *with* $\gamma_x : t \mapsto \|\hat{g}_\theta(x + t\varepsilon)\|_\infty$.

Approximated total variation. The associated arc length of an adversarial path is $\mathscr{L}(\gamma_x) = \int_0^1 |\gamma_x'(t)|\mathrm{d}t$. Using definition 2, the MIA rule we propose is given by

$$x \mapsto \mathbb{1}\{\mathscr{L}(\gamma_x) \geq \tau\}. \tag{2}$$

However, the underlying score function in Equation (2) may be intractable in practice. Hence, one needs to rely on an estimate version of such curvature length. Consider $N \in \mathbb{N} \setminus \{0\}$ and the partition sequence $(t_k)_{k<N}$ given by $0 = t_{-1} = t_0 < \ldots < t_N = 1$ where $t_k = k/N$. We approximate the arc length $\mathscr{L}(\gamma_x)$ by $\mathscr{L}_N(\gamma_x) = \sum_{k=0}^{N-1} |\gamma_x(t_k) - \gamma_x(t_{k-1})|$.

Accordingly, Algorithm 1 provides the pseudocode of our MIA strategy based on the adversarial path and its approximated curvature length.

Algorithm 1 SISYPHOS

Require: Target sample $(x_{\text{test}}, y_{\text{test}})$, target model $\widehat{g}_\theta$, adversarial strategy ψ, $N \in \mathbb{N} \setminus \{0\}$ and $\tau \in \mathbb{R}_+$.

1: Build $\widetilde{x}_{\text{test}} \leftarrow \psi((x_{\text{test}}, y_{\text{test}}), \widehat{g}_\theta)$.
2: Compute $\mathscr{L}_N(\gamma_{x_{\text{test}}})$
3: **Return** $\mathbb{1}\{\mathscr{L}_N(\gamma_{x_{\text{test}}}) \geq \tau\}$.

Note that with Riemann sums, we have $\mathscr{L}_N(\gamma_x) \to \mathscr{L}(\gamma_x)$ as $N \to +\infty$. More precisely we have the following upper bound whose proof is deferred to Appendix A.

Proposition 1 *Assume $\gamma_x \in \mathcal{C}^1([0,1])$ and denote $(I_j)_{j\in J}$ the intervals where γ_x is strictly monotone. If the subdivision $(t_k)_{k=1}^N$ satisfies $\sup_{k=1}^N (t_k - t_{k-1}) \leq \inf_{j\in J} |I_j|$, then we have*

$$0 \leq \mathscr{L}(\gamma_x) - \mathscr{L}_N(\gamma_x) \leq \|\gamma_x'\|_\infty |J| \sup_{k=1}^{N} (t_k - t_{k-1}) .$$

Remark 1 *If $\widehat{g}_\theta$ is Lipschitz then the framework of [21] is covered by Equation* (2).

4 Numerical Experiments

Hereafter we focus on numerical experiments regarding MIA. Further numerical experiments illustrating a defense mechanism are deferred to Appendix D.3.

4.1 Experimental Setting

MIA Baseline. We place ourselves in the experimental framework of [21] and compare their MIA strategy to Sisyphos (Algorithm 1) as their method is essentially the main white box comparable adversarial example-based membership inference attack. Adversarial examples are built with `FAB` adversarial strategy [19]. Experiments conducted in Appendix C and Appendix E illustrate and discuss broader experimental frameworks.
Target Dataset. The real world dataset CIFAR10 is composed of $60,000$ colored images of size 32×32 associated with 10 different labels. These data are divided into two subsets: the training set containing $50,000$ images and the $10,000$ remaining samples belonging to the test set which corresponds to the unseen data.
Target Neural Networks. We consider the Resnet model [34] publicly available in the following repository[1]. For the sake of comparison, we take the same weights as used in [21] to perform MIA.

4.2 Results

Table 1 and Figure 3 provide the comparison's result of Sisyphos and [21]. Based on our MIA experiment, Sisyphos outperforms the method of [21] regarding both AUC and accuracy.

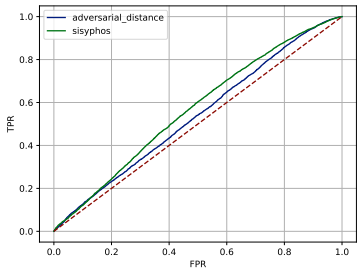


Figure 3: Comparison of ROC curves of MIA strategies relying on adversarial strategies against Resnet on CIFAR10.

MIA Strategy	Resnet	
	AUC	Accuracy
[21]	0.54	0.54
Sisyphos (Ours)	**0.59**	**0.57**

Table 1: Comparison of MIA strategies relying on adversarial examples. We report AUC (%) and Accuracy (%) scores on a balanced evaluation set. $10k$ are uniformly selected from the training set (members) and the whole $10k$ samples from the testing set are selected (non-members). All the data selected is used for evaluation

5 Conclusion and Future Works

We addressed the problem of membership inference attacks based on adversarial examples relying on general functionals. Classifiers' excess of confidence on their training samples is a privacy threat. We introduced a defense mechanism mitigating the risks of attacks with adversarial examples. Future work will explore relevant means to privacy with gradient flows [5, 50] to provide more understanding of learning with privacy. The MIA risks of classifiers splitting the input space with hyperplanes [15, 57] will be further investigated.

