Personalized Privacy-Preserving Federated Learning

Cédric Boscher, Nawel Benarba, Fatima Elhattab, Sara Bouchenak

Personalized Privacy-Preserving Federated Learning

Cédric Boscher
INSA Lyon – LIRIS
Lyon, France
cedric.boscher@insa-lyon.fr

Nawel Benarba
INSA Lyon – LIRIS
Lyon, France
nawel.benarba@insa-lyon.fr

Fatima Elhattab
INSA Lyon – LIRIS
Lyon, France
fatima.elhattab@insa-lyon.fr

Sara Bouchenak
INSA Lyon – LIRIS
Lyon, France
sara.bouchenak@insa-lyon.fr

ABSTRACT

Federated Learning (FL) enables collaborative model training among several participants while keeping local data private. However, FL remains vulnerable to privacy membership inference attacks (MIAs) that allow adversaries to deduce confidential information about participants' training data. Existing defense mechanisms against MIAs compromise model performance and utility, and incur significant overheads. In this paper, we propose DINAR, a novel FL middleware for privacy-preserving neural networks that precisely handles these issues. DINAR leverages personalized FL and follows a fine-grained approach that specifically tackles FL neural network layers that leak more private information than other layers, thus, efficiently protecting FL model against MIAs in a non-intrusive way, while compensating for any potential loss in the model accuracy. The paper presents our extensive empirical evaluation of DINAR, conducted with six widely used datasets, four neural networks, and comparing against five state-of-the-art FL privacy protection mechanisms. The evaluation results show that DINAR reduces the membership inference attack success rate to reach its optimal value, without hurting model accuracy, and without inducing computational overhead. In contrast, existing FL defense mechanisms incur an overhead of up to +35% and +3,000% on respectively FL client-side and FL server-side computation times.

CCS CONCEPTS

• **Computing methodologies** → **Distributed computing methodologies**; **Machine learning**; **Distributed artificial intelligence**; • **Security and privacy** → **Privacy-preserving protocols**.

KEYWORDS

Federated Learning, Privacy Protection, Membership Inference Attacks

ACM Reference Format:
Cédric Boscher, Nawel Benarba, Fatima Elhattab, and Sara Bouchenak. 2024. Personalized Privacy-Preserving Federated Learning. In *25th International Middleware Conference (MIDDLEWARE '24), December 2–6, 2024, Hong Kong, Hong Kong*. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3652892.3700785

MIDDLEWARE '24, December 2–6, 2024, Hong Kong, Hong Kong

ACM ISBN 979-8-4007-0623-3/24/12
https://doi.org/10.1145/3652892.3700785

1 INTRODUCTION

Advancements in Machine Learning (ML), along with the need for better privacy, have given rise to the Federated Learning (FL) paradigm. FL enables collaborative model training among decentralized participants' devices, while keeping local data private at the participants' premises [26]. Thus, participants contribute by training their respective local models using their private data, and only transmit their local model parameters to a FL server, which aggregates these parameters to produce a global model. FL has various applications, such as e-health monitoring [47], disease diagnosis [20], and fraud detection in banking systems [8].

Despite the privacy benefits offered by FL, recent studies have highlighted the vulnerability of FL systems to privacy inference attacks [24, 34]. These attacks, in particular Membership Inference Attacks (MIAs), exploit the parameters of the shared models to infer sensitive information about the training data of other participants [41]. In a white-box FL system where the model architecture and parameters are known by all participants, MIAs pose a significant threat to privacy. An attacker on the server side could discern from a client's model parameters whether a specific individual's data was included in the training process. Similarly, a malicious participant on the client side could deduce from the FL model parameters whether the data was used for training, and potentially uncover sensitive information.

To address these privacy concerns, various FL defense mechanisms have been proposed [1, 33, 36, 37]. These mechanisms leverage techniques such as cryptography and secure multiparty computation [4, 48, 52], trusted hardware execution environments [19, 28], perturbation-based methods and differential privacy [33, 43, 50]. Software and hardware-based cryptographic solutions provide interesting theoretical privacy guarantees, although at the expense of high computational overheads. On the other hand, in order to provide effective privacy, existing perturbation-based methods negatively impact model utility and quality, and incur significant overheads.

Our objective is to precisely strike a balance between FL model privacy, model utility and costs for enabling effective privacy-preserving FL, especially in the case of cross-silo FL systems where the FL server shares the global model with the participating clients and not with external parties (*e.g.*, FL-based banking systems, hospitals, etc.). In this paper, we propose DINAR, a novel FL middleware

for privacy-preserving neural networks that handles MIAs. DINAR is based on a simple yet effective approach that consists in protecting more specifically the FL model layer that is the most sensitive to membership privacy leakage. This approach is motivated by recent studies [29, 30], and by our empirical analysis in §3, which reveal the existence of a model layer that leaks more private information than other layers. DINAR follows a fine-grained and specialized approach that specifically tackles FL neural network layers that leak more private information than other layers, thus, efficiently protecting the FL model against MIAs in a non-intrusive way. And in order to compensate for any potential loss in the accuracy of the protected model, DINAR leverages personalized FL and combines it with efficient adaptive gradient descent.

DINAR runs at the FL client side, and allows protecting both the global FL model and the client models. Whereas for its own model predictions the client uses its privacy-sensitive layer as part of the model, that privacy-sensitive layer is obfuscated before sending client model updates to the FL server. Thus, the aggregated model produced by the FL server includes an obfuscated version of the privacy-sensitive layer. When the client receives the protected global model from the server, it first restores its local privacy-sensitive layer (*i.e.*, the non-obfuscated version of that layer) that was stored during the previous FL round, and integrates it into its copy of the global model, before actually using the resulting personalized model for client predictions. Furthermore, in order to improve the utility of the protected model, DINAR leverages an efficient adaptive gradient descent technique to further maximize the model accuracy [6].

Scientific Contributions. In particular, the paper makes the following contributions:

- We conduct an empirical analysis on real datasets and neural networks to characterize how much each layer of a neural network leaks membership privacy information.
- To the best of our knowledge, we propose the first fine-grained FL privacy-preserving middleware against MIAs, that specifically obfuscates the most privacy-sensitive layer, for an effective yet non-intrusive privacy protection.
- We conduct extensive empirical evaluations of DINAR with six widely used datasets and four neural networks. We also compare DINAR against five state-of-the-art FL privacy protection mechanisms. Our evaluation results show that DINAR reduces the membership inference attack success rate to reach its optimal value, without hurting model accuracy and without inducing overheads. In contrast, existing FL defense mechanisms incur an overhead of up to +3(% and +3,000% on respectively FL client-side and FL server-side computation times.
- The software prototype of DINAR is available for other researchers and practitioners at: https://github.com/sara-bouchenak/DINAR/

Paper Roadmap. The remainder of the paper is organized as follows. In §2, we provide an overview of the background and related work pertaining to FL defenses against MIAs. Section 3 motivates DINAR's approach, and §4 elaborates on the design principles that underpin DINAR. To substantiate the efficiency of the proposed solution, empirical evaluations are presented in §5. Finally, in §6 we draw our conclusions.

2 BACKGROUND AND RELATED WORK

2.1 Federated Learning

At each Federated Learning (FL) round, the FL server selects N participating clients, which train their local models θ_i using their own data D_i. Then, clients transmit their model updates to the FL server, which aggregates them to produce a global model θ shared with the clients. The classical algorithm used for model aggregation is FedAvg, a weighted averaging scheme that assigns a weight to a client's model parameters according to the relative amount of data contributed by that client. Furthermore, we consider the case where the FL server shares the global model with the participating clients and not with external parties. This is usually the case in cross-silo FL systems, such as banking systems, or hospitals [11, 12, 35, 53].

2.2 Membership Inference Attack Threat Model

Membership Inference Attacks (MIAs) aim to infer whether a data sample has been used to train a given model. Such attacks exploit vulnerabilities in the parameters and statistical properties of the trained model to reveal information about the training data.

Thus, it is important to safeguard individuals' confidentiality against MIAs that cause significant privacy violations, in particular, in areas involving highly sensitive information such as health applications, financial systems, etc.

Attacker's Objective and Capabilities. We consider the standard setting of a MIA and its underlying attacker's capabilities [41]. Namely, the objective of the attacker is to determine whether a given data sample was used for model training. An attacker can be on the client side or on the server side. If the attacker is on the client side, its goal is to determine, based on the received global FL model, whether a data sample has been used for training by other clients, without knowing to which client it actually belongs. If the attacker is on the server side, it is also able to determine, based on a received client model, whether a data sample has been used by that client for training.

