École Polytechnique Fédérale de Lausanne

Semester Project Report

Privacy Risks in Personalized Decentralized Learning

presented by

Fanny Lasne

Supervisors:

Prof. Dr. Anne-Marie Kermarrec

Dr. Sayan Biswas

Dr. Martijn de Vos

Milos Vujasinovic

June 2025

"It always seems impossible until it's done."

– NELSON MANDELA

Contents

Abstract

Decentralized Learning (DL) allows nodes to collaboratively train machine learning models without sharing raw data, making it appealing for privacy-sensitive applications such as healthcare. However, DL faces challenges in the presence of feature heterogeneity in the training data, which can degrade model performance and fairness, especially for nodes with under-represented training data. To address this, FACADE—a clustering-based DL algorithm—enables fair and personalized model training by grouping nodes with similar feature distributions into latent clusters. While FACADE improves fairness, the personalized nature of the models and the exchange of model updates between neighbors raise new privacy concerns.

In this project, I investigate the privacy vulnerabilities of FACADE by evaluating its susceptibility to loss-based Membership Inference Attacks (MIA)–a type of attack that aim to determine whether a given data sample was used to train a neighboring node's model. Using the FACADE framework, I simulate a decentralized environment with two heterogeneous clusters and implement MIA targeting shared model updates. My experiments show that increased model personalization and cluster imbalance significantly raise the success of inference attacks, suggesting that improving fairness through personalization can come at the cost of privacy. These results underscore the importance of considering privacy-fairness trade-offs in the design of DL systems.

Introduction

Decentralized learning (DL) has emerged as a promising approach for collaboratively training machine learning (ML) models in scenarios where data privacy is a critical concern. Unlike centralized learning, DL allows nodes to train models locally and exchange only model updates with neighbors, avoiding direct sharing of private data [1]. This makes DL particularly suitable for sensitive domains such as healthcare and finance, where data confidentiality is essential [2].

Despite these privacy advantages, DL faces challenges in the presence of data heterogeneity, particularly feature heterogeneity, where different nodes observe data with different feature distributions. This non-IID (non-independent and identically distributed) setting can lead to poor generalization and unfair performance. For example, in healthcare, Electronic Health Records (EHRs) vary widely across hospitals due to differences in patient demographics, treatment protocols, and data collection practices. Hospitals serving minority groups—such as specific age cohorts or patients receiving rare treatments—are often underrepresented in training data. As a result, models may fail to capture relevant patterns for these populations, leading to degraded performance and fairness, and potentially exacerbating healthcare disparities.

To address this, FACADE [3], a recent clustering-based DL framework, was designed to enable fair model training under heterogeneous feature distributions. It assigns nodes to feature-specific clusters based on similarities in their local data. Within each cluster, nodes collaboratively train personalized models that better reflect their data characteristics, thereby improving both performance

and fairness across the network.

Yet, personalization may introduce new privacy risks. Although local data is not shared, model updates—now trained on highly specific data—can leak sensitive information. One such risk arises from Membership Inference Attacks (MIAs), where an adversary aims to infer whether a particular data sample was part of a target model's training set. While MIAs have been widely studied in centralized and federated learning [4], their impact on clustered decentralized learning remains largely unexplored.

In this project, I explore the vulnerability of FACADE to loss-based MIAs, where adversaries exploit differences in prediction loss between training members and non-members. Specifically, I simulate a DL environment with two heterogeneous clusters and assess MIA effectiveness under various experimental settings. This analysis aims to shed light on the trade-off between privacy and fairness: while personalization can improve fairness, it may also increase the risk of data leakage; conversely, stronger privacy protections might come at the cost of reduced fairness. My contributions are the following:

Contrib. 1 Formalization of the threat model for loss-based MIAs in the context of FACADE's architecture.

Contrib. 2 Implementation of a loss-based MIA targeting model updates shared between nodes.

Contrib. 3 Evaluation of the attack's effectiveness under various conditions, including changes in cluster configurations, model head sizes, and non-member selection

My results provide insight into the hidden privacy costs of personalization in DL and motivate future work toward fair privacy-preserving DL frameworks.

Related Work

Collaborative Learning Machine learning has evolved from *centralized* methods, where both data aggregation and model training take place on a central server, to more *collaborative* approaches that preserve data locality [5][2]. Federated learning (FL), introduced by Google [6], is probably the most popular one. It enables clients to train models locally and share only model updates with a central server for aggregation, thus preserving privacy while leveraging diverse, distributed datasets. In contrast, decentralized learning (DL) operates without a central server. Nodes train models locally on their own data and exchange updates directly with neighboring nodes. By continuously aggregating these updates through peer-to-peer communication, the network collectively converges to a global model [1][7]. This method enhances robustness–avoiding single points of failure– and scales more efficiently than FL. FACADE [3] falls into this category of collaborative learning algorithms, addressing DL fairness concerns arising from feature heterogeneity, by leveraging clustering based on feature skew and training specialized models tailored to each cluster.