[1] https://github.com/bearpaw/pytorch-classification

References

[1] (2018). The california consumer privacy act. 1

[2] Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., and Vijayanarasimhan, S. (2016). Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*. 1

[3] Akhtar, N. and Mian, A. (2018). Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430. 2

[4] Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. (2016). Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*. 1

[5] Arbel, M., Korba, A., Salim, A., and Gretton, A. (2019). Maximum mean discrepancy gradient flow. *Advances in Neural Information Processing Systems*, 32. 6

[6] Balestriero, R., Pesenti, J., and LeCun, Y. (2021). Learning in high dimension always amounts to extrapolation. *arXiv preprint arXiv:2110.09485*. 1

[7] Barreno, M., Nelson, B., Joseph, A. D., and Tygar, J. D. (2010). The security of machine learning. *Machine Learning*, 81(2):121–148. 2

[8] Barreno, M., Nelson, B., Sears, R., Joseph, A. D., and Tygar, J. D. (2006). Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 16–25. 2

[9] Bartlett, P. L., Long, P. M., Lugosi, G., and Tsigler, A. (2020). Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070. 1

[10] Belkin, M., Hsu, D. J., and Mitra, P. (2018). Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate. *Advances in neural information processing systems*, 31. 1

[11] Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery. 1

[12] Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *The journal of machine learning research*, 3:1137–1155. 1

[13] Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. (2013). Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer. 2

[14] Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer. 2

[15] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (2017). *Classification and regression trees*. Routledge. 6

[16] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer. 1

[17] Choquette-Choo, C. A., Tramer, F., Carlini, N., and Papernot, N. (2021). Label-only membership inference attacks. In *International conference on machine learning*, pages 1964–1974. PMLR. 1

[18] Chowdhary, K. (2020). Natural language processing. *Fundamentals of artificial intelligence*, pages 603–649. 1

[19] Croce, F. and Hein, M. (2020a). Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, pages 2196–2205. PMLR. 2, 6, 13

[20] Croce, F. and Hein, M. (2020b). Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2206–2216. PMLR. 13

[21] Del Grosso, G., Jalalzai, H., Pichler, G., Palamidessi, C., and Piantanida, P. (2022). Leveraging adversarial examples to quantify membership information leakage. 1, 4, 6, 13, 14

[22] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee. 1

[23] Deng, L. (2012). The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142. 16

[24] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. 1

[25] Dwork, C. (2011). A firm foundation for private data analysis. *Communications of the ACM*, 54(1):86–95. 1, 15

[26] Espinosa, C. (1979). *APPLE II reference manual*. Apple Computer. 1

[27] Fasano, G. and Franceschini, A. (1987). A multidimensional version of the kolmogorov–smirnov test. *Monthly Notices of the Royal Astronomical Society*, 225(1):155–170. 4

[28] Forsyth, D. and Ponce, J. (2011). *Computer vision: A modern approach.* Prentice hall. 1

[29] Friedman, J., Hastie, T., Tibshirani, R., et al. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York. 2

[30] Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T.-Y., Cubuk, E. D., Le, Q. V., and Zoph, B. (2021). Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2918–2928. 1

[31] Goibert, M., Clémençon, S., Irurozki, E., and Mozharovskyi, P. (2022). Statistical depth functions for ranking distributions: Definitions, statistical learning and applications. *arXiv preprint arXiv:2201.08105*. 2

[32] Goibert, M. and Dohmatob, E. (2019). Adversarial robustness via label-smoothing. *arXiv preprint arXiv:1906.11567*. 15

[33] Harchaoui, W. (2020). *Learning representations using neural networks and optimal transport*. PhD thesis, Université de Paris. 2

[34] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778. 6

[35] Hern, A. (2017). Royal free breached uk data law in 1.6 m patient deal with google's deepmind', guardian, 3 july 2017. 1

[36] Hirschberg, J. and Manning, C. D. (2015). Advances in natural language processing. *Science*, 349(6245):261–266. 1

[37] Jalalzai, H. (2020). *Learning from multivariate extremes: theory and application to natural language processing*. PhD thesis, Institut polytechnique de Paris. 1

[38] Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., and Bengio, Y. (2017). The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 11–19. 1

[39] Jha, D., Smedsrud, P. H., Johansen, D., de Lange, T., Johansen, H. D., Halvorsen, P., and Riegler, M. A. (2021). A comprehensive study on colorectal polyp segmentation with resunet++, conditional random field and test-time augmentation. *IEEE journal of biomedical and health informatics*, 25(6):2029–2040. 1

[40] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. (2021). Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589. 1

[41] Kim, H. (2020). Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950*. 2

[42] Kumar, R., Deleu, T., and Bengio, Y. (2022). Rethinking learning dynamics in rl using adversarial networks. *arXiv preprint arXiv:2201.11783*. 2

[43] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444. 1

[44] Lev, B. (1992). Information disclosure strategy. *California Management Review*, 34(4):9–32. 1

