



HAL
open science

Scrutinizing the Vulnerability of Decentralized Learning to Membership Inference Attacks

Ousmane Touat, Jezekael Brunon, Yacine Belal, Julien Nicolas, Mohamed Maouche, César Sabater, Sonia Ben Mokhtar

► **To cite this version:**

Ousmane Touat, Jezekael Brunon, Yacine Belal, Julien Nicolas, Mohamed Maouche, et al.. Scrutinizing the Vulnerability of Decentralized Learning to Membership Inference Attacks. 2025. hal-04933985

HAL Id: hal-04933985

<https://hal.science/hal-04933985v1>

Preprint submitted on 7 Feb 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Scrutinizing the Vulnerability of Decentralized Learning to Membership Inference Attacks

Ousmane Touat
ousmane.touat@insa-lyon.fr
INSA Lyon, LIRIS, CNRS
Villeurbanne, France

Julien Nicolas
julien.nicolas@insa-lyon.fr
INSA Lyon, LIRIS, McGill, ILLS
Villeurbanne, France

Jezekael Brunon
jezekael.brunon@insa-lyon.fr
INSA Lyon, LIRIS
Villeurbanne, France

César Sabater
cesar.sabater@insa-lyon.fr
INSA Lyon, LIRIS, CNRS
Villeurbanne, France

Yacine Belal
yacine.belal@insa-lyon.fr
INSA Lyon, LIRIS
Villeurbanne, France

Mohamed Maouche
mohamed.maouche@inria.fr
INRIA
Villeurbanne, France

Sonia Ben Mokhtar
sonia.ben-mokhtar@cnrs.fr
INSA Lyon, LIRIS, CNRS
Villeurbanne, France

Abstract

The primary promise of decentralized learning is to allow users to engage in the training of machine learning models in a collaborative manner while keeping their data on their premises and without relying on any central entity. However, this paradigm necessitates the exchange of model parameters or gradients between peers. Such exchanges can be exploited to infer sensitive information about training data, which is achieved through privacy attacks (*e.g.*, Membership Inference Attacks – MIA). In order to devise effective defense mechanisms, it is important to understand the factors that increase/reduce the vulnerability of a given decentralized learning architecture to MIA. In this study, we extensively explore the vulnerability to MIA of various decentralized learning architectures by varying the graph structure (*e.g.*, number of neighbors), the graph dynamics, and the aggregation strategy, across diverse datasets and data distributions. Our key finding, which to the best of our knowledge we are the first to report, is that the vulnerability to MIA is heavily correlated to (i) the local model mixing strategy performed by each node upon reception of models from neighboring nodes and (ii) the global mixing properties of the communication graph. We illustrate these results experimentally using four datasets and by theoretically analyzing the mixing

properties of various decentralized architectures. Our paper draws a set of lessons learned for devising decentralized learning systems that reduce by design the vulnerability to MIA.

CCS Concepts: • **Computing methodologies** → **Distributed artificial intelligence**; • **Computer systems organization** → **Peer-to-peer architectures**; • **Security and privacy** → **Privacy-preserving protocols**.

Keywords: Gossip Learning, Membership Inference Attacks, Peer Sampling, Model Mixing

ACM Reference Format:

Ousmane Touat, Jezekael Brunon, Yacine Belal, Julien Nicolas, César Sabater, Mohamed Maouche, and Sonia Ben Mokhtar. 2025. Scrutinizing the Vulnerability of Decentralized Learning to Membership Inference Attacks. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 14 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 Introduction

Federated learning [42], which allows training machine learning models while keeping users' data on their premises has gained significant interest in recent years both from academia and industry [1, 18, 48, 62]. However, this approach is often criticized because of the trust it requires on the aggregation server, which may be subject to failures [29], attacks [6], scalability issues [38], *etc.* In order to deal with these challenges, Decentralized Learning, which builds on the peer-to-peer exchange of model parameters between participants is gaining an increasing focus (*e.g.*, [21, 39, 49]). Indeed, decentralized architectures have been heavily studied for decades and their scalability and resilience properties have been extensively documented both theoretically and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference'17, July 2017, Washington, DC, USA

© 2025 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM. . \$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

experimentally in the literature [2, 41, 60]. This makes decentralized learning a natural avenue to explore for large-scale, distributed learning tasks [24, 40, 63].

