Vicious Classifiers: Assessing Inference-time Data Reconstruction Risk in Edge Computing

Mohammad Malekzadeh[1]
mohammad.malekzadeh@nokia.com
Deniz Gündüz[2]
d.gunduz@imperial.ac.uk

[1] Nokia Bell Labs
Cambridge, UK

[2] Imperial College London
London, UK

Abstract

Privacy-preserving *inference*, in edge computing paradigms, encourages the users of machine-learning services to locally run a model on their *private input*, and only share the *model's outputs* for a *target task* with the server. We study how a *vicious* server can reconstruct the input data by observing only the model's outputs, while keeping the target accuracy very close to that of a *honest* server: by jointly training a *target* model (to run at users' side) and an *attack* model for data reconstruction (to secretly use at server's side). We present a new measure to assess the *inference-time reconstruction risk*. Evaluations on six benchmark datasets show the model's input can be approximately reconstructed from the outputs of a *single* inference. We propose a primary defense mechanism to distinguish *vicious* versus *honest* classifiers at inference time. By studying such a risk associated with emerging ML services, our work has implications for enhancing privacy in edge computing. We discuss open challenges and directions for future studies and release our code as a benchmark for the community at github.com/mmalekzadeh/vicious-classifiers.

1 Introduction

Emerging machine learning (ML) services build profiles of their *users* by collecting their *personal data*. Users might share some specific data with a service provider in exchange for some *target* utility. Health monitoring, wellness recommendations, dynamic pricing, or personalized content usually attract users to share their data. If the users are aware of the type of data collected about them, and explicitly confirm their consent, such data collection and profiling is usually considered legitimate [11]. However, the challenge is to ensure that the data collected by a *server* will only be used to deliver the target service they offer to their users [18]. Such data might be used to make other private inferences about the user's personality or identity, which are considered data privacy attacks.

To preserve privacy, current techniques are on-device [34, 45] or encrypted [4, 13] computations that hide inputs, as well as all the intermediate representations computed by the *model*, and only release the *outputs* to the server. Since such edge inferences for a target task might not seem sensitive to users' privacy, the model's outputs are released to the server in their raw form; as the server needs these outputs to perform their ultimate analyses and satisfy the services promised to the users. In various situations, minimal communication between users and servers is crucial, for example, in tasks such as age or identity verification. The

server requires the model's outputs to grant permission for the user to proceed with subsequent actions like account creation or payment. *We argue that such a paradigm of running ML models at the edge and only sharing the outputs with a service provider does not guarantee a meaningful privacy protection for edge users.*

As shown in Figure 1, we consider a common scenario of edge or encrypted inference, in which a *user* owns private data $\mathbf{x}$, and a semi-trusted *server* owns an N-output ML classifier $\mathcal{F}$. We put no constraint on the user's access to $\mathcal{F}$; *e.g.* users can have a complete white-box view of $\mathcal{F}$. We assume that the server only observes the model's output $\hat{\mathbf{y}} = \mathcal{F}(\mathbf{x})$ (*a.k.a. logits*), which aims to help in predicting the target information $\mathbf{y}$. Our main assumption is that $\hat{\mathbf{y}} \in \mathbb{R}^N$ is a real-valued vector of dimension N, where each $\hat{\mathbf{y}}_n \in \mathbb{R}$, for all $n \in \{1,2,\ldots,N\}$, is the *logit score* for the corresponding class or attribute $\mathbf{y}_n$. There are several reasons that a server might ask for observing the real-valued outputs $\hat{\mathbf{y}}$ to reach the ultimate decision at the server's side; compared to only observing the argmax($\hat{\mathbf{y}}$) or softmax($\hat{\mathbf{y}}$). For example, the logit scores allow the server to perform top-K predictions, to measure the uncertainty in the estimation, or to figure out adversarial or out-of-distribution samples [26].

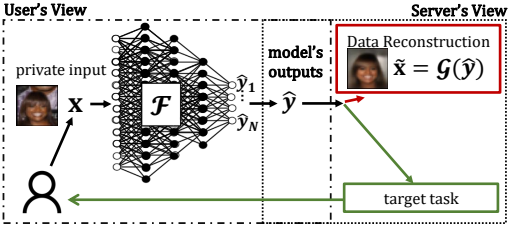

Figure 1: Processing user's input $\mathbf{x}$, the server receives only the output $\hat{\mathbf{y}} = \mathcal{F}(\mathbf{x})$. We show that for any model $\mathcal{F}$, the server can train an attack model $\mathcal{G}$ to secretly reconstruct the input from observed output, while providing the target service to the user with high accuracy.

Here are the **contributions** of our paper:

(1) Over-parameterized deep neural networks (DNNs) can be trained to efficiently encode additional information about their input data into the model's outputs which are supposed to reveal nothing more than a specific target class or attribute. We propose jointly training a multi-task model $\mathcal{F}$ (*i.e.,* a *vicious classifiers*) as a classifier of target attributes as well as a decoder model $\mathcal{G}$ (*i.e.,* an *attack model*) for reconstructing the input data from the shared outputs. The trained $\mathcal{F}$ can be efficiently useful for the target task, and also secretly encode private information that allows reconstructing the user's input data at inference time. Evaluations on MNIST, FMNIST, CIFAR10, CIFAR100, TinyImageNet, and CelebA datasets show input data can be approximately reconstructed from just the outputs of a single inference. To assess the success of a malicious server, we consider two settings, where users share either the *logits* outputs or the *softmax* outputs. For the same model, in the softmax setting, it is harder to establish a good trade-off between the accuracy of target task and the quality of reconstructed data, particularly, when the number of classes or attributes is less than 10.