[45] Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60. 1

[46] Li, X., Li, Q., Hu, Z., and Hu, X. (2022). On the privacy effect of data enhancement via the lens of memorization. *arXiv preprint arXiv:2208.08270*. 13

[47] Li, Z. and Zhang, Y. (2021a). Membership leakage in label-only exposures. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 880–895. 1

[48] Li, Z. and Zhang, Y. (2021b). Membership leakage in label-only exposures. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 880–895, New York, NY, USA. Association for Computing Machinery. 15

[49] Li, Z., Zhou, Z.-H., and Gretton, A. (2021). Towards an understanding of benign overfitting in neural networks. *arXiv preprint arXiv:2106.03212*. 1

[50] Liu, Q. (2017). Stein variational gradient descent as gradient flow. *Advances in neural information processing systems*, 30. 6

[51] Lowd, D. and Meek, C. (2005). Adversarial learning. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 641–647. 2

[52] McGraw, D. and Mandl, K. D. (2021). Privacy protections to encourage use of health-relevant digital data in a learning health system. *npj Digital Medicine*, 4(1):1–11. 1

[53] Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35. 1

[54] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533. 1

[55] Mumtaz, S. and Rodriguez, J. (2014). *Smart device to smart device communication*. Springer. 1

[56] Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. (2013). Learning with noisy labels. *Advances in neural information processing systems*, 26. 3

[57] Nelder, J. A. and Wedderburn, R. W. (1972). Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384. 6

[58] Park, J., Samarakoon, S., Bennis, M., and Debbah, M. (2019). Wireless network intelligence at the edge. *Proceedings of the IEEE*, 107(11):2204–2239. 1

[59] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830. 13

[60] Plassier, V., Vono, M., Durmus, A., and Moulines, E. (2021). Dg-lmc: A turn-key and scalable synchronous distributed mcmc algorithm via langevin monte carlo within gibbs. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8577–8587. PMLR. 1

[61] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788. 1

[62] Regulation, G. D. P. (2016). Regulation eu 2016/679 of the european parliament and of the council of 27 april 2016. *Official Journal of the European Union*. 1

[63] Rice, L., Wong, E., and Kolter, Z. (2020). Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pages 8093–8104. PMLR. 1

[64] Sanyal, A., Dokania, P. K., Kanade, V., and Torr, P. H. (2020). How benign is benign overfitting? *arXiv preprint arXiv:2007.04028*. 1

[65] Shokri, R. and Shmatikov, V. (2015). Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321. 2

[66] Shokri, R., Stronati, M., Song, C., and Shmatikov, V. (2017). Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE. 2

[67] Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144. 1

[68] Song, L. and Mittal, P. (2021). Systematic evaluation of privacy risks of machine learning models. In *USENIX Security Symposium*. 2

[69] Song, L., Shokri, R., and Mittal, P. (2019). Privacy risks of securing machine learning models against adversarial examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 241–257. 2

[70] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press. 1

[71] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826. 15

[72] Tsigler, A. and Bartlett, P. L. (2020). Benign overfitting in ridge regression. *arXiv preprint arXiv:2009.14286*. 1

[73] van der Burg, S., Wiseman, L., and Krkeljas, J. (2021). Trust in farm data sharing: reflections on the eu code of conduct for agricultural data sharing. *Ethics and Information Technology*, 23(3):185–198. 1

[74] Vapnik, V. (1992). Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pages 831–838. 2

[75] Varadi, M., Anyango, S., Deshpande, M., Nair, S., Natassia, C., Yordanova, G., Yuan, D., Stroe, O., Wood, G., Laydon, A., et al. (2022). Alphafold protein structure database: Massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1):D439–D444. 1

[76] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30. 1

[77] Villani, C., Bonnet, Y., Berthet, C., Levin, F., Schoenauer, M., Cornut, A. C., and Rondepierre, B. (2018). *Donner un sens à l'intelligence artificielle: pour une stratégie nationale et européenne*. Conseil national du numérique. 1

[78] Vogel, R., Bellet, A., Team, M., Clémençon, S., and LTCI, T. P. (2020). Learning fair scoring functions: Fairness definitions, algorithms and generalization bounds for bipartite ranking. *stat*, 1050:9. 1

[79] Vono, M., Plassier, V., Durmus, A., Dieuleveut, A., and Moulines, E. (2022). Qlsd: Quantised langevin stochastic dynamics for bayesian federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 6459–6500. PMLR. 1

[80] Xie, Q., Dai, Z., Hovy, E., Luong, T., and Le, Q. (2020). Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268. 14

[81] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32. 1

[82] Yeom, S., Giacomelli, I., Fredrikson, M., and Jha, S. (2018). Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282. IEEE. 2

[83] Yi, D., Lei, Z., Liao, S., and Li, S. Z. (2014). Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*. 1

[84] Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. (2022). Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*. 1

SUPPLEMENTARY MATERIAL: MEMBERSHIP INFERENCE ATTACKS VIA ADVERSARIAL EXAMPLES

Appendix A gathers some mathematical proofs. Appendix B provides additional details concerning the synthetic framework presented in Section 3. Appendix C collects a detailed comparison of different MIA strategies, especially regarding the construction and properties of the adversarial noise built in popular adversarial attacks libraries. Appendix D presents our defense mechanism while Appendix D.3 is a numerical illustration of the defense mechanism with the label noise injection. Finally, Appendix E invites to MIA risks based on adversarial examples in a broader scope.