Privacy Attacks As ML systems proliferate, assessing their privacy risks has become critical. Privacy attacks vary in adversary knowledge and capabilities, but all aim to extract secret information such as training data or model details [8]. A key area of study is Membership Inference Attacks (MIAs), which aim to infer whether a particular data sample was part of a model's training set. Shokri et al. [4] introduced the first MIA against black-box models, using "shadow models" trained on synthetic datasets to mimic a target model's be-

havior. While this work focused on centralized ML, many subsequent studies extended MIAs to collaborative learning, where adversaries exploit partial information like model updates. Most of these, however, concentrate on FL [9, 10], with relatively few examining MIAs in DL. Pasquini et al. [11] provided the first in-depth privacy analysis of DL, showing it does not inherently improve privacy but can increase exposure. They identify two main sources of leakage: local generalization (models overfitting local data) and system knowledge (users directly observing raw peers updates). These enable passive adversaries to perform effective MIAs and even gradient inversion with higher success than in FL. Notably, DL users can gain inference power comparable to a central server in FL, increasing the number of potential adversaries. Their findings suggest that without fully connected topologies and secure aggregation–which reduce DL's advantages–DL may offer weaker privacy guarantees than FL. Unlike these broader analyses, my project focuses on FACADE DL algorithm and implements a distinct loss-based MIA approach, setting it apart from previous methods[4, 11].

Privacy-Preserving DL Several techniques have been proposed to mitigate privacy leakage in ML. Secure aggregation [12] protocols enable collective model updates without exposing individual contributions. Additionally, mechanisms satisfying *differential privacy* [13] can be applied, typically by adding noise to model updates before sharing. However, in DL, these techniques often come with trade-offs in model performance or utility. To address this, SHATTER [14] proposes a noiseless, virtualization-based DL framework where each real node (RN) runs multiple virtual nodes (VNs). These VNs randomly exchange subsets of model parameters (chunks) with other VNs across the network. This randomized, partial communication maintains convergence while making it much harder for adversarial RNs to infer sensitive information. Empirical results show strong robustness to three standard privacy attacks, including MIAs, which become infeasible with 16 VNs per RN. This project further aims to position FACADE in relation to SHATTER with respect to susceptibility to MIA.

Preliminaries

Before presenting the main contributions, I review Facade's design, define the privacy definition considered, and outline the chosen attack strategy.

4.1 FACADE

FACADE [3] is a DL algorithm designed to ensure fair model training in settings where training data exhibits distinct subpopulation features, which can introduce biases. It considers *majority* and *minority* clusters, defined by a sensitive attribute S distinguishing population subgroups (such as gender or ethnicity). Its goal is to ensure that model's predictions $\hat{y}$ are independent of group membership and achieve equal true/false positive rates across these groups [1].

The core idea of FACADE is to equip each node with k model heads—one per cluster—alongside a shared model core. In practice, the model heads are the final few layers of the neural network. Nodes receive model components (core and heads) from their neighbors, aggregates the cores with their own, and perform cluster-wise aggregation of the heads. This produces k distinct models, each specialized for a cluster. Nodes then evaluate these models on their local data and select the one with the lowest loss—implicitly determining their cluster assignment. They finally update the chosen model using SGD and broadcast it to their neighbors along with their corresponding cluster ID. Through this dynamic selection of heads, clustering naturally emerges, allowing nodes to progressively align with similar peers as the models become more specialized.