2.3 Related Work

Cutting-edge research in countering MIAs has made significant strides through innovative approaches, encompassing cryptographic techniques, secure hardware, and perturbation-based methods as summarized in Table 1. Cryptography-based solutions such as PEFL [52], HybridAlpha [48], Secure Aggregation (SA) [54], or Chen et al. [4], offer robust privacy solutions, with interesting theoretical guarantees. However, they tend to incur high computational costs due to complex encryption and decryption processes. Furthermore, these solutions often protect either the client-side or the server-side model, but not both, leaving potential vulnerabilities in the other unprotected component. Interestingly, solutions based on Trusted Execution Environments (TEEs) emerge as another alternative for better privacy protection [19, 28, 31]. However, because of

Table 1: Comparison of FL privacy-preserving methods

Privacy-preserving category	Protection method	Model privacy	Model utility	Negligible overhead
Cryptography-based methods	PEFL [52]	✓	✓	✗✗
	HybridAlpha [48]	✓	✓	✗✗
	Chen et al. [4]	✓	✓	✗✗
	Secure Aggregation [54]	✓	✓	✗
TEE-based methods	MixNN [19]	✓	✓	✗✗
	GradSec [28]	✓	✓	✗✗
	PPFL [31]	✓	✓	✗✗
Perturbation-based methods	CDP [33]	✓	✗	✗
	LDP [3]	✓	✗	✗
	FedGP [44]	✓	✗	✗
	WDP [43]	✗	✓	✗
	PFA [21]	✓	✓	✗
	MR-MTL [22]	✗	✓	✗
	DP-FedSAM [40]	✓	✓	✗
	PrivateFL [50]	✗	✓	✗
Gradient Compression	Fu et al. [7]	✓	✓	✗
Our method	*DINAR*	✓	✓	✓

the high dimension of underlying models, striking a tight balance between privacy and computational overhead remains challenging. A recent work on TEE-based privacy-preserving FL reports a performance overhead of up to +646% on training times, and up to +5968% [31] on FL aggregation times.

On the other hand, perturbation methods such as differential privacy (DP), with algorithm-specific random noise injection, serve as interesting safeguards against potential information leakage. When applied in the context of FL, DP has two main forms, namely Local Differential Privacy (LDP) that applies on client model parameters before transmission to the FL server [3], and Central Differential Privacy (CDP) where the server applies DP on aggregated model parameters before sending the resulting model to the clients [33]. There is also Weak Differential Privacy (WDP) which applies norm bounding and Gaussian noise with a low magnitude for better model utility [43].

Recent works, such as PFA [21], MR-MTL [22], DP-FedSAM [40], and PrivateFL [50], follow such approaches. However, in practice, to effectively protect privacy, existing DP-based FL methods induce a significant impact on utility and model accuracy, as shown in the evaluation presented later in the paper. Another approach to counter MIAs in FL is through Gradient Compression (GC) techniques, which reduce the amount of information available for the attacker [7]. However, such techniques also decrease the model utility.

In summary, existing FL privacy-preserving methods tackling MIAs either rely on cryptographic techniques and secure environments which induce a high computational overhead, or reduce model utility and quality with classical perturbation-based methods. In contrast, we propose a novel method that follows a finer-grained approach, applying obfuscation on specific parts of model parameters that leak privacy-sensitive information. This results in good privacy protection, good model utility, and no perceptible computational overhead.

3 MOTIVATION FOR A FINE-GRAINED PRIVACY-PRESERVING APPROACH

Recent studies analyzed the privacy risks of neural networks at a fine-grained level, to better characterize how much each layer of the model leaks privacy information [29, 30, 49]. As claimed in these studies, a similar pattern appears in all models, namely, there is a layer that leaks more private information than other layers. To better illustrate this behavior, we conduct an empirical analysis with four different datasets and their underlying models, deployed in a FL setting*.

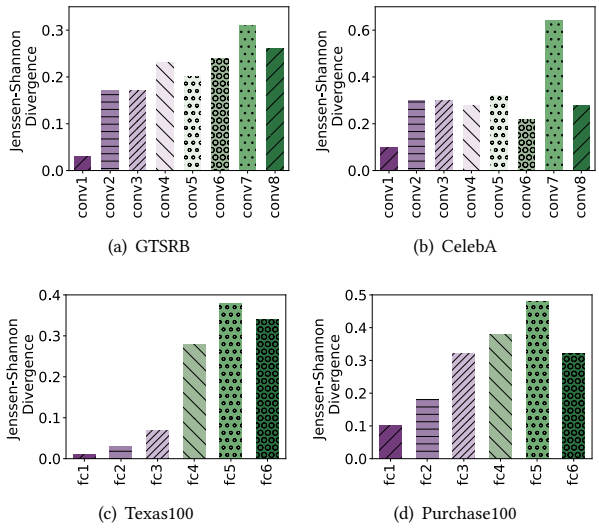


Figure 1: Neural network's layer-level analysis of divergence between member data samples and non-member data samples, when FL models are not protected against MIAs

*A description of the used datasets (GTSRB, CelebA, Texas100, Purchase100) and their underlying models can be found in §5.1 and §5.3.

More precisely, we aim to characterize how much each layer of a model contributes to an attacker's ability to perform membership inference attacks (MIAs). As described in §2.2, such an attacker is able to determine whether a particular data sample was part of the model's training set. In other words, the attacker aims to distinguish between data samples that are members of the training set and those that are not (*i.e.*, non-member data samples). To investigate this, we use a trained FL model to make two sets of predictions: one with member data samples, and another one with non-member data samples. Then, we calculate the gradients of each layer produced by the predictions with member data on the one hand, and with non-member data on the other hand. Next, we determine the generalization gap for each layer, that is the difference between the gradients of member and non-member samples. The higher the generalization gap, the more successful MIA is, *i.e.*, the easier it is for the MIA to differentiate between members and non-members, as shown in recent studies [10, 46].

Our empirical results are presented in Figure 1, where the generalization gap is computed using the widely used Jensen-Shannon divergence [27]. We observe that different layers of a model may exhibit different generalization gaps. Here, we also observe a similar behavior in all datasets and model architectures, namely, the generalization gap of the penultimate layer is notably higher than the generalization gap of the other layers. Thus, that layer leaks more privacy-sensitive information (*i.e.*, membership-related information), as shown in other studies [29, 30]. This motivates the design of DINAR, a fine-grained privacy-preserving approach that tackles precisely the most sensitive neural network layer to protect against MIAs while reducing the impact on utility and overhead.

4 DESIGN PRINCIPLES OF DINAR

We propose DINAR, a novel FL middleware for privacy protection against MIAs. The objective of DINAR is threefold: (i) improving the resilience of neural network models against MIAs, (ii) preserving the model utility, and (iii) avoiding additional computational overheads. Whereas existing privacy-preserving FL methods either apply perturbation on all model layers, or use cryptographic techniques and secure environments which induce a high computational overhead (as discussed in §2.3), the intuition behind DINAR is to specifically handle the most *privacy-sensitive layer* of a FL model, *i.e.*, the layer which reveals more data privacy information than the others. This allows a non-intrusive yet effective solution to protect FL models against MIAs.

DINAR runs on the client side. Its overall pipeline is described in Figure 2. In a preliminary phase, FL clients run a distributed consensus protocol to agree on the most privacy-sensitive layer, as described in §4.1. Then, each selected FL client interacts with the FL server at each round as usual. In addition, the client runs DINAR's privacy protection Algorithm 1 at each FL round. This consists of the successive stages of *client model personalization*, *efficient adaptive model training* for improving model utility, and *model obfuscation*, as respectively detailed in §4.3, §4.4 and §4.2.