A Proofs

Appendix A gathers proofs defered from the main paper.

A.1 Proof of Proposition 1

PROOF. Since γ_x' is continuous, note that $|J| < \infty$. In addition, for any $j \in J$ consider $K_j = \{k \in \{1, \dots, N\} :]t_{k-1}, t_k[\subset I_j\}$ and define $\bar{K} = \{1, \dots, N\} \setminus (\cup_{j \in J} K_j)$. Since γ_x is supposed in $\mathcal{C}^1([0,1])$, using Taylor's theorem we obtain

$$\gamma_x(t_k) - \gamma_x(t_{k-1}) = \int_{t_{k-1}}^{t_k} \gamma_x'(t)\mathrm{d}t.$$

Thus, we deduce that

$$\begin{aligned}
&\mathscr{L}_N(\gamma_x) \\
&= \sum_{k=1}^{N} |\int_{t_{k-1}}^{t_k} \gamma_x'(t)\mathrm{d}t| \\
&= \sum_{j \in J} \sum_{k \in K_j} |\int_{t_{k-1}}^{t_k} \gamma_x'(t)\mathrm{d}t| + \sum_{k \in \bar{K}} |\int_{t_{k-1}}^{t_k} \gamma_x'(t)\mathrm{d}t| \\
&= \mathscr{L}(\gamma_x) + \sum_{k \in \bar{K}} \left(|\int_{t_{k-1}}^{t_k} \gamma_x'(t)\mathrm{d}t| - \int_{t_{k-1}}^{t_k} |\gamma_x'(t)|\mathrm{d}t \right).
\end{aligned}$$

Hence, we have

$$0 \leq \mathscr{L}(\gamma_x) - \mathscr{L}_N(\gamma_x) \leq \sum_{k \in \bar{K}} \int_{t_{k-1}}^{t_k} |\gamma_x'(t)|\mathrm{d}t. \tag{3}$$

For any $k \in \bar{K}$, either $]t_{k-1}, t_k[\cap (\cup_{j \in J} I_j) = \emptyset$ but it yields $\int_{t_{k-1}}^{t_k} |\gamma_x'| = 0$; or there exists $j \in J$ such that $]t_{k-1}, t_k[\cap I_j \neq \emptyset$ and therefore $t_{k-1} < \sup I_j < t_k$, but using $\sup_{k=1}^{N}(t_k - t_{k-1}) \leq \inf_{j \in J} |I_j|$ ensures the uniqueness of $j \in J$ satisfying $]t_{k-1}, t_k[\cap I_j \neq \emptyset$. Hence, combining (3) with the previous line gives

$$\begin{aligned}
\sum_{k \in \bar{K}} \int_{t_{k-1}}^{t_k} |\gamma_x'(t)|\mathrm{d}t &= \sum_{j \in J} \sum_{t_{k-1} < \sup I_j < t_k} \int_{t_{k-1}}^{t_k} |\gamma_x'(t)|\mathrm{d}t \\
&\leq \|\gamma_x'\|_\infty |J| \sup_{k=1}^{N} (t_k - t_{k-1}),
\end{aligned}$$

which concludes the proof.

A.2 Proof of Remark 1

PROOF. The decision rule of the proposed approach is $D(x) = \mathbb{1}\{\mathscr{L}(\gamma_x) \geq \tau\}$. Using that $\widehat{g}_\theta$ is λ-Lipschitz, it is straightforward to make the link between the input and output space as follows

$$\begin{aligned} D(x) &= \mathbb{1}\{\mathscr{L}(\gamma_x) \geq \tau\} \\ &= \mathbb{1}\{\sup_{\pi \in \Pi} \sum_{k=1}^{n} |\widehat{g}_\theta(x_k) - \widehat{g}_\theta(x_{k-1})| \geq \tau\} \\ &= \mathbb{1}\{\lambda ||\varepsilon|| \geq \tau\} \\ &= \mathbb{1}\{||\varepsilon|| \geq \tau/\lambda\}, \end{aligned}$$

which recovers the decision rule of [21] of the form $x \mapsto \mathbb{1}\{||\varepsilon|| \geq \tau'\}$ with $\tau' = \tau/\lambda$. Note that in practice and in the experiments conducted in Section 4, the values of τ can be optimized through cross validation.

B Further Details on Motivational Experiments

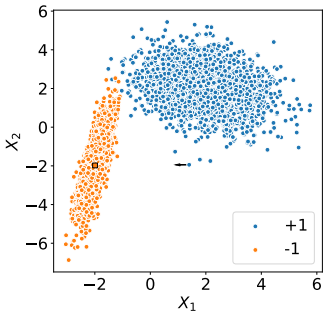


Figure 4: Illustration of the *simplistic* adversarial noise (black arrow) for a bivariate sample labeled $+1$ directed towards the barycenter of data labeled -1 (black square).