[1] i.e., ensure demographic parity and equalized odds

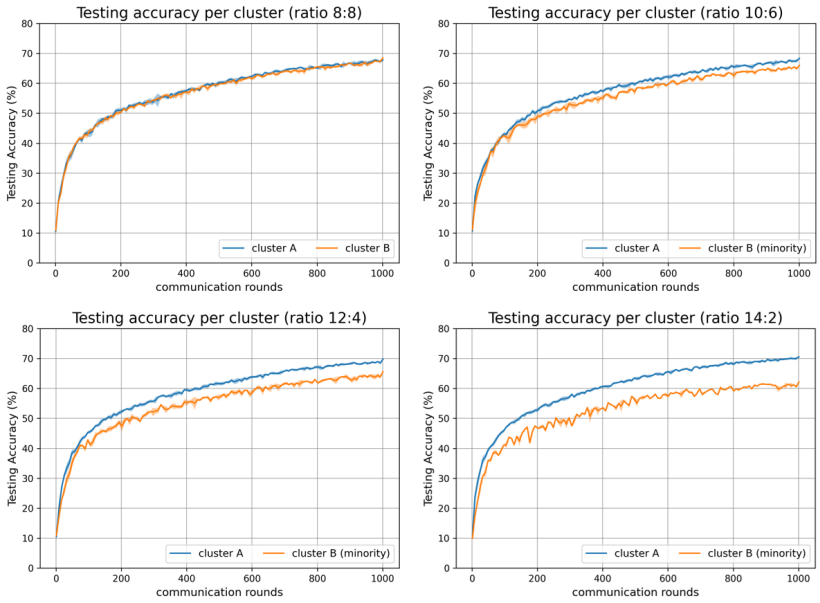


Figure 4.1: FACADE average test accuracy by cluster obtained on CIFAR-10, for different cluster ratios

As illustrated in Figure 4.1, experiments on the CIFAR-10 dataset demonstrate that FACADE maintains relatively high accuracy for both majority and minority clusters, even as cluster imbalance increases.

4.2 Privacy

The notion of privacy considered in this work is *differential privacy* (DP) [13] which, provides a formal framework to protect the privacy of individuals in a database–the training dataset in ML contexts. An algorithm is said to satisfy DP if it generates similar outputs on any two neighboring datasets, i.e., datasets differing by a single data point.

This notion can be framed as a statistical hypothesis testing problem. Let $\mathcal{A}$ be an adversary attempting to determine whether a particular data point x was used to train a model. $\mathcal{A}$ performs the following hypothesis test:

- H_0: $x \notin D$ — the point was not in the training set.
- H_1: $x \in D$ — the point was in the training set.

These hypotheses correspond to neighboring datasets D and D', where one

contains x and the other does not. If $\mathcal{A}$ can effectively distinguish between the model outputs $\mathcal{M}(D)$ and $\mathcal{M}(D')$, then x's membership to the training set is revealed, and DP is violated.

In this project, I investigates this privacy risk empirically in FACADE by conducting MIAs which, as describe in Section 3, directly implement the hypothesis test described above. The stronger the adversary's ability to distinguish between H_0 and H_1, the further the model deviates from DP.

4.3 Loss-based MIA

To evaluate the privacy of FACADE, I used a membership inference attack, which serves as a widely accepted benchmark for privacy assessment due to its close relationship with many other privacy attacks [11]. Specifically, I implemented a *loss-based* MIA following the approach proposed in SHATTER [14], chosen for its simplicity and demonstrated empirical effectiveness. Since this work presents the first privacy analysis of FACADE, starting with a well-understood and lightweight method offers a practical and informative baseline.

Loss-based MIA relies on the observation that models typically produce lower loss values on training samples (members) than on unseen data (non-members), allowing loss magnitude to serve as a proxy for membership. The adversary queries the target model with input samples to obtain their loss values, and uses the negative loss values as confidence scores. Membership predictions are made by comparing these scores against a threshold τ: samples above τ are classified as members, others as non-members. To evaluate the attack's effectiveness without relying on a fixed threshold, it is common to sweep τ across all possible values to compute the true and false positive rates, which are then used to plot the ROC[2] curve. The *area under the ROC curve* (ROC-AUC) quantifies the adversary's success: an AUC of 0.5 corresponds to random guessing, while an AUC of 1.0 indicates a perfect attack.

[2]Receiver Operating Characteristic

Membership Inference Attack on FACADE

In this section, I present a practical implementation of a membership inference attack against FACADE, providing a first empirical assessment of its privacy leakage. Since FACADE's privacy properties have not been previously studied, this evaluation offers an initial insight into its robustness against MIA and lays the groundwork for future privacy analyses.

5.1 Threat Model

FACADE is designed to operate within a *permissioned network*, where participation is restricted to authenticated and verified entities.

We consider an *Honest-but-Curious* (HbC) adversarial model, in which adversaries strictly follow the prescribed FACADE protocol but aim to extract as much information as possible from the data they legitimately receive [15]. The goal of such adversaries is to infer sensitive information about the private training data of one or more honest participants in the system, refered to as victims or targets.

In FACADE network, any node participating in the protocol can act as an adversary, since each node observes model updates from its neighbors at each communication round. These adversaries are non-colluding and are limited to attacking only their direct neighbors–regardless of cluster affiliation. We assume that aggregated models are neither publicly shared nor leaked at any point during the protocol's execution.