Algorithm 1: DINAR privacy protection on FL Client$_i$

Inputs: θ: global model parameters; p: private layer index
Output: θ_i: client model parameters
Local variables: θ_i^{p*}: parameters of private layer of client model; $(B^i, Y) = \{(B_1^i, Y_1), \ldots, (B_x^i, Y_x)\}$: training batches of Client$_i$; η: learning rate

1 **Model Personalization**
2 **for** j in $\{1..J\}$ **do**
3 **if** $j \neq p$ **then**
4 $\theta_i^j \leftarrow \theta^j$; // Use j^{th} layer parameters from global model
5 **else**
6 $\theta_i^j \leftarrow \theta_i^{p*}$; // Restore parameters of client's private layer
7 **Adaptive Model Training**
8 $G \leftarrow 0$; // Set initial accumulated gradients matrix
9 **foreach** local training epoch **do**
10 **foreach** $(B_k^i, Y_k) \in (B^i, Y)$ **do**
11 $\widehat{Y}_k \leftarrow \theta_i(B_k^i)$; // Perform local prediction
12 $loss \leftarrow \mathcal{L}(Y_k, \widehat{Y}_k)$; // Compute model loss
13 $G \leftarrow G + \nabla_\theta \cdot loss^2$; // Compute new cumulated gradients
14 $\theta_i \leftarrow \theta_i - \eta \frac{\nabla_\theta \cdot loss}{\sqrt{G}+1e^{-5}}$; // Update local model
15 **Model Obfuscation**
16 $\theta_i^{p*} \leftarrow \theta_i^p$; // Save parameters of client's private layer
17 $\theta_i^p \leftarrow$ random_values ; // Obfuscate parameters of private layer
18 **return** θ_i

4.1 DINAR Initialization

This preliminary phase of DINAR runs before the FL learning process (*i.e.*, the successive FL rounds), and aims to determine the most privacy-sensitive layer p of the neural network. To do so, the FL clients run a distributed consensus protocol. Each $Client_i$ has a set of raw data D_i, that will first be prepared following classical data preprocessing techniques [38], which results in a set of data D_i^m that will be actually used for model training, and a set of data D_i^n not used for training.

We assume each $Client_i$ has a set of data used for training, D_i^m, and a set of data not used for training, D_i^n. $Client_i$ evaluates the privacy sensitivity of its model layers by measuring the generalization gap, computing the Jensen-Shannon divergence between the gradients of each layer resulting from the predictions of member data samples D_i^m, and non-member data samples D_i^n. Layers exhibiting higher generalization gaps indicate greater privacy sensitivity. Consequently, the layer with the highest generalization gap is the $Client_i$ most privacy-sensitive layer p_i.

To achieve a consensus on the index of the most privacy-sensitive layer p to obfuscate among all clients, even in the presence of Byzantine faults where some clients may be compromised or behave maliciously, we use a broadcast distributed voting method [2] which is based on distributed multi-choice voting (DMVR) algorithm [39]. This method involves each client broadcasting its p_i to all other FL clients. Upon receiving all indices, it ensures that the value with the

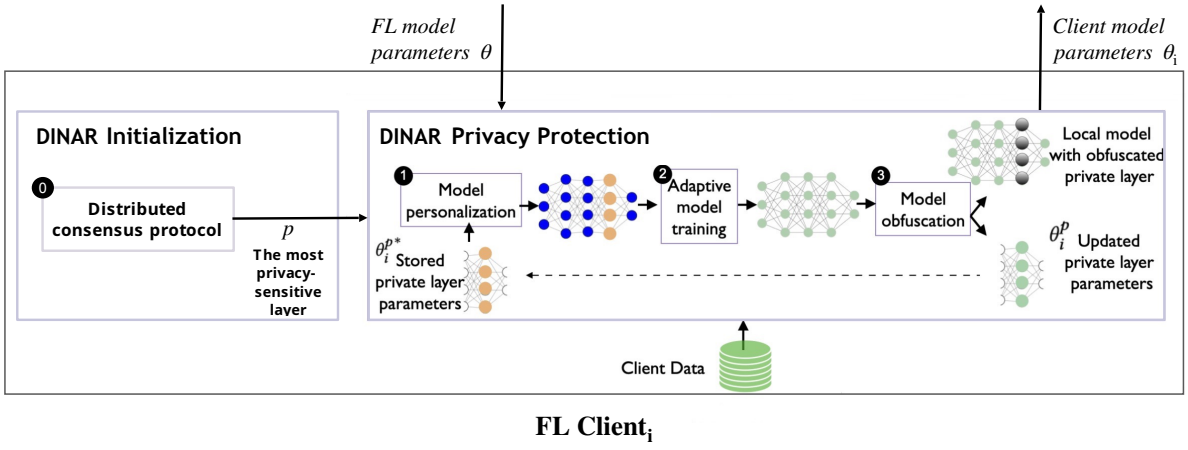


Figure 2: DINAR middleware

absolute majority, is chosen as the final index of the FL most privacy-sensitive layer to obfuscate. Based on experimental results that align with prior studies [29, 30], the algorithm typically converges to the penultimate layer of the model. At the end of DINAR Initialization phase, once the index p of the layer to be obfuscated is chosen by the consensus protocol, all clients (whether correct or not during DINAR Initialization phase) that are afterwards involved in the FL learning rounds will apply obfuscation on that same layer p of their local models.

4.2 Model Obfuscation

In the following, we consider a model θ with J layers, where $\theta^1 \dots \theta^J$ are the parameters of the respective layers $1 \dots J$. At each FL round, Client$_i$ that participates in that round updates its model parameters θ_i through local training. Before sending the local model updates to the FL server, the client obfuscates the privacy-sensitive layer of its model, namely θ_i^p that is the client model parameters of layer p. This obfuscation can be performed by simply replacing the actual value of θ_i^p by random values. The resulting local model updates are sent to the FL server for aggregation. Note that the raw parameters of the privacy-sensitive layer (*i.e.*, before obfuscation) are stored at the client side in θ_i^{p*}, and will be used in other stages of the DINAR pipeline.

4.3 Model Personalization

As a first step of DINAR pipeline for Client$_i$, it first receives the parameters of the global model θ. Here, θ^p, *i.e.*, the model parameters of the privacy-sensitive layer p, correspond to obfuscated values. Client$_i$ integrates to its local model parameters θ_i all global model layer parameters, except the parameters θ^p of the layer p. Instead, for that layer, the client restores θ_i^{p*}, its previously stored, non-obfuscated layer p parameters. Thus, while the global FL model is protected against MIAs, Client$_i$ makes use of an effective personalized local model. This approach contributes in maintaining good model utility. Client model's privacy-sensitive information remains protected, while client data still contributes to the overall improvement of the global model through collaborative training. Finally, with personalized FL, the resulting personalized client models are used by the clients for their predictions, whereas the global FL model is used for the overall learning process but not for predictions.

4.4 Efficient Adaptive Model Training

To overcome model convergence challenges, DINAR uses adaptive gradient descent to optimize the loss function $\mathcal{L}$ for each Client$_i$. This method updates model parameters θ_i at each local epoch with a learning rate $\eta \in [0, 1]$, effectively handling local minima and saddle points [6]. As described in Algorithm 1, lines 13-14, the optimizer first updates the cumulative errors G of the model. Then, model parameters θ_i are updated by adjusting them based on the gradient direction $\nabla_\theta \cdot$ loss, taking into account the accumulated sum of squared gradients G, which allows to apply an adaptive coefficient to the initial learning rate ∇ over time.

The use of adaptive gradient descent is motivated by its effectiveness in managing convergence with complex models such as Convolutional Neural Networks (CNNs). It generally exhibits a slower learning rate compared to algorithms such as Adam and RMSProp, particularly during the initial iterations [16, 32]. Furthermore, it dynamically adjusts the learning rate across different dimensions, similarly to Adam. However, Adagrad's lack of momentum can prevent client drift and worsen convergence issues in environments with increasing number of client participants and non-IID data distributions [14, 15].

In summary, DINAR provides a heuristics-based fine-grained privacy protection of FL neural networks. We acknowledge that a theoretical analysis of the proposed privacy-preserving FL solution is desirable. However, many recent works recognize that the formal quantification of the leakage of private information associated with the model layers and gradients is still a scientific conundrum [29, 30, 49].

5 EXPERIMENTAL EVALUATION

In the following, we first describe the various datasets (in §5.1) and baselines (in §5.2) we used in our experiments, as well as the underlying experimental setup (in §5.3). Then, we present analytical insights of the pertinence of DINAR (in §5.4), before describing the experimental results when evaluating privacy protection (in §5.5), its cost (in §5.6), its trade-off with utility (in §5.7), and the behavior of DINAR under different non-IID settings (in §5.8).

5.1 Datasets

We conduct experiments using a diverse set of datasets and models, encompassing four image datasets (Cifar-10, Cifar-100, CelebA, and GTSRB), two tabular dataset (Purchase100, Texas100), and a raw audio dataset (Speech Commands). For each dataset, half of the data is used as the attacker's prior knowledge to conduct MIAs [41], and the other half is partitioned into training (80%) and test (20%) sets. These datasets are summarized in Table 2, and detailed below.

Table 2: Used datasets and models

Dataset	#Records	#Features	#Classes	Data type	Model
Cifar-10	50,000	3,072	10	Images	ResNet20
Cifar-100	50,000	3,072	100	Images	ResNet20
GTSRB	51,389	6,912	43	Images	VGG11
CelebA	202,599	4,096	32	Images	VGG11
Speech Commands	64,727	16,000	36	Audio	M18
Purchase100	97,324	600	100	Tabular	6-layer FCNN
Texas100	67,330	6,170	100	Tabular	6-layer FCNN

CelebA. CelebFaces Attributes Dataset is a large face images dataset, with 202,599 images for facial recognition and attribute detection. A subset of 40,000 images, resized to 64x64 pixels, was randomly selected. We create 32 classes by combining five pre-annotated binary facial attributes (Male, Pale Skin, Eyeglasses, Chubby, Mouth slightly opened) for each picture [23]. The VGG11 architecture was employed for image processing [42].