However, whether federated or decentralized, collaborative learning has been shown to be vulnerable to privacy attacks and in particular to Membership Inference Attacks (MIA) [43, 46, 68]. This family of attacks aims at predicting whether a given data item has been used or not in the model training set, which can reveal sensitive information about the data owner (e.g., a model trained to predict the risk of heart disease, could reveal sensitive information such as a patient's history of cholesterol levels, blood pressure, or family medical history [33]). Moreover, since MIA is closely related to other privacy attacks [67], the assessment of the vulnerability against such attack provides valuable insight into a model's overall vulnerability to other privacy attacks [50].

In this context, an active debate started in the research community to understand whether decentralization increases or decreases the vulnerability to MIA compared to FL architectures. For instance, Pasquini et al. [50] points out the vulnerability of decentralized topologies, while Ji et al. [26] contradicts this study by claiming that FL topologies are weaker. While these papers contribute to advancing the global knowledge on this topic, these studies are limited because they mainly focus on static graph topologies, ignoring decades of research on dynamic peer-to-peer systems enabled by their underlying random peer sampling protocols (RPS) [25, 61].

In this paper, we aim to contribute to this debate by analyzing the key factors that make decentralized architectures more or less vulnerable to MIA. Indeed, there are various ways of deploying a decentralized learning task, which differ according to the graph topology (e.g., k -regular graph with variable view sizes), graph dynamics (e.g., static graph, vs graph built over RPS protocols), communication synchronicity (synchronous vs asynchronous model exchange) and model mixing and dissemination strategy.

In this study, we focus on the MIA vulnerability of decentralized learning over asynchronous, k -regular graphs (also defined as Gossip Learning in [22, 49]) and vary all the remaining factors. Specifically, (i) we explore two local model mixing and dissemination strategies by introducing the *send-all-merge-once* (SAMO) strategy; (ii) we vary graph dynamics; (iii) we vary graph density through the number of neighbors. Beyond these key architectural features, we further consider four commonly used datasets and corresponding models and study the impact of i.i.d and non-i.i.d data distributions. We evaluate the MIA vulnerability of the overall decentralized architecture by considering that the attacker can observe all the messages exchanged in the system and run the MIA against each participant using messages received from its neighbors.

Our key finding is that MIA vulnerability is heavily correlated with two factors: (i) the *local model mixing* and dissemination strategy, which determines how a node aggregates models received from his neighbors and how he disseminates his updated model and (ii) the *global mixing properties of the communication graph*. For instance, we show that a static highly connected graph has a similar vulnerability to a dynamic graph of the same size. Instead, in weakly connected graphs (e.g., 2-regular graphs), graph dynamics brought by RPS protocols clearly decrease MIA vulnerability. Beyond local and global mixing properties, we also observe that learning tasks over non-i.i.d data distributions clearly increase MIA vulnerability.

Summarizing, our paper makes the following contributions:

- **Comprehensive Study of MIA Vulnerabilities in Decentralized Learning:** To the best of our knowledge, we propose the first study that scrutinizes how MIA vulnerability is affected by decentralized learning settings, focusing on the roles of graph connectivity, graph dynamics, model mixing, and data distributions.
- **Insights into Mixing Properties and Privacy:** We identify that both local and global model mixing properties are critical determinants of MIA vulnerability. Our findings reveal how dynamic graph settings and enhanced view sizes improve model mixing and mitigate privacy risks. Note that all our results can be reproduced using the code available in the following public (yet anonymized) repository ¹.
- **Recommendations for Privacy-Aware Decentralized Learning beyond generalization error:** Based on our findings, we provide design recommendations for decentralized learning systems that aim to balance utility with reduced MIA vulnerabilities, emphasizing the importance of dynamics and robust mixing protocols. We show that generalization error, which has been highlighted by previous works [50], is not the only key factor in determining the privacy risk.

The remainder of this paper is structured as follows: we first present preliminaries on decentralized learning, gossip learning, and a variant of gossip learning we consider in this paper as well as background on MIA and our threat model in Section 2. We then present our experimental analysis in Section 3. We complement our study with a theoretical analysis of graph mixing properties in Section 4. Then, we summarize our actionable recommendations in Section 5. Finally, we discuss related research efforts in Section 6 before concluding the paper in Section 7.