5.2 FACADE Framework Overview

Given that the membership inference attack implemented in this project operates at training time, exploiting the model updates exchanged between nodes, I first outline the structure of the FACADE framework, to set up the context in which the attack takes place.

Components FACADE's network consists of n **nodes** $N_1, \ldots, N_n$, each holding a local dataset Z_i drawn from a global data space $\mathcal{Z}$. The data across nodes is generated from k distinct distributions $\mathcal{D}_1, \ldots, \mathcal{D}_k$, with each node's dataset drawn from exactly one of these distributions. Nodes sharing the same data distribution $\mathcal{D}_j$ form cluster j. As detailed in Section 4.1, each node maintains k **models** sharing a common core.

Interactions Nodes communicate through a dynamic, randomized **communication topology** that is refreshed every round to ensure effective mixing of model updates. In each round, nodes receive model updates from their current neighbors–defined by the topology–and apply an **aggregation mechanism** to combine these with their local models. They then update their cluster assignment and perform local training on the model corresponding to their newly assigned cluster using their local dataset. (cf. Section 4.1)

5.3 Attack Implementation

The primary modifications enabling the attack in the original codebase involve adapting the averaging method within the `CurrentModelSharing` class to implement the procedure outlined in Algorithm 1–alongside several minor adjustments were made to utility methods across other classes to ensure seamless integration with the existing framework. A second key addition is the introduction of a dedicated `LossMIA` class, which encapsulates the implementation logic of the membership inference attack, described in Algorithm 2.

The set of victim nodes $\mathcal{V}$ is initialized once at the start of training, with an

equal number of victims randomly selected from each cluster to ensure balanced representation. The MIA attack is triggered every m communication rounds and executed by any node receiving a model update from at least one node in $\mathcal{V}$. This guarantees that each victim is attacked every m round, since all nodes have at least one neighbor. Attackers are not fixed, as the dynamic and randomized nature of the topology would produce inconsistent attack coverage, compromising the reliability and interpretability of the attack results.

The attack is performed after receiving neighbors' model updates but before local aggregation (at local step (b) in 1), to preserves the individual characteristics of each model update. This enables a more accurate attribution of model behavior to specific training data, thereby enhancing the effectiveness of membership inference. Aggregation would otherwise obscure these distinctions, limiting the attack's efficiency.

Algorithm 1: MIA integration into FACADE's training algorithm

This attack assumes a fixed set $\mathcal{V}$ of victim nodes throughout the training process, with an equal distribution of victims across all clusters.

Round $0 \leq t \leq T$ in FACADE consists of the following steps:

  Refresh randomized communication topology $\mathcal{G}$.

  Local steps. For all $i \in [n]$:

    (a) *Receive models.* N_i receives models and the corresponding cluster IDs from each of its neighbors.

    (b) **If** $t \mod m = 0$:

      For all model update θ_c received from neighbor c:

        If $c \in \mathcal{V}$:

          target_model $\leftarrow \theta_c$

          mb_set $\leftarrow c$'s training data

          non_mb_set $\leftarrow$ attack test set ▷ more details in 6.1

          result = LOSSMIA(target_model, mb_set, non_mb_set)

          store result

    (c) *Aggregation.* N_i aggregates the cores and performs a cluster-wise aggregation of the heads to obtain aggregated models for each cluster.

    (d) *Cluster identification.* N_i obtains the cluster ID corresponding to the model θ_i^* that gives the least loss on its local data.

    (e) *Local training.* N_i performs the SGD steps on θ_i^* to get a new locally optimized model θ_i

  Communication. For all $i \in [n]$: N_i shares θ_i and its corresponding cluster ID with its neighbors.

Algorithm 2: LossMIA

Input: $target\_model, \mathcal{D}_{in}, \mathcal{D}_{out}$
Output: A dictionary containing the negative training loss values for both "in" and "out" training samples

Set $target\_model$ to evaluation mode
$loss_{in} \leftarrow \text{zeros}(|\mathcal{D}_{in}|)$
$loss_{out} \leftarrow \text{zeros}(|\mathcal{D}_{out}|)$
  For $data\_samples$ in $\mathcal{D}_{in}$
    $losses \leftarrow \text{ModelEval}(target\_model, data\_samples)$
    $loss_{in}.\text{append}(-losses)$
  For $data\_samples$ in $\mathcal{D}_{out}$
    $losses \leftarrow \text{ModelEval}(target\_model, data\_samples)$
    $loss_{out}.\text{append}(-losses)$
return $\{\text{in} : loss_{in}, \text{out} : loss_{out}\}$

Algorithm 2 implements a loss-based MIA by evaluating the target model's behavior on given known training member ($\mathcal{D}_{in}$) and non-member data samples ($\mathcal{D}_{out}$). For each batch from both datasets, the per-sample loss is computed via the ModelEval routine, which simply computes the element-wise cross-entropy loss between the model's predictions and the true labels for a given input batch, returning the loss values for each individual sample. Negative loss values are recorded in separate arrays–one for members and one for non-members–so that they can be used after training to assess the attack performance as described in Section 4.3.