Cifar-10 and Cifar-100. These are image datasets that consist of 60,000 images categorized into 10 classes for Cifar-10, and contains 100 classes for Cifar-100 [18]. These datasets encompass a wide range of objects such as airplanes, automobiles, birds, cats, and more. Each image in these datasets has a resolution of 32x32 pixels. For our experiments, we employ the ResNet-20 model.

GTSRB. German Traffic Sign Recognition Benchmark dataset comprises 51,389 records across 43 classes, specifically designed for traffic sign recognition. It captures real-world traffic scenarios, including variations in lighting, weather conditions, and camera angles. This dataset is widely used for evaluating traffic sign recognition algorithms and developing machine learning models for autonomous driving. We use VGG11 model architecture for this dataset [9, 42].

Purchase100. It is a tabular dataset adapted from Kaggle's "Acquire Valued Shoppers" challenge, consisting of 97,324 records with 600 binary features representing customer purchases. The goal was to classify customers into 100 types based on their buying behavior [41]. For classification, we use a fully-connected neural network architecture with layers of sizes 4096, 2048, 1024, 512, 256, and 128, leveraging Tanh activation functions and a fully-connected classification layer [13].

Speech Commands. This dataset is a Google-released audio waveform for speech recognition classification [45]. It consists of 64,727 utterances from 1,881 speakers pronouncing 35 words (respectively 35 classes). Each audio record was transformed into a frequency spectrum with a duration of 1 second. For classification, we use the M18 classifier, a convolutional model with 18 layers and 3.7M parameters [5].

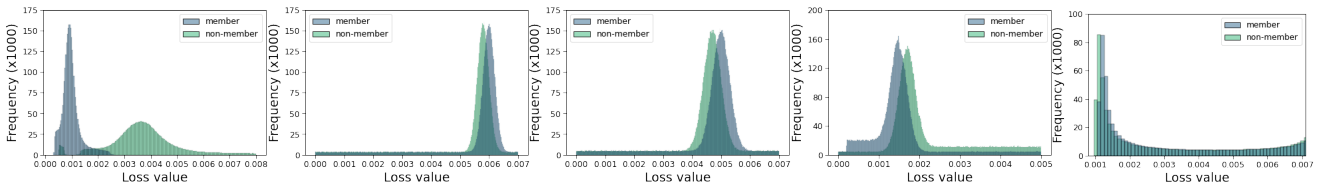
Texas100. This is a tabular dataset derived from the hospital discharge data published by the Texas Department of State Health Services [41]. It contains 67,330 records with 6,170 binary features representing patient information such as external causes of injury, diagnosis, procedures, hospital ID, and length of stay. The dataset's primary objective is to classify patient data into 100 classes based on the most frequent medical procedures. For classification, we use the same neural network model used for the Purchase100 dataset.

5.2 Baselines

Our evaluation compares DINAR with different defense scenarios, including five state-of-the-art solutions, as well as the no defense scenario. Thus, we consider LDP, CDP, and WDP state-of-the-art solutions that use differential privacy (DP). We also consider a cryptographic solution based on Secure Aggregation (SA) [54], and another defense solution based on Gradient Compression (GC) [7]. For LDP and CDP, we set the privacy budget parameter $\epsilon = 2.2$ and the probability of privacy leakage $\delta = 10^{-5}$, following the findings of [33]. In the case of WDP, a norm bound of 5 is considered, and Gaussian noise with a standard deviation of $\sigma = 0.025$ is applied. These settings ensure an optimal level of privacy preservation in our experiments.

5.3 Experiment Setup

The software prototype of DINAR is available in https://github.com/sara-bouchenak/DINAR/. All the experiments are conducted on an NVIDIA A40 GPU. We use PyTorch 1.13 to implement DINAR, and the underlying classification models. For the state-of-the-art defense mechanisms based on differential privacy, we employ the Opacus library [51]. In our experiments, we consider a FL system with 5 FL clients using Cifar-10, Cifar-100, GTSRB, CelebA and Speech Commands datasets, and 10 clients using Purchase100. The data are carefully divided into disjoint splits for each FL client. The number of FL rounds was chosen in such a way that the FL model reaches a stable state. That is, 50 FL rounds were necessary for Cifar-10, Cifar-100, GTSRB and CelebA, 80 FL rounds for Speech Commands, and 300 rounds for Purchase100. Each FL client performs 5 local epochs per round with all datasets, but Purchase100 that needed 10 local epochs. Each dataset is split into 80% for training, and 20% for testing. The learning rate is set to 10^{-3}, and the batch size is 64. We evaluate FL privacy-preserving methods by measuring the attack AUC, as well as the model accuracy, and several cost-related metrics, as described in Appendix A.

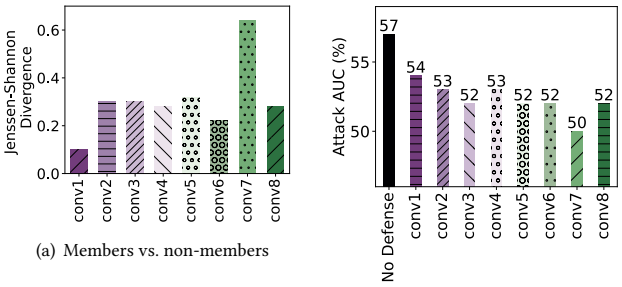


(a) No Defense (b) LDP (c) CDP (d) WDP (e) DINAR

Figure 3: Model loss distribution with different FL privacy-preserving techniques. The dark curve shows the loss distribution for member data samples, and the light curve shows the distribution for non-members of Cifar-10 dataset

5.4 Analytic Insights on the Pertinence of DINAR

In order to provide an insight on DINAR's ability to preserve both privacy and model utility, we analyze the impact of DINAR on the behavior of the protected model, and compare it to state-of-the-art solutions. First, in Figure 3, we measure the loss of the attacked model separately for member data samples that were used for training by clients, and non-member data samples, considering different defense methods. We evaluate the effectiveness of each defense technique in reducing loss distribution discrepancies between members and non-members, and in minimizing loss values. Ideally, the loss distribution of members and non-members should match, thus, resulting in model's lack of insightful information to distinguish members and non-members. Here, we observe that in the no defense case, the loss distributions between members and non-members are very different, thus, enabling successful MIAs. DP-based techniques (*i.e.*, LDP, CDP, WDP) reduce loss distribution discrepancies, however, at the expense of more frequent high loss values (*i.e.*, lower model utility) due to the noise added to all model layers' parameters. In contrast, a fine-grained obfuscation approach as followed by DINAR results in similar and more frequently low loss distributions of members and non-members, making MIAs more difficult and maintaining a good model quality.



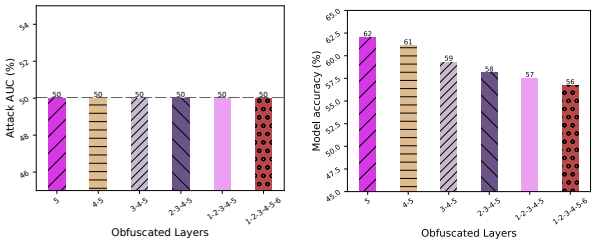
(a) Members vs. non-members

(b) Local model privacy

Figure 4: Analysis of fine-grained protection applied to each layer – CelebA dataset based on a neural network with 8 convolutional layers

Furthermore, we analyze the behavior of the fine-grained privacy protection approach of DINAR if it is applied more specifically to the one or the other of the layers of the neural network. Figure 4 puts into perspective two aspects of this analysis. On the one hand, Figure 4(a) shows how much one can determine the divergence between member data samples that were used for model training and non-member data samples, by analyzing the one or the other of model layers. On the other hand, Figure 4(b) presents the result of a fine-grained protection that obfuscates the one or the other of local model layers. We observe that obfuscating the layer that leaks more membership information is actually sufficient to reach the optimal protection of the overall client model against MIAs[†]. Whereas obfuscating other layers that leak less membership information is not sufficient for the protection of the overall client model. This is the basis of the heuristics provided by DINAR.