(2) To measure the risk of data reconstruction, previous works [12, 39, 46] mostly use mean-squared error (MSE), peak signal-to-noise ratio (PSNR), or structural similarity index measure (SSIM). Euclidean distance-based measures assume that features are uncorrelated, which is not true for real-world data, such as images where pixels often have high correlation. We believe *the risk of a reconstruction attack* depends not only on the similarity of the reconstruction to the original data, but also on the likelihood of that sample data. To this end, we propose a new measure of reconstruction success rate based on the Mahalanobis distance, which considers the covariance matrix of the data. Our proposed measure, called *reconstruction risk*, also offers a probabilistic view on data reconstruction attack and thus, offers a principled way to evaluate the success of an attack across models and datasets.

(3) Distinguishing honest models from vicious ones is not trivial, and blindly applying perturbations to the outputs of all models can damage the utility received from honest servers. Users usually observe a trained model $\mathcal{F}$ that is claimed to be trained for a target task, but the exact training objective is unknown. Whether $\mathcal{F}$ only performs the claimed task or it also secretly performs another task is unknown. To this end, we propose a method for *estimating the likelihood of a model being vicious*, based on the idea that a model trained only for the target task should not be far from the "ideal" solution for the target task; if it is only trained using the claimed objective function. On the other hand, if the model is vicious and is trained using another objective function to perform other tasks in parallel to the claimed one, then the model probably has not converged to the ideal solution for the target task. Our proposed defense can work even in black-box scenarios (*e.g.* encrypted computing), and provides a practical estimation for distinguishing honest vs. vicious models by only using a very small set of data points labeled for the target task (see Appendix §I).

By uncovering a major risk in using emerging ML services, this paper helps advance privacy protection for the users' of ML services. Our proposed analysis is just a first look, thus we conclude this paper by discussing current challenges and open directions for future investigations. We open-source our code at github.com/mmalekzadeh/vicious-classifiers.

2 Methodology

Problem Formulation and Threat Model. Let $\mathcal{X} \times \mathcal{Y}$ denote the joint distribution over *data* and *labels* (or *attributes*). We assume each data point $(\mathbf{x}, \mathbf{y}) \sim (\mathcal{X} \times \mathcal{Y})$ either (1) exclusively belongs to one of the N classes (*i.e.,* categorical $\mathcal{Y} = \{1, 2, \ldots, N\}$), or (2) has N binary attributes (*i.e.,* $\mathcal{Y} = \{0,1\}^N$). Let the *server* train a *model* $\mathcal{F}$ on a *target* task $\mathcal{Y}$, where $\mathcal{F}(\mathbf{x}) = \hat{\mathbf{y}} = [\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \ldots, \hat{\mathbf{y}}_N]$ denotes the model's *outputs*; *i.e.,* prediction scores (logits) over $\mathcal{Y}$. For the categorical case, $\mathcal{Y} = \{1, 2, \ldots, N\}$, each $\hat{\mathbf{y}}_i$ shows the logit score for the class i (*e.g.* the score for class i in CIFAR10 dataset). For the binary case, $\mathcal{Y} = \{0,1\}^N$, each $\hat{\mathbf{y}}_i$ shows the logit score for the attribute i (*e.g.* the score for attribute i in CelebA dataset, such as "smiling" attribute). We allow $\mathcal{F}$ to have any arbitrary architecture; *e.g.* to be a single model with N outputs, or to be an ensemble of N models each with a single output, or any other architectural choice. Model $\mathcal{F}$ is trained by the server (which acts as the attacker), thus $\mathcal{F}$ is white-box to the server. At test time, the users will have a complete white-box view of $\mathcal{F}$. We consider two settings: (1) *logit outputs*, where $\hat{\mathbf{y}} \in \mathbb{R}^N$, and (2) *softmax outputs*, where using the standard softmax function, $\hat{\mathbf{y}}$ is normalized to a probability distribution over the possible classes (see Appendix §A).

Training of Target Classifier and Attack Models. We present an algorithm for jointly training $\mathcal{F}$ and $\mathcal{G}$ (Figure 2). The server trains model $\mathcal{F}$ that takes data $\mathbf{x}$ as input and produces N-outputs. Outputs are attached to the *classification loss function* L^C, which computes the amount of inaccuracy in predicting the true attribute $\mathbf{y}$, and thus provides gradients for updating $\mathcal{F}$. For categorical attributes, we use the standard categorical cross-entropy loss

$$L^C(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{n=1}^{N} -\mathbf{y}_n \log \hat{\mathbf{y}}_n, \tag{1}$$

and for binary attributes, we use the class-weighted binary cross-entropy

$$L^C(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{N} \sum_{n=1}^{N} \eta_n \mathbf{y}_n \log \hat{\mathbf{y}}_n + (1 - \mathbf{y}_n) \log(1 - \hat{\mathbf{y}}_n), \tag{2}$$