In Section 3, the simulated data is generated with `make_classification` function from scikit learn [59], the number of cluster per class is set to 1 as illustrated in Figure 4. The number of sample is set to 2000. The class of classifier $\mathcal{G}$ is the class of multi-layer perceptrons with 2 hidden layers of size 10. The number of iterations is set to 5000 to guarantee convergence of the classifier's weights. The boxplots are obtained over 50 iterations of the experiment.

C Comparing MIA based on Adversarial Strategies: an Ablation Study

In this section, we present the main difference of the MIA strategies mentioned in the paper, namely Sisyphos (Algorithm 1) and [21]. First, we recall that [21]'s MIA strategy corresponds to a specific case of functional regarding the MIA framework introduced in this paper. In a similar way as results depicted in [46], through an ablation study we analyze the influence of three elements. First, we study in Section C.1, the influence of the adversarial strategy. In Section C.2, we focus on the influence of the experimental framework. Finally in Section C.3, the influence of the classifier pre-trained model is discussed.

C.1 Influence of the Adversarial Strategy

Hereafter the difference between two adversarial strategies are depicted. [21] leverages adversarial examples based on `AutoAttack` [20]. In Experiments from Section 4, the chosen adversarial strategy is `FAB` [19]. Let (x_i, y_i) denote an image sample x_i from CIFAR10 with label y_i. Let ε_i be some adversarial noise fooling Resnet neural network on x_i obtained either with `FAB` or `AutoAttack`. Figure 5 reports the results of Resnet's accuracy (%) on the partially noisy samples $x_i + \frac{k}{N}\varepsilon_i$. It appears that the partial noise $\frac{k}{N}\varepsilon_i$ is more likely to fool Resnet classifier with ε_i is obtained with `AutoAttack` than `FAB`. Hence one may prefer to work with `FAB` to obtain minimal noise required to perform some adversarial strategy.

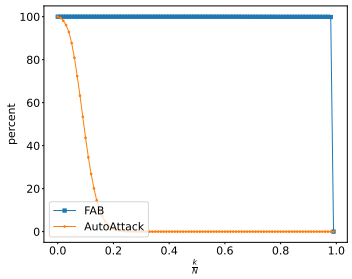


Figure 5: Evolution of the ratio (%) of correctly classified samples $x_i + \frac{k}{N}\varepsilon_i$ as a function of $\frac{k}{N}$ for increasing values of k with $0 \leq k \leq N$. ε_i is built either with `AutoAttack` or `FAB`.

C.2 Influence of the Experimental Framework

To train neural network on images, padding or image rotation are common augmentation schemes [80]. Without loss of generalization, one may assume that some random input image to be classified -after the training phase- is not padded. Figure 6 shows examples of CIFAR10 images without padding and their respective counterpart with padding or rotation.

Figure 6: (Left) Input CIFAR10 images and (Right) corresponding images with padding or rotation.

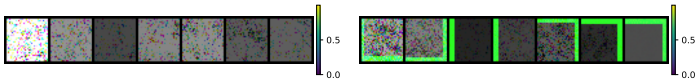


Figure 7: (Left) `FAB` Adversarial noise on samples from Figure 6 (Left). (Right) `FAB` Adversarial noise on samples from Figure 6 (Right).

Figure 7 and Figure 8 represent the adversarial noise obtained by an untargeted adversarial attack. Note that some samples in Figure 7 and Figure 8 appear as minor noise (*i.e.* dark sample) since no noise is required to change the class from the true class, it implies that the classifier did not successfully classify the input image.

Experiments conducted in Section 4 reproduce the experimental framework of [21][2]. Figure 9 is the counterpart of Figure 6 in the framework where padding on targeted training data is removed, `FAB` remains the adversarial strategy for both MIA strategy. Comparing the performance of both MIA strategies on both figures, it appears that padding and other preprocessing on train data may favor the success of membership inference attackers, more especially MIA strategy relying on adversarial examples inducing large change in the input (padded) data. In both settings, Sysiphos (Algorithm 1) outperforms [21].

C.3 Influence of the classifier's pre-trained model

In this section, we assess the influence of Resnet classifier's pretraining and weights on the success of MIA. The PytorchCV public repository[3] provides the weights of Resnet neural networks which are obtained independently from the weights used in Section 4.

Figure 10 is the counterpart of Figure 3 where Resnet's weights are replaced with the weights from PytorchCV. Comparing results from both figures, one can conclude that the target model pretraining has influence on the success of MIAs.

[2]https://github.com/ganeshdg95/Leveraging-Adversarial-Examples-to-Quantify-Membership-Information-Leakage

[3]https://pypi.org/project/pytorchcv/

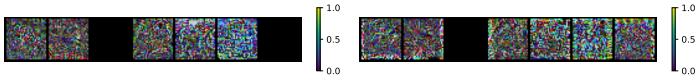


Figure 8: (Left) `AutoAttack` Adversarial noise on samples from Figure 6 (Left). (Right) `AutoAttack` Adversarial noise on samples from Figure 6 (Right).