We also evaluate the impact of protecting more than one layer of the model, as presented in Figure 5. Here, obfuscating more layers does not improve model privacy, which is already optimal by protecting a single layer. On the other hand, the more layers are obfuscated, the more the utility of the model is negatively impacted. Note that a similar behavior is observed with other datasets and models, although not presented here due to space limitation.



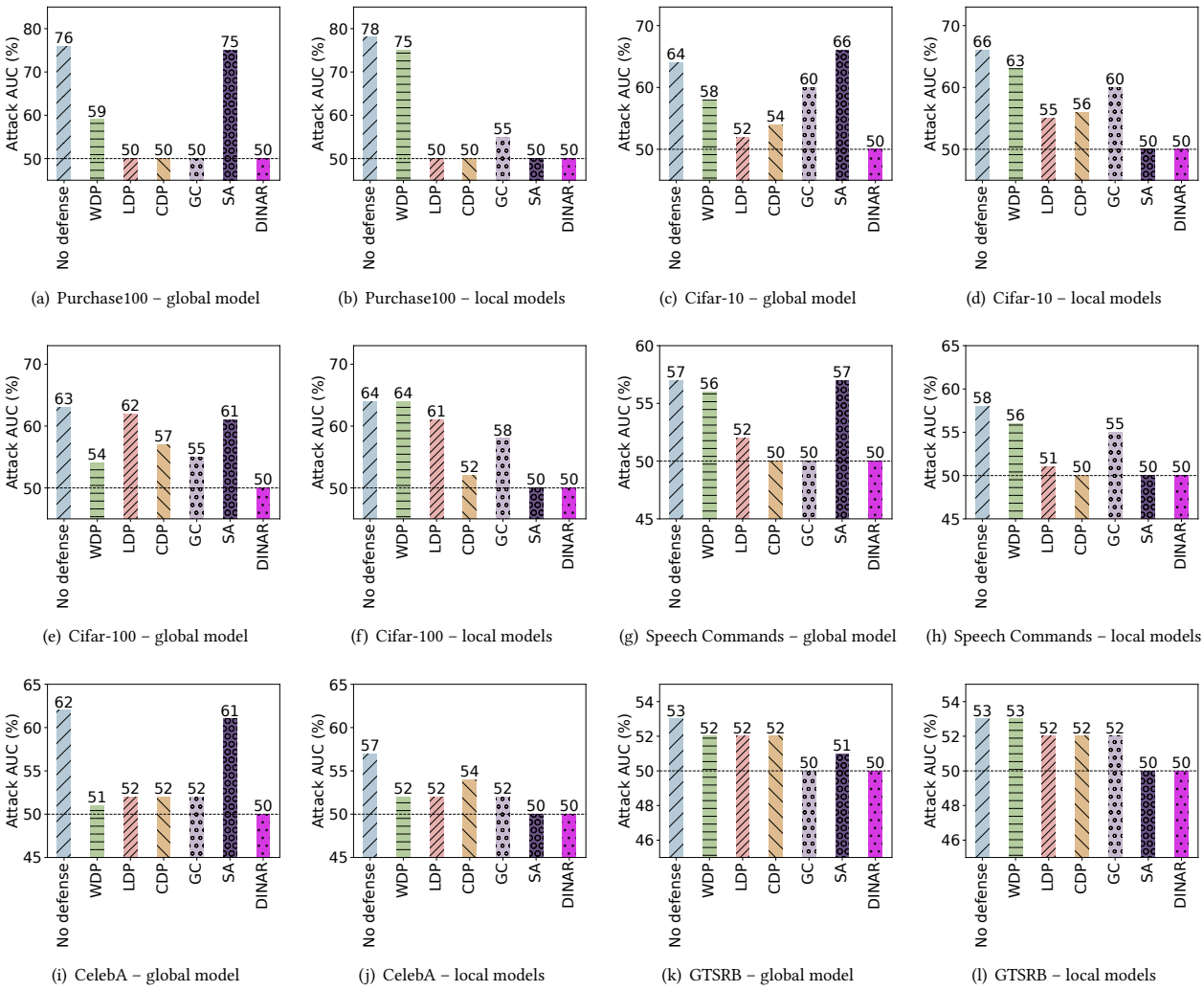
(a) Model privacy (b) Model utility

Figure 5: Impact of protecting more than one layer on local model privacy and utility – Purchase100 dataset based on a neural network with 6 layers

5.5 Evaluation of Privacy Protection

In the following, we evaluate the effectiveness of DINAR and other protection mechanisms in countering MIAs, *i.e.*, minimizing the attack AUC against both global FL model and clients' local models.

[†]50% is the optimal attack AUC that could be reached by a random protection approach, since determining the occurrence of a MIA is a binary decision.



(a) Purchase100 – global model (b) Purchase100 – local models (c) Cifar-10 – global model (d) Cifar-10 – local models

(e) Cifar-100 – global model (f) Cifar-100 – local models (g) Speech Commands – global model (h) Speech Commands – local models

(i) CelebA – global model (j) CelebA – local models (k) GTSRB – global model (l) GTSRB – local models

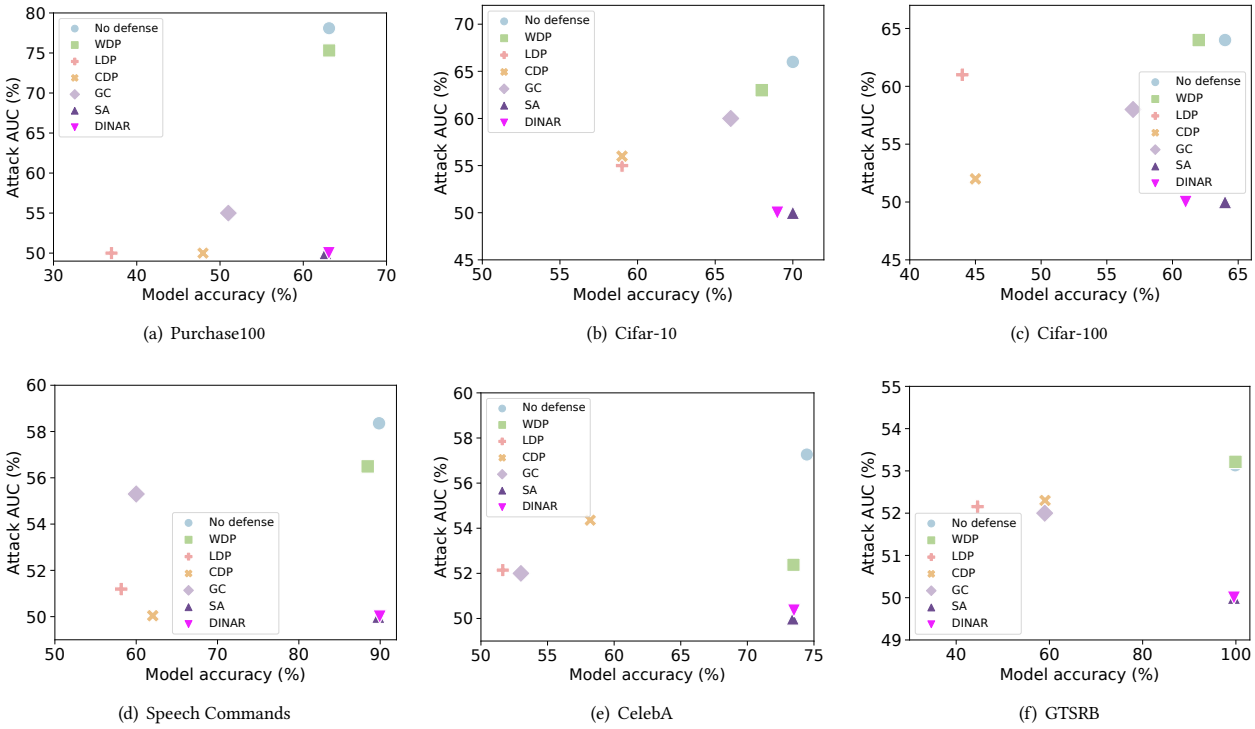
Figure 6: Privacy evaluation – The horizontal dashed line represents the optimal value of attack AUC (50%)

The attacker runs the MIA described in [41], on different datasets and their underlying models. Figure 6 presents the average attack AUC against global model and local models. The results show that DINAR exhibits privacy protection rates reaching the optimal value of 50% of attack AUC†, across all datasets, on both server and client sides, indicating a strong level of privacy protection. Differential privacy-based methods (*i.e.*, *WDP*, *LDP*, *CDP* are less consistent, often failing to mitigate attacks effectively. *LDP* and *CDP* reach 50% attack AUC for the Purchase100 dataset, but struggle with the remaining datasets. *GC* fails in most cases, while *SA* reduces the privacy leakage of local models to 50% but does not protect the global model.