where η_n denotes the weight of class 1 for attribute $\mathbf{y}_n \in \{0,1\}$, and it is defined as the number of samples labeled 0 divided by the number of samples labeled 1 in the training dataset. CelebA dataset [23] used in our experiments is highly unbalanced for several attributes. Our motivation for using the class-weighted binary loss function is to obtain a fairer classification for unbalanced labels. While training $\mathcal{F}$, the model's outputs are fed into another model $\mathcal{G}$, which aims to reconstruct the original data. The output of $\mathcal{G}$ is attached to a reconstruction loss function L^R, producing gradients for updating both $\mathcal{F}$ and $\mathcal{G}$.

In this paper, we benchmark image datasets in our experiments; thus, we utilize the loss functions used in image processing tasks [44]. In particular, we employ a weighted sum of (*i*) *structural similarity index measure* (SSIM) [35] and (*ii*) *Huber loss* [14] which is a piecewise function including both mean squared error (known as MSELoss) and mean absolute error (MAE, also known as L1Loss) [30]. This design choice of combining a perceptually-motivated loss (*i.e.,* SSIM) with a statistically-motivated loss (*i.e.,* MSELoss or L1Loss) is inspired by the common practice used by previous work in image-processing literature [40, 44]:

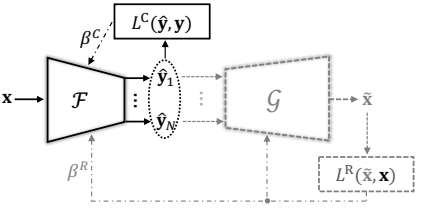

Figure 2: $\mathcal{F}$ is the target model, $\mathcal{G}$ is the attack model, L^C is the classification loss, L^R is the attack reconstruction loss. Both $\mathcal{F}$ and $\mathcal{G}$ are DNNs. Hyperparameters β^C and β^R control the trade-offs between classification and reconstruction tasks.

$$L^R(\tilde{\mathbf{x}},\mathbf{x}) = \alpha SSIM(\tilde{\mathbf{x}},\mathbf{x}) + \gamma Huber(\tilde{\mathbf{x}},\mathbf{x}), \tag{3}$$

where α and γ are the hyperparameters for data reconstruction. Note that, depending on the data type and the attack's purpose, one can use other reconstruction loss functions.

The ultimate loss function. For optimizing the parameters of $\mathcal{F}$, we use:

$$L^{\mathcal{F}} = \beta^C L^C(\hat{\mathbf{y}},\mathbf{y}) + \beta^R L^R(\tilde{\mathbf{x}},\mathbf{x}), \tag{4}$$

where β^C and β^R are the weights that allow us to move along different possible local minimas and both are non-negative real-valued. For optimizing the parameters of $\mathcal{G}$, we only use L^R, but notice that there is an implicit connection between $\mathcal{G}$, $\mathcal{F}$, and L^C since $\mathcal{G}$ acts on $\hat{\mathbf{y}} = \mathcal{F}(\mathbf{x})$. Algorithm 1 (Appendix D) summarizes the explained training procedure.

2.1 Reconstruction Risk

We define $\mathcal{S}(\cdot,\cdot)$ as a measure of *reconstruction risk*, and $\mathcal{S}(\tilde{\mathbf{x}},\mathbf{x}) \geq \mathtt{R}$ means the risk of reconstructing $\mathbf{x}$ is more than $\mathtt{R}$ based on the measure $\mathcal{S}$. Considering the *reconstruction* of data, $\tilde{\mathbf{x}}$, we use $\mathcal{S}$ to measure *privacy loss*. A pivotal question is: what is the most suitable and general $\mathcal{S}$ for computing and evaluating the attacker's success? Previous works (see Appendix §B) mostly rely on common measures such as MSE or SSIM. We propose our measure of reconstruction risk and we compare it with other measures in §3.

Basics. For two random vectors $\mathbf{x}$ and $\tilde{\mathbf{x}}$ of the same distribution with covariance matrix $\Sigma_{\mathbb{D}}$, the Mahalanobis distance (MD) is a dissimilarity measure between $\mathbf{x}$ and $\tilde{\mathbf{x}}$: $\mathsf{d}(\mathbf{x},\tilde{\mathbf{x}}) = \sqrt{(\mathbf{x}-\tilde{\mathbf{x}})\Sigma_{\mathbb{D}}{}^{-1}(\mathbf{x}-\tilde{\mathbf{x}})}$. Similarly, we can compute $\mathsf{d}(\mathbf{x},\mu_{\mathbb{D}})$, where $\mu_{\mathbb{D}}$ is the mean of distribution that $\mathbf{x}$ is drawn from. Notice that if $\Sigma_{\mathbb{D}}$ is the identity matrix, MD reduces to the Euclidean distance (and thus the typical MSE). Notice that in practice, *e.g.* for real-world

data types such as images, the $\Sigma_{\mathbb{D}}$ is rarely close to the identity matrix. The pixels of an image are correlated to each other. Similarly, the sample points of time-series signals are temporally correlated to each other, and so on. Therefore, for computing MD, we need to approximate $\Sigma_{\mathbb{D}}$ using a sample dataset. In our experiments, we approximate $\Sigma_{\mathbb{D}}$ via samples in the training dataset. Another characteristic of MD is that when the data follows a multivariate normal distribution, the probability density of an observation $\mathbf{x}$ is uniquely determined by MD:

$$\Pr(\mathbf{x}) = \frac{1}{\sqrt{det(2\pi\Sigma_{\mathbb{D}})}} \exp\Big(-\frac{\big(\mathsf{d}(\mathbf{x},\mu_{\mathbb{D}})\big)^2}{2}\Big). \tag{5}$$