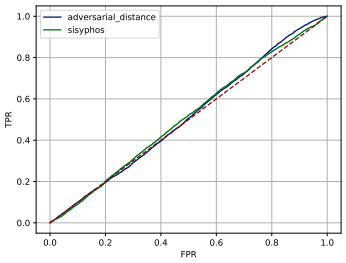


Figure 9: Comparison of ROC curves of MIA scores relying on `FAB` adversarial strategy against Resnet on CIFAR10 without padding in the training set.

D Towards a Defense Mechanism

D.1 Privacy by Design

Most obfuscation mechanisms tend to add noise directly to the samples X_i to reduce the probability of identifying the training data [25]. Alternative means analyze the influence of regularization of neural network [48] or label smoothing [32]. Hereafter we design a mechanism preventing peaks of confidence of the neural network, mitigating MIA related risks. The strategy consists in reducing the vulnerability by adding controlled and voluntary noise to the one-hot encoded labels –counterparts of Y_i– while training the deep model. The core idea is the following: *as the model's generalization improves through the training phase, the model should less require simplistic labels.* Algorithm 2 describes the defense mechanism.

Algorithm 2 Defense Mechanism

Require: Training dataset $\mathcal{D}_{\text{train}} = \{(X_i, Y_i)\}_{i=1}^n$, class of classifier $\mathcal{G}$, batch size $0 < m \leq n$, a non-decreasing non-negative function M valued in $[0, (K-1)/K]$.

Initialize $\widehat{g}_\theta$ with random $\theta \in \Theta$
Set $L_1 = L_2 = 0$.
Set $\widetilde{Y}_i = (\mathbb{1}\{j = Y_i\})_{j=1}^K$.
Set $0 < \zeta < (K-1)/K$.
while θ not converged **or** $L_1 \leq L_2$ **do**
  Sample $\{(X_1, \widetilde{Y}_1), \ldots, (X_m, \widetilde{Y}_m)\}$ from $\mathcal{D}_{\text{train}}$.
  Update $L_1 = L_2$.
  Update $L_2 = 1/m \sum_{i=1}^m \ell(\widetilde{Y}_i, \widehat{g}_\theta(X_i))$.
  Update θ by descending L_2.
  Update $\widetilde{Y}_i = [\mathbb{1}\{j = Y_i\}(1 - \frac{K}{K-1}\zeta) + \frac{\zeta}{K-1}]_{j=1}^K$.
  Update $\zeta = M(\zeta)$.
end while
Return $\widehat{g}_\theta$.

Contrary to [], the defense method described above do not change label from sample X_i with some randomization to another label: it provides a mixture of labels in a similar way as label smoothing [71].

Remark 2 (Random noise injection) *The label noise may be applied on a random subset of a batch instead of the whole batch. In this way, Appendix D.2 provides an alternative of Algorithm 2 with a randomized noise injection.*

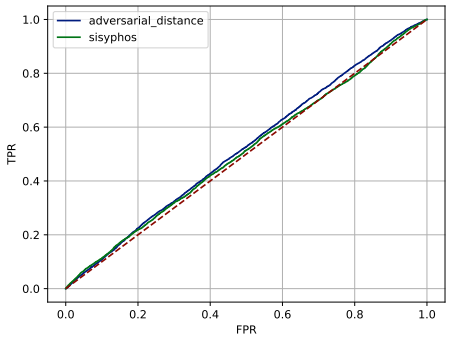


Figure 10: Comparison of ROC curves of MIA scores relying on FAB adversarial strategy against Resnet Pytorch.

D.2 Alternative Defense Mechanism

In Section D.1, we introduce the defense mechanism relying on noisy labels to build a classifier $\widehat{g}_\theta$ hopefully robust to MIA based on adversarial examples. Algorithm 3 provides an alternative version of Algorithm 2 where the voluntary label noise is randomly injected in a subset of the batches of while training. The experimental influence of the randomization of Algorithm 2 is depicted in Section D.3.

Algorithm 3 Randomized SISYPHOS

Require: Training dataset $\mathcal{D}_{\text{train}} = \{(X_i, Y_i)\}_{i=1}^n$, class of classifier $\mathcal{G}$, batch size $0 < m \leq n$, a non-decreasing non-negative function M valued in $[0, (K-1)/K]$, noise injection probability $p \in]0, 1[$.

Initialize $\widehat{g}_\theta$ with random $\theta \in \Theta$.
Set $L_1 = L_2 = 0$.
Set $\widetilde{Y}_i = (\mathbb{1}\{j = Y_i\})_{j=1}^K$.
Set $0 < \zeta < (K-1)/K$.
while θ not converged **or** $L_1 \leq L_2$ **do**
  Sample $\{(X_1, \widetilde{Y}_1), \ldots, (X_m, \widetilde{Y}_m)\}$ from $\mathcal{D}_{\text{train}}$.
  Update $L_1 = L_2$.
  Update $L_2 = 1/m \sum_{i=1}^m \ell(\widetilde{Y}_i, \widehat{g}_\theta(X_i))$.
  Update θ by descending L_2.
  for i **in** $\{1, \ldots, m\}$ **do**
    With probability p update $\widetilde{Y}_i$,
    $$\widetilde{Y}_i = \left(\mathbb{1}\{j = Y_i\}(1 - \tfrac{K}{K-1}\zeta) + \tfrac{\zeta}{K-1}\right)_{j=1}^K.$$
  end for
  Update $\zeta = M(\zeta)$.
end while
Return $\widehat{g}_\theta$.