5.6 Cost of Privacy-Preserving Mechanisms

We evaluate the overheads induced by DINAR and various privacy-preserving FL mechanisms on three key metrics including client-side model training duration, server-side model aggregation duration, and peak GPU memory usage for training and privacy protection. In Table 3, we compare the costs of the different defense mechanisms to the FL baseline using the GTSRB dataset with VGG11 model, although other evaluations were conducted with other datasets and models, resulting in similar observations.

The methods mitigating MIAs that operate on the client side to preserve privacy, including LDP, WDP, GC, and SA, increase client-side model training duration. We observe that differential privacy-based methods can significantly increase training duration. Despite Opacus framework improvements, there is still a significant cost. In the worst-case scenario, *WDP* increases training time by 35%. Similarly, GC increases it by 21% due to gradient compression operations, and SA by 21% due to client cryptographic operations.



(a) Purchase100 (b) Cifar-10 (c) Cifar-100

(d) Speech Commands (e) CelebA (f) GTSRB

Figure 7: Trade-off between privacy and utility for local models in different FL defense scenarios

Table 3: Overhead of FL defense mechanisms compared to FL baseline.

	Training Duration per FL round on client side (s)	Aggregation duration on server side (s)	GPU Memory usage on client side (Mb)
WDP	**+35%**	+0%	**+257%**
LDP	+7%	+0%	**+267%**
CDP	+0%	**+3000%**	**+261%**
GC	**+21%**	+0%	**+252%**
SA	**+21%**	+4%	+0%
DINAR (Ours)	*+0%*	*+0%*	*+0%*

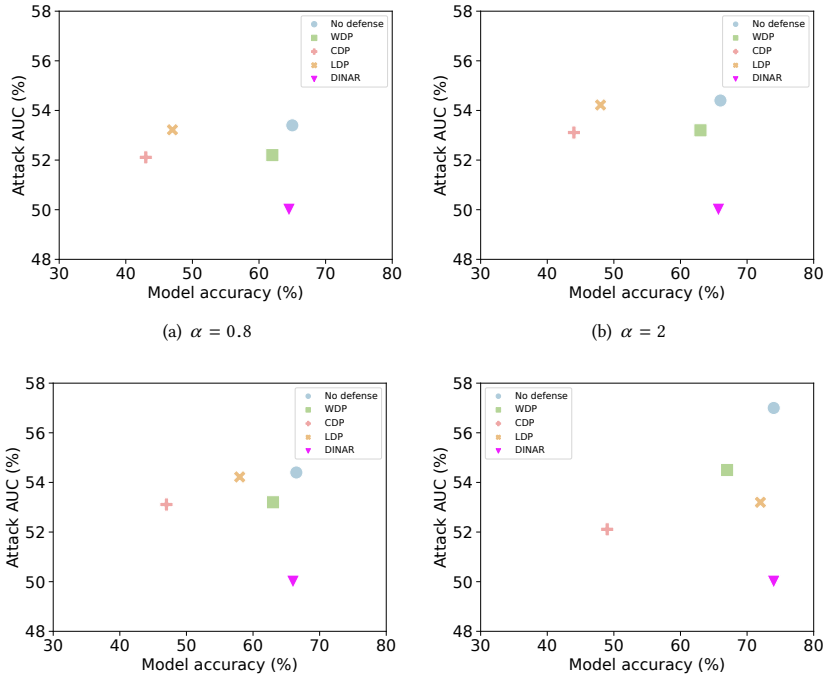
However, DINAR mitigates these overheads without compromising performance. For aggregation time, *CDP* increases duration by up to 30 times due to noise addition to the aggregated parameters before client transmission, while SA increases duration by 4% due to cryptographic operations on the server side. In contrast, DINAR, *LDP*, *WDP* and GC show similar aggregation times to the FL baseline, presenting a more efficient alternative.

Regarding GPU memory usage, differential privacy methods increase usage by 267% due to noise storage, privacy budget management, and aggregation buffer maintenance. GC increases memory usage by 252% due to storing the difference between original and compressed gradients. However, DINAR, which avoids noise addition and privacy budget management, has no significant impact on GPU memory usage. Overall, DINAR optimizes cost metrics, exhibiting performance similar to the FL baseline while preserving data privacy without compromising cost performances.

5.7 Analyzing Privacy vs. Utility Trade-off

With the objective of empirically confirming DINAR's ability to balance both privacy and model utility in a FL system, we conduct the experiments on different datasets. We run the same attack scenario as presented in §5.5, introducing both privacy and model utility metrics. Due to space limitations, we report the local models' results only.

Figure 7 shows our results by plotting both metrics on two axes: the x-axis represents the average local model accuracy, while the y-axis plots the overall attack AUC we previously defined. In a best-case scenario, the dot should be located in the bottom-right corner of each plot, meaning that the effective defense mechanism both preserves the model accuracy and decreases the attack AUC to 50%. We observe that *WDP*, *CDP* and *LDP* achieve reasonable attack mitigation but often reduce model utility. For example, on the Purchase100 dataset, *WDP* reduces attack AUC by 2%, while *CDP* reduces it by 28%; however, with a significant reduction of model accuracy by 20%. In contrast, DINAR reaches the optimal attack AUC, with a model accuracy drop lower than 1%. In most cases, DINAR strikes a balance between privacy preservation and utility, demonstrating the effectiveness of mitigating membership information leakage in a fine-grained FL approach.



(a) $\alpha = 0.8$ (b) $\alpha = 2$

(c) $\alpha = 5$ (d) $\alpha = \infty$ (i.e., IID)

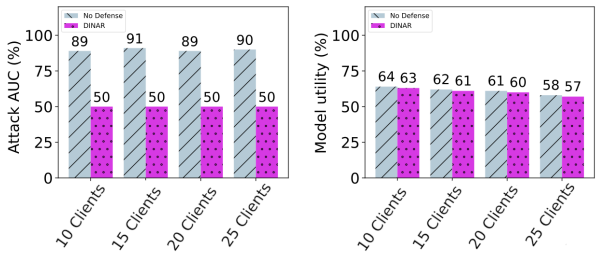
Figure 8: Privacy leakage vs. model utility under different non-IID FL settings – GTSRB dataset

5.8 Privacy Protection under Non-IID FL Settings

In the following, we consider different non-IID FL settings, and evaluate their impact on the actual privacy protection achieved by different protection methods. We vary the non-IID FL dataset distribution using the Dirichlet function [17] and its α parameter. The lower the Dirichlet's α value is, the more non-IID FL distribution is. Figure 8 presents the results of the evaluation of different non-IID distributions of the GTSRB dataset, and compares the utility and the resilience of clients' models to membership inference attacks when different privacy protection methods are applied, as well as when no defense is applied. Overall, for all cases except DINAR, the lower the non-IID distribution is, the higher the attack success rate is since the membership inference shadow model is able to better learn on such data. In the case of DINAR, the privacy protection is independent of the underlying non-IID setting and remains minimal at 50%. When it comes to model utility, obviously, the lower the non-IID distribution is, the higher the model utility is, although, DINAR reaches the highest model accuracy when protecting the model.

5.9 DINAR under Different Numbers of FL Clients

We evaluate the impact of varying numbers of FL clients on the actual performance of DINAR. Figure 9 reports the attack AUC and the accuracy of client models, with different numbers of the FL clients, comparing DINAR against the no defense baseline. In each case, the whole Purchase100 dataset was divided into subsets for the different FL clients. Obviously, the fewer the clients are, the higher the client model accuracy is, since fewer clients implies more data per client. However, and independently of the number of clients, DINAR is able to counter MIAs with an attack AUC of 50%.



(a) Model privacy (b) Model utility

Figure 9: Model privacy and model utility under different numbers of FL clients – Purchase100

5.10 Differential Privacy-Based Mechanisms with Different Budgets

We evaluate the resilience of LDP to MIAs with several differential privacy budgets [50]. We also compare LDP against DINAR, and the case where no defense is applied, as presented in Figure 10.

Obviously, small privacy budgets which apply higher noise provide better privacy. However, in order to reach the best privacy protection of 50%, LDP drastically degrades the model accuracy to 13%. Whereas, DINAR is able to keep a high model accuracy close to the no defense baseline, while effectively protecting against MIAs.