The multivariate normal distribution is the most common probability distribution that is used for approximating data distribution [29]. Thus, with the assumption that one can approximate data distribution using a multivariate normal distribution, MD can be utilized for computing the probability density of an observation $\mathbf{x}$; *i.e.,* $\Pr(\mathbf{x})$.

Our Measure. Motivated by the characteristics of MD, we assume that the risk of a reconstruction $\widetilde{\mathbf{x}}$ of a sample $\mathbf{x}$ depends on both $\mathsf{d}(\mathbf{x},\widetilde{\mathbf{x}})$ and $\Pr(\mathbf{x})$: the more unlikely is a sample (lower $\Pr(\mathbf{x})$), the more important is the value of $\mathsf{d}(\mathbf{x},\widetilde{\mathbf{x}})$ (the more informative is a specific reconstruction). For the reconstructions of two independent samples $\mathbf{x}^1$ and $\mathbf{x}^2$ with $\mathsf{d}(\mathbf{x}^1,\widetilde{\mathbf{x}}^1) = \mathsf{d}(\mathbf{x}^2,\widetilde{\mathbf{x}}^2)$ and $\Pr(\mathbf{x}^1) < \Pr(\mathbf{x}^2)$, the risk of $\widetilde{\mathbf{x}}^1$ should be higher than $\widetilde{\mathbf{x}}^2$. Our intuition is: because $\mathbf{x}^1$ is less likely than $\mathbf{x}^2$, then $\mathbf{x}^1$ is easier to be identified when attackers observe $\widetilde{\mathbf{x}}^1$, compared to $\mathbf{x}^2$ when attackers observe $\widetilde{\mathbf{x}}^2$. Because $\mathbf{x}^1$ is less likely (or more unique) than $\mathbf{x}^2$, then a reconstruction of $\mathbf{x}^1$ will give the attacker more information.

In general, the intuition is that reconstructing a data point that belongs to a more sparse part of the population is riskier than reconstructing those that belong to a more dense part of the population. To this end, we define the *reconstruction risk* of a model with respect to a benchmark test dataset $\mathbb{D} = \{x^n\}_{n=1}^N$ as

$$\mathtt{R} = \frac{1}{N}\sum_{n=1}^{N} \mathsf{d}(\mathbf{x}^n,\mu_{\mathbb{D}})\big/\mathsf{d}(\mathbf{x}^n,\widetilde{\mathbf{x}}^n). \tag{6}$$

The less likely a sample, or the better its reconstruction quality, the higher is its contribution to the risk of the dataset. Our measure gives a general notion of reconstruction risk that depends on the characteristics of the entire dataset, and not just each sample independently. Moreover, our measure can be used across different data types and is not restricted to images or videos.

Remark. Our proposed $\mathtt{R}$ is task-agnostic. For example, for face images, background reconstruction might not be as important as eyes or mouth reconstruction. For task-specific risk assessments, one might need to perform preprocessing. For example, by image segmentation and applying the computation of $\mathtt{R}$ only to that segment of the photo that includes the face.

3 Experimental Results

Our experimental setup is detailed in Appendix §E. Our main results are reported in Tables 1 (and Appendix Table 6). We compare the *accuracy* of the target task and the *reconstruction quality* for different trade-offs. We consider two settings: during training the attack model, $\mathcal{G}$ receives (i) the *logits* outputs of $\mathcal{F}$, or (ii) the *softmax* outputs. To compare the trade-offs between *accuracy* and *reconstruction quality*, we also show two extremes in training $\mathcal{F}$: *classification only* (when $\beta^R = 0$) and *reconstruction only* (when $\beta^C = 0$). The main findings are summarized as follows.

Table 1: Reconstruction quality vs. classification accuracy in different settings and for different datasets. We repeat each experiment for five different random seeds and report the mean and standard deviation.