Privacy Evaluation of FACADE

This section presents the experiments conducted to evaluate the vulnerability of FACADE to the attack introduced in Section 5, under different conditions.

6.1 Experimental Design

The MIA was implemented using the publicly available FACADE codebase [3], built on top of the DECENTRALIZEDPY framework [7].

Dataset I conduct experiments on the CIFAR-10 dataset [16], following the evaluation setup used in FACADE. It contains 60,000 32x32 color images in 10 classes, with 6,000 images per class. There are 50,000 training and 10,000 test samples. To simulate feature heterogeneity without label skewness—as done in FACADE—the dataset is first uniformly partitioned into smaller subsets. Feature skew is then added by applying distinct random rotations to each cluster, ensuring that feature distributions differ while label distribution is maintained.

Network and Cluster Configurations All experiments were run on a single machine using 16 processes, each simulating a node in a regular graph topology of degree 4. The nodes were divided into two clusters, and while the total number of nodes remained fixed, cluster sizes were varied to study how imbalance affects privacy—as FACADE's fairness guarantee suggests the *minority* cluster may be more vulnerable. Three cluster configurations were tested: 8:8 (balanced), 12:4, and 14:2. For example, in the latter, 14 nodes use upright images, while 2 nodes have images rotated by 180° to induce feature heterogeneity.

Model GN-LeNet was used [17]. It has about 120k parameters, consisting of three convolution layers and one feed-forward layer. Initially, the attack was performed on models trained with FACADE, using the last fully connected layer as the head and the rest as the shared core. In a second stage, the head size was increased to include two layers (last convolution + feed-forward) and eventually the entire model to explore the total personalization case.

Attack parameters Each experiment ran for $T = 1000$ communication rounds, with attackers initiating MIAs every 32 rounds on a fixed set of victims. We chose 2 victim nodes per cluster to support all configurations (e.g., three victims is not feasible under a 14:2 ratio). In Algorithm 1, the member set is simply the victim's training data. On the other hand, three non-member sets were tested: the *majority* cluster's test set, the *minority* cluster's test set, and their *union*. This will lets us quantify how non-member data composition—combined with cluster size imbalance—impacts the vulnerability of each cluster's data.

Attack Success Metric The effectiveness of the attack is evaluated using the true positive rate (TPR) and the false positive rate (FPR). As discussed in Section 4.3, the attack performance is quantified using the *ROC-AUC metric*, which represents the area under the curve when plotting TPR against FPR.

6.2 Results and Analysis

6.2.1 Cluster Ratio & Non-Member data

I first evaluate how the the performance of MIA evolves over the course of FACADE's training, considering different cluster configurations and attack settings—specifically, the choice of non-member sets used by the attackers. Figure 6.1 illustrates the success of the MIA for each cluster ratio (represented by color), each cluster (indicated by line style), and each type of non-member set. In the following, I discuss the results shown in the three subplots, proceeding from left to right.

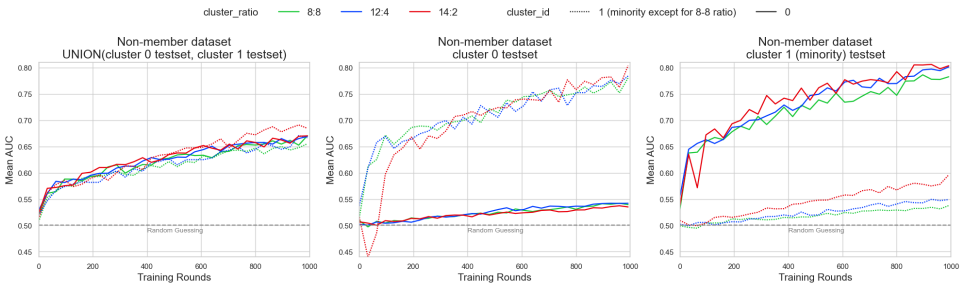


Figure 6.1: Impact of *cluster ratio* on MIA performance across training rounds for different non-member sets (cf. 1) with fix head size (1: the last NN-layer)

The leftmost plot shows the attack performance when the non-member set is composed of unseen samples **drawn from both clusters' feature distributions**. After 400 training rounds, the mean AUC exceeds 0.6 *across all configurations*, indicating that attackers begin to be threatening relatively early (before the algorithm converges). AUC continues to rise steadily for both clusters as training proceeds. Greater class imbalance amplifies attack success overall, with the minority cluster eventually reaching higher AUCs than the majority cluster, especially in the 14:2 setting. However, the difference in attack success remains negligible, even under strong imbalance, with a maximum AUC gap of approximately 0.0207 between cluster 0 and cluster 1.