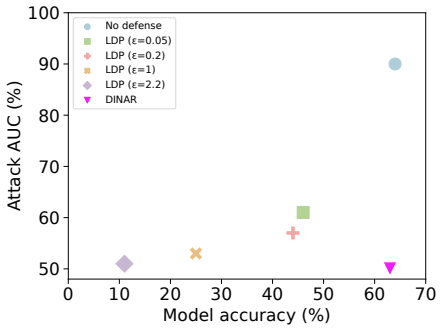


Figure 10: Privacy leakage vs. model utility under different DP budgets – Purchase100

5.11 Ablation Study

In order to evaluate the impact of adaptive learning in DINAR on the actual performance of the model, we conduct an ablation study where DINAR uses other state-of-the-art optimization techniques, such as Adam [16], ADGD [25], and AdaMax [16]. Figure 11 shows the effectiveness of model accuracy in DINAR. Furthermore, although not shown in the figure, all considered optimization techniques provide the same privacy protection level, *i.e.*, an attack AUC of 50%.

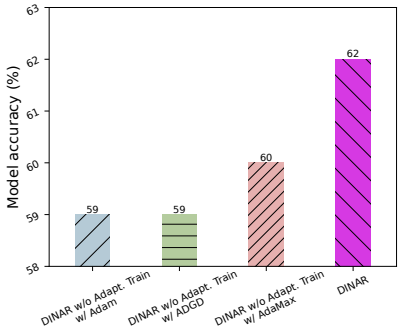


Figure 11: Ablation study – Comparing DINAR that uses adaptive training with variants of DINAR using other optimization techniques – Purchase100

6 CONCLUSION

We propose DINAR, a heuristic-based method to better protect the privacy of FL systems against membership inference attacks both for global FL model and client models.

DINAR follows a simple yet effective fine-grained approach that consists in protecting more specifically the model layer that is the most sensitive to membership privacy leakage. This provides effective and non-intrusive FL privacy protection. Furthermore, DINAR compensates for any potential loss in model accuracy through the use of personalized FL models and adaptive gradient descent, thereby maximizing model utility. We empirically evaluate the proposed method using various widely used datasets and different neural network models, comparing it to state-of-the-art FL privacy protection mechanisms. The results demonstrate the effectiveness of DINAR in terms of privacy, utility, and cost. Future research directions include investigating DINAR's resilience against other privacy threats, such as property inference attacks and model inversion attacks.

ACKNOWLEDGEMENTS

This work was partly supported by the French ANR project ByBloS (ANR-20-CE25-0002-01). Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER, and several universities as well as other organizations2.

A APPENDIX – DETAILED DESCRIPTION OF EVALUATION METRICS

Attack AUC. The attack success rate on a given model measures the percentage of successful MIAs conducted by an adversary. The attack AUC (Area Under the Curve) is a single value that measures the overall performance of the binary classifier implementing MIAs. The AUC value is within the range [50%–100%], where the minimum value represents the performance of a random MIA attacker, and the maximum value would correspond to a perfect attacker. The attack AUC is a robust overall measure to evaluate the performance of MIAs because its calculation involves all possible attacker's binary classification thresholds. Since the weakest (*i.e.*, most naive) MIA attacker would reach a minimum attack AUC of 50%, the best defense against MIAs would approach that optimal value of attack AUC of 50%. Thus, we use attack AUC as a means to evaluate the privacy of a model.

Overall Model Privacy Metric. In a FL system that consists of a global FL model θ and N client models $\theta_1, \ldots, \theta_N$, we define two privacy metrics. The first metric measures the privacy leakage from the global model θ, and the second metric assesses the local privacy from the client side by evaluating the average privacy leakage from all clients' local models. Given the function F_{AUC} for computing the attack AUC of a model, the local model privacy of the FL system is computed as follows:

$$\frac{\sum_{i=1}^{N} F_{AUC}(\theta_i)}{N}$$

Overall Model Utility Metric. We evaluate the utility of a protected model by measuring its accuracy, namely the ratio of correctly classified instances to the total number of instances. Considering DINAR's approach for protecting FL clients' models, we consider the average of accuracy of clients' protected models. Given N clients, θ_i the model of each Client$_i$, and F_{Acc} the function that calculates accuracy of a model, the overall model utility metric is

as follows:

$$\frac{\sum_{i=1}^{N} F_{Acc}(\theta_i)}{N}$$

Cost-Related Metrics. We also evaluate the additional costs that can be induced by a privacy-preserving FL mechanism, both in terms of execution time and memory usage. We measure the necessary time for a client to train a model during a FL round. We also measure the necessary time for the FL server to perform aggregation of client model updates. Finally, we measure the memory used by a client during model training.

REFERENCES

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

[2] Hamidreza Bandealinaeini and Saber Salehkaleybar. 2020. Broadcast Distributed Voting Algorithm in Population Protocols. IET Signal Processing 14, 10 (2020), 846–853.

[3] Mahawaga Arachchige Pathum Chamikara, Dongxi Liu, Seyit Camtepe, Surya Nepal, Marthie Grobler, Peter Bertók, and Ibrahim Khalil. 2022. Local Differential Privacy for Federated Learning. In Computer Security - ESORICS 2022 - 27th European Symposium on Research in Computer Security, Vol. 13554. Springer, Copenhagen, Denmark, 195–216.

[4] Zhenzhu Chen, Anmin Fu, Yinghui Zhang, Zhe Liu, Fanjian Zeng, and Robert H. Deng. 2021. Secure Collaborative Deep Learning Against GAN Attacks in the Internet of Things. IEEE Internet Things Journal 8, 7 (2021), 5839–5849.

[5] Wei Dai, Chia Dai, Shuhui Qu, Juncheng Li, and Samarjit Das. 2017. Very Deep Convolutional Neural Networks for Raw Waveforms. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, March 5-9, 2017. IEEE, New Orleans, LA, USA, 421–425.

[6] John Duchi and Elad Hazan. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. Journal of Machine Learning Research 12 (July 2011), 2121–2159.

[7] Chong Fu, Xuhong Zhang, Shouling Ji, Jinyin Chen, Jingzheng Wu, Shanqing Guo, Jun Zhou, Alex X. Liu, and Ting Wang. 2022. Label Inference Attacks Against Vertical Federated Learning. In 31st USENIX Security Symposium, USENIX Security 2022, August 10-12, 2022. USENIX Association, Boston, MA, USA, 1397–1414.

[8] Stefanie Grimm, Stefanie Schwaar, and Patrick Holzer. 2021. Federated Learning for Fraud Detection in Accounting and Auditing. ERCIM News 2021, 126 (2021), 30.

[9] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. 2013. Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark. In The 2013 International Joint Conference on Neural Networks, IJCNN 2013, August 4-9, 2013. IEEE, Dallas, TX, USA, 1–8.

[10] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. 2020. Measuring the Effects of Non-Identical Data Distribution for Federated Visual Classification. arXiv:2002.12062

[11] Chao Huang, Jianwei Huang, and Xin Liu. 2022. Cross-Silo Fderated Learning: Challenges and Opportunities,. arXiv:2206.12949

[12] Chao Huang, Ming Tang, Qian Ma, Jianwei Huang, and Xin Liu. 2023. Promoting Collaborations in Cross-Silo Federated Learning: Challenges and Opportunities. IEEE Communications Magazine (2023).

[13] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. 2019. MemGuard: Defending Against Black-Box Membership Inference Attacks via Adversarial Examples. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, November 11-15, 2019. ACM, London, UK, 259–274.

[14] Jiayin Jin, Jiaxiang Ren, Yang Zhou, Lingjuan Lyu, Ji Liu, and Dejing Dou. 2022. Accelerated Federated Learning with Decoupled Adaptive Optimization. In International Conference on Machine Learning, ICML 2022, 17-23 July 2022 (Proceedings of Machine Learning Research, Vol. 162). PMLR, Baltimore, Maryland, USA, 10298–10322.

[15] Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. 2020. Mime: Mimicking Centralized Stochastic Algorithms in Federated Learning. arXiv:2008.03606

[16] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In 3rd International Conference on Learning Representations, ICLR 2015, May 7-9, 2015, Conference Track Proceedings, Yoshua Bengio and Yann LeCun (Eds.). San Diego, CA, USA.

[17] Samuel Kotz, Narayanaswamy Balakrishnan, and Norman L Johnson. 2019. Continuous Multivariate Distributions, Volume 1: Models and Applications. Vol. 334. John Wiley & Sons.

[18] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. 2010. CIFAR-10 (Canadian Institute for Advanced Research). http://www.cs.toronto.edu/ kriz/cifar.html.

[19] Thomas Lebrun, Antoine Boutet, Jan Aalmoes, and Adrien Baud. 2022. MixNN: Protection of Federated Learning Against Inference Attacks by Mixing Neural Network Layers. In Middleware '22: 23rd International Middleware Conference, November 7 - 11, 2022. ACM, Quebec, QC, Canada, 135–147.