Outputs	Dataset	β^R/β^C	PSNR (dB)	SSIM	R	ACC (%)
Logits	MNIST	0/1	0.000 ± 0.000	0.000 ± 0.000	1.000 ± 0.000	99.54 ± 0.11
		1/1	22.090 ± 0.095	0.920 ± 0.003	1.378 ± 0.021	99.53 ± 0.09
		3/1	22.367 ± 0.075	0.926 ± 0.001	1.394 ± 0.012	99.52 ± 0.04
		5/1	22.336 ± 0.049	0.927 ± 0.001	1.392 ± 0.011	99.55 ± 0.04
		1/0	22.169 ± 0.052	0.926 ± 0.001	1.398 ± 0.009	10.00 ± 0.00
	FMNIST	0/1	0.000 ± 0.000	0.000 ± 0.000	1.000 ± 0.000	94.33 ± 0.12
		1/1	20.525 ± 0.056	0.783 ± 0.001	1.243 ± 0.003	94.58 ± 0.05
		3/1	20.883 ± 0.061	0.799 ± 0.001	1.272 ± 0.006	94.24 ± 0.08
		5/1	20.871 ± 0.068	0.803 ± 0.001	1.271 ± 0.004	94.30 ± 0.16
		1/0	21.021 ± 0.032	0.810 ± 0.001	1.281 ± 0.008	10.00 ± 0.00
	CIFAR10	0/1	0.000 ± 0.000	0.000 ± 0.000	1.000 ± 0.000	91.86 ± 0.69
		1/1	15.388 ± 0.039	0.377 ± 0.002	1.009 ± 0.003	91.34 ± 0.58
		3/1	15.550 ± 0.042	0.406 ± 0.001	1.013 ± 0.001	90.84 ± 0.35
		5/1	15.581 ± 0.043	0.414 ± 0.003	1.010 ± 0.003	90.62 ± 0.62
		1/0	15.784 ± 0.037	0.468 ± 0.000	1.026 ± 0.002	10.00 ± 0.00
	CIFAR100	0/1	0.000 ± 0.000	0.000 ± 0.000	1.000 ± 0.000	68.13 ± 0.66
		1/1	16.757 ± 0.142	0.473 ± 0.012	1.051 ± 0.003	67.50 ± 0.66
		3/1	18.463 ± 0.311	0.646 ± 0.018	1.147 ± 0.023	64.57 ± 1.20
		5/1	19.047 ± 0.235	0.701 ± 0.008	1.201 ± 0.015	61.25 ± 1.34
		1/0	20.693 ± 0.097	0.821 ± 0.002	1.454 ± 0.011	1.00 ± 0.00
	TinyImgNet	0/1	0.000 ± 0.000	0.000 ± 0.000	1.000 ± 0.000	46.96 ± 0.25
		1/1	16.763 ± 0.171	0.473 ± 0.014	1.042 ± 0.004	45.98 ± 0.56
		3/1	19.036 ± 0.190	0.692 ± 0.011	1.166 ± 0.016	42.71 ± 0.22
		5/1	20.072 ± 0.181	0.766 ± 0.009	1.261 ± 0.019	37.57 ± 1.39
		1/0	23.166 ± 0.104	0.900 ± 0.004	1.796 ± 0.028	0.50 ± 0.00
	CelebA	0/1	0.000 ± 0.000	0.000 ± 0.000	1.000 ± 0.000	88.43 ± 0.00
		1/1	19.081 ± 0.044	0.813 ± 0.000	1.238 ± 0.002	88.03 ± 0.06
		3/1	19.619 ± 0.087	0.837 ± 0.002	1.287 ± 0.009	86.41 ± 0.37
		5/1	19.850 ± 0.018	0.845 ± 0.000	1.308 ± 0.002	85.82 ± 0.20
		1/0	20.292 ± 0.021	0.858 ± 0.001	1.346 ± 0.002	50.00 ± 0.00
Softmax	MNIST	1/1	16.061 ± 0.186	0.684 ± 0.012	1.037 ± 0.006	99.54 ± 0.05
		5/1	21.036 ± 0.127	0.897 ± 0.004	1.272 ± 0.018	99.62 ± 0.04
	FMNIST	1/1	18.009 ± 0.243	0.684 ± 0.011	1.086 ± 0.010	94.38 ± 0.06
		5/1	19.915 ± 0.182	0.771 ± 0.006	1.204 ± 0.007	94.29 ± 0.35
	CIFAR10	1/1	13.934 ± 0.246	0.256 ± 0.031	1.003 ± 0.001	91.75 ± 0.38
		5/1	15.389 ± 0.061	0.409 ± 0.004	1.015 ± 0.001	90.47 ± 0.56
	CIFAR100	1/1	12.846 ± 0.125	0.202 ± 0.002	1.002 ± 0.001	67.21 ± 0.18
		5/1	16.580 ± 0.138	0.510 ± 0.002	1.054 ± 0.001	65.65 ± 1.21
	TinyImgNet	1/1	13.466 ± 0.042	0.203 ± 0.003	1.003 ± 0.000	44.08 ± 1.70
		5/1	17.168 ± 0.084	0.569 ± 0.010	1.066 ± 0.003	42.38 ± 0.68

Each *experiment* includes training $\mathcal{F}$ and $\mathcal{G}$ on the training dataset, for 50 epochs, and choosing $\mathcal{F}$ and $\mathcal{G}$ of the epoch in which we achieve the best result on the validation set according to loss function in Equation (4). In each experiment, via the test dataset, we evaluate $\mathcal{F}$ by measuring the accuracy of $\mathcal{F}$ in estimating the public task using Equations (7) or (8), and we evaluate $\mathcal{G}$ by computing the reconstruction quality measured by PSNR, SSIM, and our proposed reconstruction risk $\mathtt{R}$ in (6). To compute $\mathtt{R}$, we approximate $\mu_{\mathbb{D}}$ and $\Sigma_{\mathbb{D}}$ via samples in the training set. For a fair comparison, we use a random seed that is fixed throughout all the experiments, thus the same model initialization and data sampling are used.