D.3 Numerical Experiments regarding our Defense Mechanism

In this section, we present numerical experiments run to illustrate the defense mechanism depicted in Section D and its alternative algorithm from Section D.2.

Dataset. We work with MNIST dataset [23]: a database of handwritten digits with a training set of 60,000 examples and a test set of 10,000 examples. The digits have been size-normalized and centered in a fixed-size image. The original black and white (bilevel) images from NIST were size normalized to fit in a 20x20 pixel box while preserving their aspect ratio. The resulting images contain grey levels as a result of the anti-aliasing technique used by the normalization algorithm. The images were centered in a 28x28 image by computing the center of mass of the pixels, and translating

the image so as to position this point at the center of the 28x28 field. Each training and test example is assigned to the corresponding handwritten digit between 0 and 9.

Experimental framework. The target model is a multi-layer perceptron and the defense mechanisms are Algorithm 2 and Algorithm 3. For both defense mechanisms, the label noise is constant through training (*i.e.* for any $\zeta, M(\zeta) = \zeta$) similarly to label smoothing. We train a multi-layer perceptron with one layer containing 100 neurons.

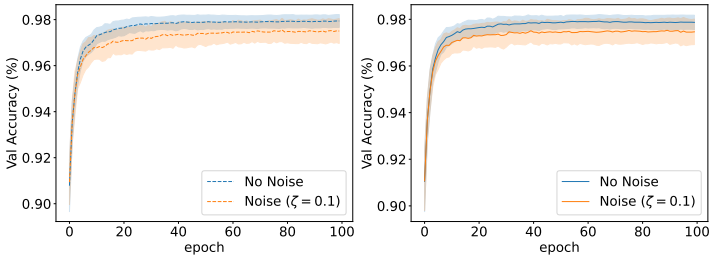


Figure 11: Comparison of $\widehat{g}_\theta$'s accuracies (%) on the validation set during the training phase of $\widehat{g}_\theta$ with (Left) Algorithm 2 and (Right) Algorithm 3 with $p = 0.5$. The amount of label noise for both algorithms is set to $\zeta = 0.1$.

Results. Figure 11 provides the evolution of $\widehat{g}_\theta$'s accuracy on the validation set when label noise is injected (with noise level $\zeta = 0.1$). One can observe similar accuracy between the two settings considered. Figure 12 represents the loss values of $\widehat{g}_\theta$ for the validation and train sets through training with noisy labels following Algorithm 2 or Algorithm 3.

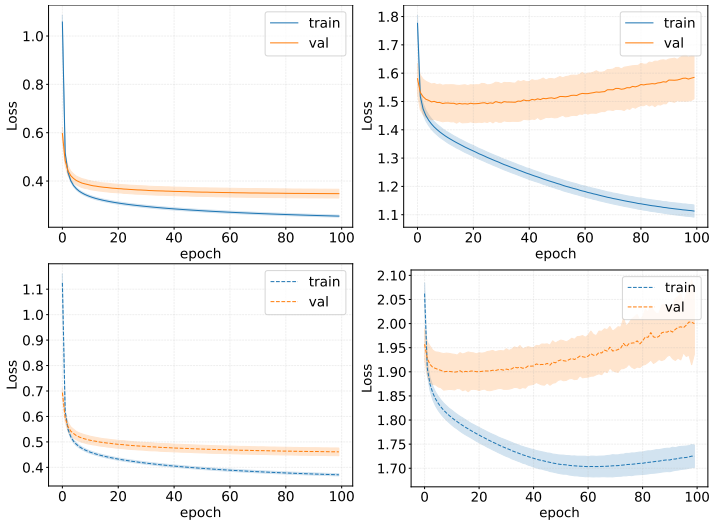


Figure 12: Comparison of $\widehat{g}_\theta$'s loss on the train and validation sets during the training phase of $\widehat{g}_\theta$. The top row represents Algorithm 2 while the bottom row gathers results from Algorithm 3 with $p = 0.5$. The left column corresponds to noise level $\zeta = 0.1$ and the right column when the noise level $\zeta = 0.9$.

Interestingly, one can observe in Figure 12 (Bottom Right) that when large noise ($\zeta = 0.9$) is added to the training labels, two antagonist forces seem to be present: first, the loss minimization focuses on the learning task (*i.e.* predicting the correct class for each sample with the best accuracy possible) and then starts overfitting to the voluntary injected noise in the remaining labels which thus decreases the maximum score granted to the predicted label. In this way, such noise injection could be relevant to better assess early-stopping while training neural networks in addition to some improved robustness to MIAs.