2 Preliminaries and Setup

In this section we present an overview of decentralized learning (Section 2.1) and the two learning protocols we consider,

¹<https://anonymous.4open.science/r/ExposingVulneDL>

Base Gossip Learning (Section 2.2) and a variant of this protocol (Section 2.3). We detail the neighbor selection strategy for these two protocols in Section 2.4. Then, we introduce membership inference attacks (Section 2.5) and our considered threat model (Section 2.6).

2.1 Decentralized Learning

Decentralized Learning refers to a set $\mathcal{V} = \{1, \dots, n\}$ of nodes, which aim to collaboratively train a machine learning model without relying on a central parameter server. Given a set of attributes \mathcal{Z} , a set of labels \mathcal{Y} and a model domain \mathcal{M} , each node $i \in \mathcal{V}$ has its local dataset denoted as $\mathcal{D}_i \subset \mathcal{X} = \mathcal{Z} \times \mathcal{Y}$ and their joint goal is to learn a global model $\theta^* \in \mathcal{M}$ which minimizes the loss function:

$$\mathcal{L}(\theta^*) = \frac{1}{n} \sum_{i=1}^n l_i(\theta^*, \mathcal{D}_i), \quad (1)$$

where $l_i(\cdot, \cdot) : \mathcal{M} \times \wp(\mathcal{X}) \rightarrow \mathbb{R}$ is the local loss function² of node i and $\mathcal{D} = \bigcup_{i \in \mathcal{V}} \mathcal{D}_i$ as defined in [5].

In decentralized learning, each node performs three types of operations over models: local model updates, model exchanges, and model aggregation as described below.

Local Updates. Each node $i \in \mathcal{V}$ updates its current model θ_i using its local training dataset \mathcal{D}_i as follows:

$$\theta_i = \theta_i - \eta \nabla l_i(\theta_i, \mathcal{D}_i), \quad (2)$$

where η is the learning rate and $\nabla l_i(\theta_i, \mathcal{D}_i)$ is the gradient of $l_i(\theta_i, \mathcal{D}_i)$.

As each node may update its model parameters by performing more than one local step, we will slightly abuse notation and still note $\eta \nabla l_i(\theta_i, \mathcal{D}_i)$ the sum of gradients obtained by sequentially applying a previously defined number of local steps with a given learning rate η .

Model Exchange. Each node $i \in \mathcal{V}$ exchanges its current model with one or many other nodes in its neighborhood. This neighborhood can be static or dynamic as further discussed in Section 2.4. Let $p \in \mathbb{N}$ be our time parameter, $\mathcal{N}_i^{(p)} \subset \mathcal{V}$ is the view of party i at time p . In practice, parties obtain their view through a random peer sampling service (RPS) [11, 25]. At each time $p \in \mathbb{N}$, the sets of views of all nodes $\{\mathcal{N}_i^{(p)}\}_{i \in \mathcal{V}}$ define a graph $G^{(p)} = (\mathcal{V}, E^{(p)} \subset \mathcal{V} \times \mathcal{V})$ where an edge $(i, j) \in E^{(p)} \iff j \in \mathcal{N}_i^{(p)}$.

Local Aggregation. To gain information from their neighbors, nodes aggregate their current model with received models. While for different purposes there exist many alternatives to perform this aggregation [30, 37, 52], we consider simple averaging of the current incoming models as in [35].

In the following section, we precisely describe how these operations are performed in the protocols we study.

2.2 Base Gossip Learning

In this paper, we focus on a particular type of decentralized learning, known as Gossip learning (GL) [22, 49], which assumes that nodes behave asynchronously. In Gossip Learning, each node initially starts with a common model θ_0 and a set of neighbors also referred to as a view, and periodically and independently wakes up to perform the above three operations *i.e.*, local updates, model exchange, and local aggregation.

Specifically, the GL algorithm defined in [49] is presented in Algorithm 1. In this algorithm, each node starts with a shared initial model θ_0 (line 1). When a node $i \in \mathcal{V}$ wakes up at time $p \in \mathbb{N}$, it selects a random neighbor from $\mathcal{N}_i^{(p)}$ and sends its current model θ_i (lines 3 and 4). When it receives a model θ_j , it aggregates it with its current model θ_i (line 7) and performs a local update (line 8).