(1) Trade-offs. When the logit outputs are available, the attacker can keep the classification accuracy very close to that achieved by a honest model, while achieving a reconstruction quality close to that of reconstruction only. For instance, even for relatively complex samples from TinyImgNet, we observe that with about 4% loss in accuracy (compared to classification-only setting), we get a reconstruction quality of around 19 dB PSNR and 0.7 SSIM; which is not as perfect as reconstruction only but can be considered as a serious privacy risk. We observe serious privacy risks for other, less complex data types. We do not perform any hyper-parameter or network architecture search (as it is not the main focus of our work). However, our results show that if one can perform such a search and find a configuration that achieves better performance (compared to our default WideResNet) in classification- and reconstruction-only settings, then such a model is also capable of achieving a better trade-off. Thus, our current results can be seen as a lower bound on the capability of an attacker.

(2) Data Type. For grayscale images (MNIST and FMNIST) we demonstrate very successful attacks. For colored images (CIFAR10, CIFAR100, TinyImgNet), it is harder to achieve as good trade-offs as those achieved for grayscale images. However, as one would expect, when the number of classes goes up, *e.g.* from 10 to 100 to 200, the quality of reconstruction also becomes much better, *e.g.* from PSNR of about 15 to about 18 to about 20 dB, for CIFAR10, CIFAR100, and TinyImgNet respectively.

(3) Logits vs. Softmax. As one may expect, transforming logits into softmax probabilities will make it harder to establish a good trade-off. However, we still observe reasonably good trade-offs for low-complexity data types. The difficulty is more visible when the data complexity goes up. For TinyImgNet in the softmax setting, we can get almost the same reconstruction quality of the logit setting (around 17dB PSNR and 0.57 SSIM); however, the classification accuracy in the softmax setting drops by about 3% to compensate for this. Notice that for the CelebA dataset, we cannot transform the outputs into softmax as the attributes are binary. Instead, we can use the sigmoid function, which is a one-to-one function, and allow the server to easily transform the received sigmoid outputs into logit outputs. Hence, the server can train $\mathcal{F}$ and $\mathcal{G}$ in the logit setting, and after training just attach a sigmoid activation function to the output layer. The fact that the shortcoming of the sigmoid setting can be easily resolved by such a simple trick will facilitate such attacks as releasing sigmoid values might look less suspicious. As a side note, *softmax* functions, unlike sigmoids, are not one-to-one; since $softmax(x) = softmax(x+a)$ for all real-valued a. Thus, such a trick cannot be applied to categorical attributes with more than two classes, where users might release the *softmax* outputs instead of raw ones. In such settings, a server can replace categorical attributes of size C with C binary attributes. We leave the investigation of such a replacement to a future study.

(4) The value of N. For the CelebA, we observe that the reconstruction quality improves with the number of binary attributes N; however, the improvement is not linear. With $\beta^R/\beta^C = 3$ we have about 0.14 points improvement in SSIM and about 2 dB in PSNR when going from $N = 1$ to $N = 5$ attributes, but when moving from $N = 5$ to $N = 10$ we observe an improvement of only 0.05 in SSIM and about 1 db in PSNR. A similar diminishing increase

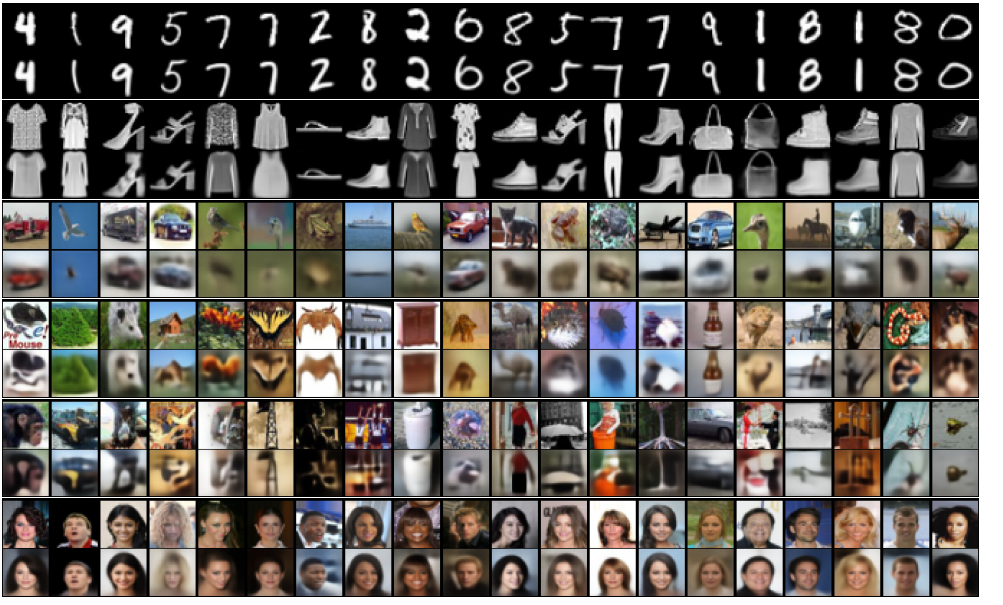

Figure 3: Examples of image reconstruction with the logit outputs for $\beta^R/\beta^C = 3/1$ in Table 1 for MNIST, FMNIST, CIFAR10, CIFAR100, TinyImageNet, and CelebA datasets (from top to bottom). For each dataset, the first row consists of the original images and the second row is the reconstructed data by the attacker.

happens also when we move from $N = 10$ to $N = 40$ attributes. In sum, these results suggest that the proposed attack achieves meaningful performance on CelebA with just a few outputs ($10 \leq N \leq 20$), and such scenarios of collecting a few binary attributes can lie within several applications provided by real-word ML service providers.