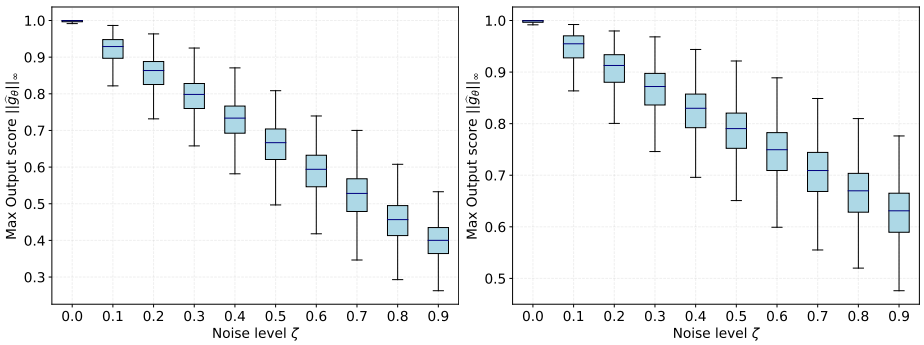


Figure 13: Boxplots of the excess of confidence of $\widehat{g}_\theta$ for varying noise levels.

Figure 13 describes the evolution of the excess of confidence of $\widehat{g}_\theta$ with varying noise levels for (Left) Algorithm 2 and (Right) Algorithm 3. We can see that as more noise is injected in the labels, the excess confidence of $\widehat{g}_\theta$ decreases, thus being more robust to MIAs based on adversarial examples.

E Broader MIA Experiments

The MIA framework in this paper targets to predict if some sample $(X_i, Y_i) \in \mathcal{D}_{\text{test}}$ belongs to from the training set $\mathcal{D}_{\text{train}}$, under the assumption that $\mathcal{D}_{\text{test}} \cap \mathcal{D}_{\text{train}} \neq \emptyset$. In practice, access to the exact samples (X_i, Y_i) is unlikely. In this section, we want to assess if classifier's excess of confidence can still be noticed when one has solely access to $(\widetilde{X}_i, Y_i)$ with $\widetilde{X}_i$ remaining close to X_i. The experiment detailed in this section suggests that the adversarial attack designed in this paper may be generalized to more complex and real world settings.

Figure 14: Samples from Olivetti faces dataset.

Dataset. The Olivetti dataset contains a set of face images taken between April 1992 and April 1994 at AT&T Laboratories Cambridge. There are ten different images $\{X_i^j\}_{i=1}^{10}$ of each distinct subject j among the 40 subjects. The label Y_i associated to a sample X_i consists in assessing if the person in X_i wears glasses (*i.e.* $Y_i = 1$ if and only if the input image X_i contains glasses). For some subjects, the images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). Figure 14 illustrates samples X_1^j, X_2^j (Left, Center) of subject j and $X_1^{j'}$ (Right) of subject j'. Subject j wears no glasses in both presented images while subject j' wears glasses in the presented image.

Experimental framework. Let $\mathcal{D}_{\text{train}} = \{(X_i^j, Y_i)\}_{i<10}^{j\leq 39}$ denote the training set used to build a classifier $\widehat{g}_\theta$. Let $\mathcal{D}_{\text{test}} = \{(X_i^j, Y_i)\}_{i=10}^{j\leq 39} \cup \{(X_i, Y_i)\}_{i\leq 10}^{j=40}$ contain the remaining samples. By construction, there is no sample X_i belonging to the intersection between $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ although some subjects belong to both $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$. $\mathcal{D}_{\text{test}}$ contains 10 images of subject 40 unseen in the training set. We compare the maximum scores of $\widehat{g}_\theta$ for new samples of subjects seen in the training data and new samples of subject unseen in the training phase.

Results. Boxplots in Figure 15 report the max score of $\widehat{g}_\theta$ in two subsets of $\mathcal{D}_{\text{test}}$. The mean of the scores is denoted in orange. The median of the scores is denoted in dashed green. Figure 15 essentially suggests that exploiting the excess of confidence of a classifier to retrieve training data is possible even only with access to modified inputs of the training set. The classifier $\widehat{g}_\theta$ shows larger confidence on its prediction when subjects have already been encountered in the training phase. Therefore, one may generalize MIA strategies with adversarial examples relying on functional in the output space of $\widehat{g}_\theta$ to real world settings.

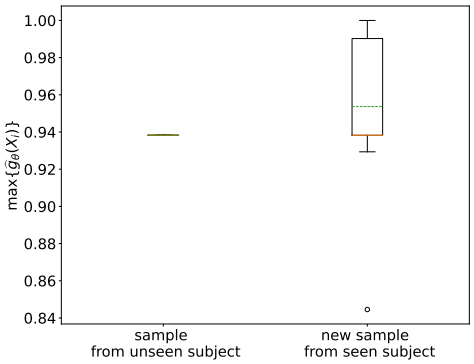


Figure 15: (Left) Boxplots of the max score $\widehat{g}_\theta$ on samples X_i^{40}s from subject 40 from the test set $\mathcal{D}_{\text{test}}$. (Right) Boxplots of the max score $\widehat{g}_\theta$ on the remainder of samples in the test set $\mathcal{D}_{\text{test}}$ with subjects seen in $\mathcal{D}_{\text{train}}$.