[20] Wenqi Li, Fausto Milletarì, Daguang Xu, Nicola Rieke, Jonny Hancox, Wentao Zhu, Maximilian Baust, Yan Cheng, Sébastien Ourselin, M. Jorge Cardoso, and Andrew Feng. 2019. Privacy-Preserving Federated Brain Tumour Segmentation. In Machine Learning in Medical Imaging - 10th International Workshop, MLMI 2019, Held in Conjunction with MICCAI 2019, October 13, 2019, Proceedings (Lecture Notes in Computer Science, Vol. 11861). Springer, Shenzhen, China, 133–141.

[21] Junxu Liu, Jian Lou, Li Xiong, Jinfei Liu, and Xiaofeng Meng. 2021. Projected Federated Averaging with Heterogeneous Differential Privacy. Proceedings of the VLDB Endowment 15, 4 (2021), 828–840.

[22] Ken Liu, Shengyuan Hu, Steven Z Wu, and Virginia Smith. 2022. On Privacy and Personalization in Cross-Silo Federated Learning. Advances in Neural Information Processing Systems 35 (2022), 5925–5940.

[23] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In 2015 IEEE International Conference on Computer Vision, ICCV 2015, December 7-13, 2015. IEEE Computer Society, Santiago, Chile, 3730–3738.

[24] Lingjuan Lyu, Han Yu, and Qiang Yang. 2020. Threats to Federated Learning: A Survey. arXiv:2003.02133

[25] Yura Malitsky and Konstantin Mishchenko. 2020. Adaptive Gradient Descent Without Descent. In Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119). Virtual Event, 6702–6712.

[26] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017 (Proceedings of Machine Learning Research, Vol. 54). PMLR, Fort Lauderdale, FL, USA, 1273–1282.

[27] M.L. Menéndez, J.A. Pardo, L. Pardo, and M.C. Pardo. 1997. The Jensen-Shannon Divergence. Journal of the Franklin Institute 334, 2 (1997), 307–318.

[28] Aghiles Ait Messaoud, Sonia Ben Mokhtar, Vlad Nitu, and Valerio Schiavoni. 2022. Shielding Federated Learning Systems Against Inference Attacks with ARM TrustZone. In Middleware '22: 23rd International Middleware Conference, Quebec, QC, Canada, November 7 - 11, 2022. ACM, QC, Canada, 335–348.

[29] Fan Mo, Anastasia Borovykh, Mohammad Malekzadeh, Hamed Haddadi, and Soteris Demetriou. 2021. Layer-Wise Characterization of Latent Information Leakage in Federated Learning. In The International Conference on Learning Representations (ICLR), Vol. Workshop on Distributed and Private Machine Learning. Virtual.

[30] Fan Mo, Anastasia Borovykh, Mohammad Malekzadeh, Hamed Haddadi, and Soteris Demetriou. 2021. Quantifying Information Leakage from Gradients. arXiv:2105.13929

[31] Fan Mo, Hamed Haddadi, Kleomenis Katevas, Eduard Marin, Diego Perino, and Nicolas Kourtellis. 2021. PPFL: Privacy-Preserving Federated Learning with Trusted Execution Environments. In MobiSys '21: The 19th Annual International Conference on Mobile Systems, Applications, and Services, 24 June - 2 July, 2021. ACM, Virtual Event, Wisconsin, USA, 94–108.

[32] Mahesh Chandra Mukkamala and Matthias Hein. 2017. Variants of RMSProp and Adagrad with Logarithmic Regret Bounds. In Proceedings of the 34th International Conference on Machine Learning, ICML 2017, 6-11 August 2017 (Proceedings of Machine Learning Research, Vol. 70). PMLR, Sydney, NSW, Australia, 2545–2553.

[33] Mohammad Naseri, Jamie Hayes, and Emiliano De Cristofaro. 2020. Toward Robustness and Privacy in Federated Learning: Experimenting with Local and Central Differential Privacy. arXiv:2009.03561

[34] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-Box Inference Attacks Against Centralized and Federated Learning. In 2019 IEEE Symposium on Security and Privacy, SP 2019, May 19-23, 2019. IEEE, San Francisco, CA, USA, 739–753.

[35] Jean Ogier du Terrail, Samy-Safwan Ayed, Edwige Cyffers, Felix Grimberg, Chaoyang He, Regis Loeb, Paul Mangold, Tanguy Marchand, Othmane Marfoq, Erum Mushtaq, et al. 2022. FLamby: Datasets and Benchmarks for Cross-Silo Federated Learning in Realistic Healthcare Settings. Advances in Neural Information Processing Systems 35 (2022), 5315–5334.

[36] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian J. Goodfellow, and Kunal Talwar. 2017. Semi-Supervised Knowledge Transfer for Deep Learning from Private Training Data. arXiv:1610.05755

[37] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. 2018. Scalable Private Learning with PATE. arXiv:1802.08908

[38] Tye Rattenbury, Joseph M. Hellerstein, Jeffrey Heer, Sean Kandel, and Connor Carreras. 2017. Principles of Data Wrangling: Practical Techniques for Data Preparation. O'Reilly Media, London.

[39] Saber Salehkaleybar, Arsalan Sharif-Nassab, and S Jamaloddin Golestani. 2015. Distributed Voting/Ranking with Optimal Number of States per Node. IEEE Transactions on Signal and Information Processing over Networks 1, 4 (2015), 259–267.

[40] Yifan Shi, Yingqi Liu, Kang Wei, Li Shen, Xueqian Wang, and Dacheng Tao. 2023. Make Landscape Flatter in Differentially Private Federated Learning. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, June 17-24, 2023. IEEE, Vancouver, BC, Canada, 24552–24562.

[41] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership Inference Attacks Against Machine Learning Models. In 2017 IEEE Symposium on Security and Privacy, SP 2017, May 22-26, 2017. IEEE Computer Society, San Jose, CA, USA, 3–18.

[42] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In 3rd International Conference on Learning Representations, ICLR 2015, May 7-9, 2015, Conference Track Proceedings. San Diego, CA, USA.

[43] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H. Brendan McMahan. 2019. Can You Really Backdoor Federated Learning? arXiv:1911.07963

[44] Aleksei Triastcyn and Boi Faltings. 2020. Federated Generative Privacy. IEEE Intelligent Systems 35, 4 (2020), 50–57.

[45] Pete Warden. 2018. Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition. arXiv:1804.03209

[46] Di Wu, Saiyu Qi, Yong Qi, Qian Li, Bowen Cai, Qi Guo, and Jingxian Cheng. 2023. Understanding and Defending Against White-Box Membership Inference Attack in Deep Learning. Knowledge-Based Systems 259 (2023), 110014.

[47] Qiong Wu, Xu Chen, Zhi Zhou, and Junshan Zhang. 2022. FedHome: Cloud-Edge Based Personalized Federated Learning for In-Home Health Monitoring. IEEE Transactions on Mobile Computing 21, 8 (2022), 2818–2832.

[48] Runhua Xu, Nathalie Baracaldo, Yi Zhou, Ali Anwar, and Heiko Ludwig. 2019. HybridAlpha: An Efficient Approach for Privacy-Preserving Federated Learning. In Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2019, November 15, 2019. ACM, London, UK, 13–23.

[49] Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. 2020. A Theory of Usable Information Under Computational Constraints. arXiv:2002.10689

[50] Yuchen Yang, Bo Hui, Haolin Yuan, Neil Zhenqiang Gong, and Yinzhi Cao. 2023. PrivateFL: Accurate, Differentially Private Federated Learning via Personalized Data Transformation. In 32nd USENIX Security Symposium, USENIX Security 2023, August 9-11, 2023. USENIX Association, Anaheim, CA, USA, 1595–1612.

[51] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. 2021. Opacus: User-Friendly Differential Privacy Library in PyTorch. arXiv:2109.12298

[52] Jiale Zhang, Bing Chen, Shui Yu, and Hai Deng. 2019. PEFL: A Privacy-Enhanced Federated Learning Scheme for Big Data Analytics. In 2019 IEEE Global Communications Conference, GLOBECOM 2019, December 9-13, 2019. IEEE, Waikoloa, HI, USA, 1–6.

[53] Ning Zhang, Qian Ma, and Xu Chen. 2023. Enabling Long-Term Cooperation in Cross-Silo Federated Learning: A Repeated Game Perspective. IEEE Transactions on Mobile Computing 22, 7 (2023), 3910–3924.

[54] Yifeng Zheng, Shangqi Lai, Yi Liu, Xingliang Yuan, Xun Yi, and Cong Wang. 2022. Aggregation Service for Federated Learning: An Efficient, Secure, and More Resilient Realization. IEEE Transactions on Dependable and Secure Computing 20, 2 (2022), 988–1001.