(5) The properties of $\mathtt{R}$. We demonstrate the properties of our proposed reconstruction risk $\mathtt{R}$, compared to PSNR and SSIM; based on our main results reported in Table 1. The values of $\mathtt{R}$ are in *conformity* with the values of PSNR and SSIM: the higher $\mathtt{R}$ is, it indicates the higher PSNR and SSIM are. However, the values of PSNR and SSIM depend on the complexity of the data type, but the values of $\mathtt{R}$ demonstrate more *homogeneity* across data types. For example, in some situations, PSNR is almost the same but SSIM is different. For instance, with a similar PSNR of 19 dB for both CIFAR100 and CelebA, we observe different SSIM of 0.7 for CIFAR100 and 0.8 for CelebA). By combining both PSNR and SSIM, one can conclude that the model reveals more information on CelebA than CIFAR100. This can be seen by the values $\mathtt{R}$ in which we have 1.23 for CelebA compared to 1.2 for CIFAR100. On the other hand, there are situations in which SSIM is almost the same but PSNR is different. For instance, SSIM of 0.81 for both FMNIST and CelebA corresponds to a PSNR of 21 dB for FMNIST and 19 dB for CelebA. Similarly, by combining these two measures, we expect the model to leak more information for FMNIST than CelebA. Again, this conclusion can be made by observing $\mathtt{R}$ which is 1.28 for FMNIST and 1.23 for CelebA. Overall, $\mathtt{R}$ provides a more consistent and unified measure of reconstruction risk, as it is based on a more general notion of distance than other data-specific measures (see §2.1).

(6) Qualitative comparison. Figure 3 (and similarly Appendix Figure 4) show examples of data reconstruction. An interesting observation is that the reconstructed images are very similar to the original samples. We emphasize that in this paper we used off-the-self DNNs, and leave the design and optimization of dedicated DNN architectures to future studies.

4 Discussion

Limitations. Addressing all open questions when studying such a privacy risk is challenging. Throughout our evaluations, we decided to fix some variables, such as fixing data type to image data and the model architecture to WideResNet. We were motivated by the aim to allocate space and resources for thorough analysis and evaluation of more important variables, such as sample size, input and output complexity, hyper-parameters of the algorithms, potential defense, etc. Overall, our theoretical analysis, as well as the independence of our algorithm to input and model, suggests that similar results can be generalized to other data types and model architectures.

Future Work. (1) We only considered a single inference, but there are scenarios where multiple inferences are made on a user's private data; such as ensemble prediction using multiple models or Monte Carlo dropout. (2) Combining the outputs' of multiple models on the same data to improve the reconstruction quality is an open question. (3) We mainly focus on understanding the attack, and our initial effort on the defense is to inspire the community, to investigate more efficient and effective defenses. Our proposed defense mechanism needs several rounds of training and defining a threshold for attack detection. Considering defenses that potentially do not need training or can detect vicious models more accurately is also a key topic to explore. (4) Our focus is on classification; however, the foundational principles of our work apply to regression as well. Hence, one may consider exploring such potential attacks for regression models. (5) Finally, our proposed reconstruction risk can be further improved by comparing it with similar measures introduced for other attacks, such as the 'calibrated score' in [36] on membership inference attacks.

5 Conclusion

A growing paradigm in edge computing, motivated by efficiency and privacy, is "bringing the code to the data". In this work, we challenge the privacy aspect of this paradigm by showing the possibility of data reconstruction from the outputs' of a target machine learning task. We benchmark data reconstruction risk and offer a unified measure for assessing the risk of data reconstruction. While detecting such a privacy attack is not trivial, we also take an initial step by proposing a practical technique for examining ML classifiers. We believe that our paper will serve as an inspiration for further explorations, in both attack and defense methods, to enhance data privacy in edge computing.

Acknowledgment. This work was partially supported by the UK EPSRC grant (grant no. EP/T023600/1 and EP/W035960/1) under the CHIST-ERA program. We would like to thank James Townsend for a fruitful discussion about this work.

References

[1] Shengwei An, Guanhong Tao, Qiuling Xu, Yingqi Liu, Guangyu Shen, Yuan Yao, Jingwei Xu, and Xiangyu Zhang. Mirror: Model inversion for deep learning network with high fidelity. In *Proceedings of the 29th Network and Distributed System Security Symposium*, 2022.

[2] Arthur T Benjamin and Jennifer J Quinn. *Proofs that Really Count: the Art of Combinatorial Proof*. Number 27. MAA, 2003.