This algorithm is illustrated with an example depicted in the left part of Figure 1. In this figure, node x receives a model from node y_1 (step ①), then performs a local update after aggregating the two models (step ②). Later, it receives a model from node y_2 (step ③) and performs again model aggregation and local update (step ④). Finally, upon waking up, it randomly selects one of its neighbors, *i.e.*, node z_1 to which it sends its latest model (step ⑤).

We observe in this algorithm that models are aggregated in a pairwise manner, after which a local update is performed by the model owner. Furthermore, we observe that the model dissemination strategy is slow as each node sends its updated model to only one randomly selected neighbor. To increase model mixing, we propose to investigate in this paper an alternative to this classical gossip learning algorithm, which we present in the next section.

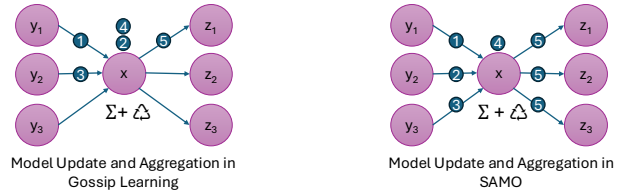


Figure 1. Model Update and Aggregation in Gossip Learning (left) vs in SAMO (right)

2.3 Send-All-Merge-Once Protocol

We hereby introduce in Algorithm 2 a variant of the GL protocol, called *Send-All-Merge-Once* (SAMO) that is designed to improve the speed of model mixing over time. In SAMO, when a node receives a model, it stores it (as shown in line 11) instead of directly aggregating it and performing local updates. Later, when a node wakes up, if models were received it aggregates them with its own (line 4) and performs local updates (line 5). Finally, it sends its current model to all its neighbors (line 8). This algorithm is illustrated with

²for a set S we denote by $\wp(S)$ to the set of all subsets of S

Algorithm 1 Base GL Protocol

-
- 1: **Input:** Communication topologies $\mathcal{N}_i^{(p)}$ for each time p and node $i \in \mathcal{V}$; initial model $\theta_i = \theta_0$, local data-sets \mathcal{D}_i , loss function $l_i(\cdot, \cdot)$ for all $i \in \mathcal{V}$; learning rate η
 - 2: **for** each node $i \in \mathcal{V}$ that **wakes up** within time frame p **do**
 - 3: Select a node $j \in \mathcal{N}_i^{(p)}$ at random
 - 4: Send θ_i to node j
 - 5: **end for**
 - 6: **for** each node $i \in \mathcal{V}$ that **receives a model** θ_j : **do**
 - 7: Set $\theta_i \leftarrow \frac{\theta_i + \theta_j}{2}$ {aggregate with incoming model}
 - 8: Set $\theta_i \leftarrow \theta_i - \eta \nabla l_i(\theta_i, \mathcal{D}_i)$ {perform local updates }
 - 9: **end for**
-

an example depicted in the right part of Figure 1. In this figure, node x receives models from its neighbors y_1 , y_2 and y_3 and stores them locally (steps ①, ② and ③). Later, upon waking up, node x aggregates these models with its local model performs local updates (step ④), and sends the new model to all its neighbors (step ⑤).

As exchanges involve all the neighbors of a node, SAMO has a higher communication cost than the Base GL Protocol. Nevertheless, it better resists MIA (see Section 3).

Algorithm 2 SAMO Protocol

-
- 1: **Input:** Communication topologies $\mathcal{N}_i^{(p)}$ for each time frame p and node $i \in \mathcal{V}$, initial model $\theta_i = \theta_0$, local datasets \mathcal{D}_i , incoming model sets $\Theta_i = \{\theta_0\}$ and loss function $l_i(\cdot, \cdot)$ for all $i \in \mathcal{V}$; learning rate η
 - 2: **for** each node $i \in \mathcal{V}$ that **wakes up** within time frame p **do**
 - 3: **if** $|\Theta_i| > 1$ **then**
 - 4: Set $\theta_i \leftarrow \frac{1}{|\Theta_i|} \sum_{\theta \in \Theta_i} \theta$ {aggregate models}
 - 5: Set $\theta_i \leftarrow \theta_i - \eta \nabla l_i(\theta_i, \mathcal{D}_i)$ {do local updates }
 - 6: Set $\Theta_i \leftarrow \{\theta_i\}$
 - 7: **end if**
 - 8: Send θ_i to each node $j \in \mathcal{N}_i^{(p)}$
 - 9: **end for**
 - 10: **for** each node $i \in \mathcal{V}$ that **receives a model** θ_j **do**
 - 11: Set $\Theta_i \leftarrow \Theta_i \cup \theta_j$
 - 12: **end for**
-