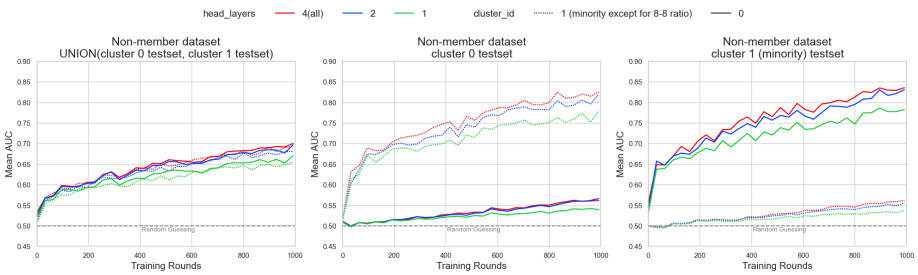
The middle plot reveals a marked disparity in attack effectiveness between clusters when non-member data samples are **only drawn from cluster 0** (majority) feature distribution. While AUC values for cluster 0 remain low ($\approx$ 0.54 for all ratios), those for cluster 1 are much higher—ranging between 0.77 and 0.80. This indicates that the attack is substantially more effective against cluster 1 (minority) members. Cluster imbalance has minimal impact on attack success for cluster 0, and only a modest effect for cluster 1, with an AUC increase of $\approx$ 0.02 between the balanced (8:8) and imbalanced (14:2) settings.

The rightmost plot exhibits a pattern similar to the middle panel but with reversed roles. When non-member samples are **only drawn from cluster 1** (minority) feature distribution, attack performance is significantly higher for

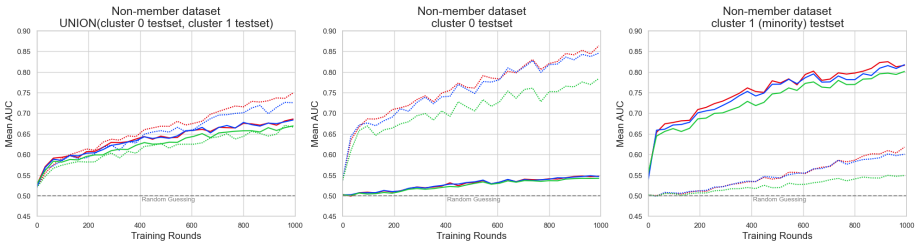
cluster 0, with AUCs ranging from 0.77 to 0.80, while AUCs for cluster 1 remain comparatively low (between 0.54 and 0.60). In this setting, class imbalance appears to have a more pronounced effect on the attack success for cluster 1. Although overall AUC remains relatively small, the 14:2 setting approaches 0.6, indicating moderate attack effectiveness under severe imbalance.

6.2.2 Head size

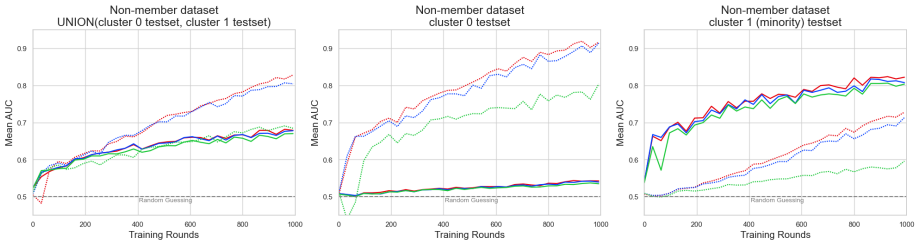
Figure 6.2 shows how the size the head in FACADE affects the success of the MIAs across different cluster ratios and non-member data samples selection.



(a) 8:8 cluster configuration



(b) 12:4 cluster configuration



(c) 14:2 cluster configuration

Figure 6.2: Impact of *head size* on MIA performance across training rounds for different non-member sets and 3 cluster configurations

The results are presented for three cluster configurations–8:8 (balanced), 12:4 (imbalanced), and 14:2 (highly imbalanced)–each analyzed under three non-member test set conditions: union of both clusters, cluster 0 only, and cluster 1 only.