[3] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning Attacks against Support Vector Machines. In *International Conference on Machine Learning*, 2012.

[4] Florian Bourse, Michele Minelli, Matthias Minihold, and Pascal Paillier. Fast Homomorphic Evaluation of Deep Discretized Neural Networks. In *Annual International Cryptology Conference*, 2018.

[5] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[6] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The Secret Sharer: Evaluating and Testing Unintended Memorization In Neural Networks. In *USENIX Security Symposium*, 2019.

[7] Rich Caruana. Multitask Learning. *Springer Machine Learning*, 28(1), 1997.

[8] Michael Crawshaw. Multi-Task Learning with Deep Neural Networks: A Survey. *arXiv:2009.09796*, 2020.

[9] Cynthia Dwork, Aaron Roth, et al. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4), 2014.

[10] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In *ACM SIGSAC Conference on Computer and Communications Security*, 2015.

[11] GDPR. Data Protection and Online Privacy. https://europa.eu/youreurope/citizens/consumers/internet-telecoms/data-protection-online-privacy/, 2018.

[12] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33, 2020.

[13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based. In *Springer Annual Cryptology Conference*, 2013.

[14] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The Elements of Statistical Learning: Data mining, Inference, and Prediction*. 2009.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[16] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning. In *IEEE Symposium on Security and Privacy*, 2018.

[17] Xue Jiang, Xuebing Zhou, and Jens Grossklags. Comprehensive Analysis of Privacy Leakage in Vertical Federated Learning During Prediction. *Proc. Priv. Enhancing Technol. (PETS)*, (2):263–281, 2022.

[18] Michael Kearns and Aaron Roth. *The Ethical Algorithm: The Science of Socially Aware Algorithm Design*. Oxford University Press, 2019.

[19] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2014.

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning Multiple Layers of Features from Tiny Images. 2009.

[21] Ya Le and Xuan Yang. Tiny Imagenet Visual Recognition Challenge. *CS 231N*, 2015.

[22] Yann LeCun. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[23] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. In *International Conference on Computer Vision*, 2015.

[24] Xinjian Luo, Yuncheng Wu, Xiaokui Xiao, and Beng Chin Ooi. Feature Inference Attack on Model Predictions in Vertical Federated Learning. In *IEEE International Conference on Data Engineering (ICDE)*, 2021.

[25] Mohammad Malekzadeh, Anastasia Borovykh, and Deniz Gündüz. Honest-but-curious nets: Sensitive attributes of private inputs can be secretly coded into the classifiers' outputs. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 825–844, 2021.

[26] Andrey Malinin and Mark Gales. Predictive Uncertainty Estimation via Prior Networks. *Advances in neural information processing systems*, 2018.

[27] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting Unintended Feature Leakage in Collaborative Learning. In *IEEE Symposium on Security and Privacy (S&P)*, 2019.

[28] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. Towards Poisoning of Deep Learning Algorithms with Back-Gradient Optimization. In *ACM Workshop on Artificial Intelligence and Security (AISec)*, 2017.

[29] Kevin P Murphy. *Probabilistic Machine Learning: An Introduction*. MIT Press, 2021.

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, 2019.

[31] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. In *Network and Distributed Systems Security*, 2018.

[32] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models. In *IEEE Symposium on Security and Privacy (S&P)*, 2017.

[33] Congzheng Song and Vitaly Shmatikov. Overlearning Reveals Sensitive Attributes. In *Conference on Learning Representations*, 2020.

[34] Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. Distributed Deep Neural Networks over the Cloud, the Edge and End Devices. In *International Conference on Distributed Computing Systems*. IEEE, 2017.

[35] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image Quality Assessment: from Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4), 2004.

[36] Lauren Watson, Chuan Guo, Graham Cormode, and Alexandre Sablayrolles. On the importance of difficulty calibration in membership inference attacks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=3eIrli0TwQ.

[37] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv:1708.07747*, 2017.

[38] Ziqi Yang, Jiyi Zhang, Ee-Chien Chang, and Zhenkai Liang. Neural network inversion in adversarial setting via background knowledge alignment. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019.

[39] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16337–16346, 2021.

[40] Jihyeong Yoo and Qifeng Chen. SinIR: Efficient General Image Manipulation with Single Image Reconstruction. In *International Conference on Machine Learning*, 2021.

[41] Sergey Zagoruyko and Nikos Komodakis. Wide Residual Networks. In *BMVC*, 2016.

[42] Yu Zhang and Qiang Yang. A Survey on Multi-Task Learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[43] Zhifei Zhang, Yang Song, and Hairong Qi. Age Progression/Regression by Conditional Adversarial Autoencoder. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[44] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss Functions for Image Restoration with Neural Networks. *IEEE Transactions on computational imaging*, 3(1), 2016.

[45] Zhi Zhou, Xu Chen, En Li, Liekang Zeng, Ke Luo, and Junshan Zhang. Edge Intelligence: Paving the Last Mile of Artificial Intelligence with Edge Computing. *Proceedings of the IEEE*, 107(8), 2019.

[46] Ligeng Zhu, Zhijian Liu, and Song Han. Deep Leakage From Gradients. In *Advances in Neural Information Processing Systems*, 2019.