8:8 cluster configuration In the balanced setting, the mean AUC increases with the number of layers in the model head across all non-member set configurations. The most notable gains occur when increasing the head depth from 1 to 2 layers. For example, cluster 0 exhibits AUC improvements of roughly 0.03 with the union set, 0.03 with cluster 0, and 0.05 with cluster 1 as the non-member source. However, further increasing the head size from 2 to 4 layers yields only marginal benefits, with AUC gains below 0.01. Interestingly, when the non-member set is drawn only from a single cluster, the performance improvement is more pronounced for the *other* cluster—i.e., the one whose feature distribution differs from the non-member data.

12:4 cluster configuration In this setting, larger head sizes result in more pronounced AUC increase for the *minority* cluster (cluster 1). Moving from 1 to 2 head layers yields performance gains of 0.06 with both the union and cluster 0 non-member sets, and 0.05 with the cluster 1 set. In contrast, AUC variations for cluster 0 are minimal—particularly when the non-member data is drawn from its own distribution. For instance, attacks on cluster 0 using cluster 0 non-member data show no improvement with additional head layers.

14:2 cluster configuration In the most imbalanced setting, the AUC increases even more sharply with head depth for the minority cluster. The improvement in AUC when increasing the head from 1 to 2 layers is consistently 0.11 across all setups. In the worst-case scenario, where only data from cluster 0's distribution is used as non-member data, the AUC score reaches 0.91, indicating a highly successful attack. In contrast, the impact of head depth on attack performance for cluster 0 nearly disappears, with same AUC scores across different head sizes.

6.3 Summary

The results from Section 6.1 demonstrate that the success of membership inference attacks on FACADE is influenced by both *cluster imbalance* and the *feature distribution* of the non-member dataset used for the attack. Although class imbalance slightly increases overall vulnerability—particularly for nodes in the minority cluster—its impact is secondary to that of the disparity in distribution between member and non-member data. Attacks are significantly more successful when the non-member data samples are drawn from a different feature distribution than the target's one. In contrast, when the non-member set includes samples drawn from both clusters' data distribution, the attack remains efficient (better than random with AUC scores above 0.65 4.3), though more balanced across clusters. Section 6.2.2 further reveals that *increasing the depth of the model head* amplifies privacy leakage, especially for the minority cluster. As class imbalance increases, deeper heads (particularly moving from 1 to 2 layers) lead to important gains in attack performance. Once again, the effect is stronger when there is a distributional divergence between the attack's member and non-member sets.

These results support the claim that model design choices, such as head size and thus models personalization, along with cluster imbalance in FACADE, critically influence privacy risks during training. The findings also underscore the sensitivity of MIA performance to the attacker's access to and choice of non-member data. This highlights the critical need to consider realistic threat models when assessing privacy risks.

Discussion and Future Work

This section discusses some limitations of the current approach and outlines directions for future research.

Limitations One limitation concerns the assumptions made about the *attacker's knowledge*. Cross-cluster MIAs rely on the attacker having access to data from a different feature distribution than their own—such as through auxiliary datasets or prior knowledge of population subgroups. While this scenario is realistic in domains like healthcare, where subpopulations are well-characterized, it may not apply in more opaque systems. In such cases, the attacker may be unable to construct a non-member set with sufficient distributional divergence, reducing the effectiveness of the attack.

Another limitation stems from the *experimental setup*. The study used cluster-based synthetic distributions to conduct controlled experiments. While this approach effectively isolates the effects of distributional shifts, it may oversimplify the complexities of real-world data. Additionally, the experiments were limited to two clusters, whereas real-world datasets often involve richer, more nuanced heterogeneity with multiple overlapping subgroups. Therefore, the results may not fully generalize to more realistic, diverse settings.

Future directions To build on this work, future research should explore the *scalability* and *generalization* of the findings. This includes testing the framework in larger decentralized networks, experimenting with a broader range of cluster configurations, and introducing other forms of distribution shifts (e.g.,

color variations). Expanding to real-world datasets would also help assess the robustness and practical relevance of the observed privacy vulnerabilities.

Another interesting direction involves broadening the scope of *attack strategies*. This study focused on loss-based MIAs under a specific threat model. Investigating more advanced adversaries—including those capable of collusion—could reveal new risks. Evaluating a variety of attacker capabilities and assumptions will offer a more comprehensive assessment of privacy exposure.

Finally, a natural continuation is to incorporate *privacy-preserving mechanisms* into FACADE. Techniques such as differential privacy or homomorphic encryption could be explored to mitigate leakage while preserving model utility and fairness.

Conclusion

This project presents the first empirical privacy evaluation of FACADE. By implementing a loss-based membership inference attack, I evaluated FACADE's vulnerability to inference threats under varying conditions, including cluster imbalance, model head size, and distributional properties of the attack data.

The results show that while FACADE effectively improves fairness under training data feature heterogeneity, it also increases the distinctiveness of local models, making them more prone to MIAs. Notably, higher model personalization–through deeper heads or extended training—correlates with increased MIA success, especially for minority-cluster nodes. Furthermore, an attacker's access to non-member data from a different feature distribution than the target's data significantly boosts inference power, highlighting the sensitivity of privacy leakage to distributional mismatches.

These findings highlight the fundamental privacy–fairness trade-off in personalized decentralized learning. They underscore the importance of accounting for adversarial capabilities and data distribution when evaluating privacy risks, and they motivate the integration of privacy-preserving techniques to reduce privacy leakage without severely undermining fairness.

Achieving both fairness and privacy in decentralized learning is not a matter of choosing one over the other, but of designing systems that explicitly and thoughtfully balance the two.

Acknowledgements

I would like to sincerely thank Sayan Biswas, Martijn de Vos, and Milos Vujasinovic for their guidance, encouragement, and insightful feedback throughout this project. Their support was instrumental in helping me navigate the complexities of both the theoretical and experimental aspects of this work.

I also gratefully acknowledge the Scalable Computing Systems Laboratory (SaCS) at EPFL for offering me the opportunity to carry out this project within their lab, and for providing a collaborative environment that was both intellectually and personally rewarding.

Bibliography

[1] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," in *NIPS'17*, vol. 30, Long Beach, California, USA, 2017, pp. 5336–5346. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/f75526659f31040afeb61cb7133e4e6d-Paper.pdf

[2] S. Saif, M. J. Islam, M. Z. B. Jahangir, P. Biswas, A. Rashid, M. A. A. Nasim, and K. D. Gupta, "A comprehensive review on understanding the decentralized and collaborative approach in machine learning," 2025. [Online]. Available: https://arxiv.org/abs/2503.09833

[3] S. Biswas, A.-M. Kermarrec, R. Sharma, T. Trinca, and M. De Vos, "Fair decentralized learning," in *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, 2025, pp. 714–734.

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

[5] S. Abdulrahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5476–5497, 2021.

[6] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," 2023. [Online]. Available: https://arxiv.org/abs/1602.05629

[7] A. Dhasade, A.-M. Kermarrec, R. Pires, R. Sharma, and M. Vujasinovic, "Decentralized learning made easy with decentralizepy," in *Proceedings of the 3rd Workshop on Machine Learning and Systems*, ser. EuroMLSys '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 34–41. [Online]. Available: https://doi.org/10.1145/3578356.3592587

[8] M. Rigaki and S. Garcia, "A survey of privacy attacks in machine learning," *ACM Comput. Surv.*, vol. 56, no. 4, Nov. 2023. [Online]. Available: https://doi.org/10.1145/3624010

[9] S. Truex and M. Malan, "Privacy in practice: Research challenges in the deployment of privacy-preserving ml," in *2024 IEEE 6th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)*, 2024, pp. 157–165.

[10] L. Bai, H. Hu, Q. Ye, H. Li, L. Wang, and J. Xu, "Membership inference attacks and defenses in federated learning: A survey," *ACM Comput. Surv.*, vol. 57, no. 4, Dec. 2024. [Online]. Available: https://doi.org/10.1145/3704633

[11] D. Pasquini, M. Raynal, and C. Troncoso, "On the (in)security of peer-to-peer decentralized machine learning," in *2023 IEEE Symposium on Security and Privacy (SP)*, 2023, pp. 418–436.

[12] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 1175–1191. [Online]. Available: https://doi.org/10.1145/3133956.3133982

[13] C. Dwork, "Differential privacy," in *Automata, Languages and Programming*, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–12.

[14] S. Biswas, M. Even, A.-M. Kermarrec, L. Massoulié, R. Pires, R. Sharma, and M. de Vos, "Noiseless privacy-preserving decentralized learning," *Proceedings on Privacy Enhancing Technologies*, vol. 2025, no. 1, p. 824–844, Jan. 2025. [Online]. Available: http://dx.doi.org/10.56553/popets-2025-0043

[15] A. Paverd, A. Martin, and I. Brown, "Modelling and automatically analysing privacy properties for honest-but-curious adversaries," *Tech. Rep*, 2014.

[16] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[17] K. Hsieh, A. Phanishayee, O. Mutlu, and P. Gibbons, "The non-IID data quagmire of decentralized machine learning," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 4387–4398. [Online]. Available: https://proceedings.mlr.press/v119/hsieh20a.html