2.4 Peer Sampling Service

In this paper, we aim to study both static and dynamic topologies. For both GL and SAMO, we assume that nodes rely on a Random Peer Sampling Service such as [25]. Specifically, in both protocols, neighbors of each node $i \in \mathcal{V}$, are chosen as follows. Initially, $\mathcal{N}_i^{(1)}$ is determined by a k -regular graph $G^{(1)}$ chosen at random by the peer sampling service. In the

static setting, neighbors do not change anymore. In the dynamic setting we adopt the PeerSwap method [17]: when node i wakes up at time p , it swaps the view with a randomly selected neighbor before doing anything else. More precisely, $\mathcal{N}_i^{(p)}$ is set to $\mathcal{N}_j^{(p-1)} \setminus \{i\} \cup \{j\}$ through the peer sampling service, which also updates

$$\begin{aligned} \mathcal{N}_j^{(p)} &\leftarrow \mathcal{N}_i^{(p-1)} \setminus \{j\} \cup \{i\} \\ \mathcal{N}_k^{(p)} &\leftarrow \mathcal{N}_k^{(p-1)} \setminus \{i\} \cup \{j\} \text{ for all } k \in \mathcal{N}_i^{(p-1)} \setminus \{j\} \\ \mathcal{N}_k^{(p)} &\leftarrow \mathcal{N}_k^{(p-1)} \setminus \{j\} \cup \{i\} \text{ for all } k \in \mathcal{N}_j^{(p-1)} \setminus \{i\} \end{aligned}$$

to keep views consistent. The swap is equivalent to changing the position of i and j in the current graph, which remains k -regular.

2.5 Membership Inference Attack

A Membership Inference Attack (MIA) is a privacy attack whose goal is to infer for a given model $\theta \in \mathcal{M}$ whether or not a sample $x \in \mathcal{X}$ belongs to the data used to train this model [55]. More formally, it is an algorithm $\mathcal{A} : \mathcal{M} \times \mathcal{X} \rightarrow \{0, 1\}$ such that $\mathcal{A}(\theta, x) = 1$ if it predicts that x belongs to the training dataset of model θ and $\mathcal{A}(\theta, x) = 0$ otherwise.

Many MIAs have been proposed and range from expensive approaches that train ML models to predict membership such as neural shadow models [55] to information-theoretic estimators that leverage prediction entropy or confidence [54, 57]. In our work, we use the Modified Prediction Entropy Attack [57]. We first explain the Modified Prediction Entropy (MPE) measure and then the attack.

Let $P : \mathcal{Y} \rightarrow [0, 1]$ be a probability distribution over \mathcal{Y} and $y \in \mathcal{Y}$. The MPE measure $M_{\text{MPE}}(\cdot, \cdot)$ is defined as follows:

$$\begin{aligned} M_{\text{MPE}}(P, y) &= - (1 - P(y)) \log(P(y)) \\ &\quad - \sum_{y' \in \mathcal{Y} \setminus \{y\}} P(y') \log(1 - P(y')). \end{aligned} \quad (3)$$

The MPE Attack used to discriminate if the sample $(z, y) \in \mathcal{X}$ was used to train model $\theta \in \mathcal{M}$ is defined as follows:

$$\mathcal{A}_{\text{MPE}}(\theta, (z, y)) = \begin{cases} 1 & \text{if } M_{\text{MPE}}(\theta[z], y) \leq \tilde{\tau} \\ 0 & \text{otherwise} \end{cases}. \quad (4)$$

where $\theta[z] : \mathcal{Y} \rightarrow [0, 1]$ is the probability distribution that model θ assigns to each label in \mathcal{Y} given input attributes $z \in \mathcal{Z}$ and $\tilde{\tau}$ is the attack threshold computed using training dataset \mathcal{D}_i of a target node (or victim) $i \in \mathcal{V}$ as well as a test dataset \mathcal{D}_i^t that was not used to train θ . The idea is that the prediction entropy can discriminate between member and non-member data, as the target model θ has a lower prediction entropy on train data.

It is unrealistic to deploy \mathcal{A}_{MPE} in practice as it uses the training dataset of the victim to get the optimal threshold $\tilde{\tau}$. However, this approach offers an upper bound on the worst-case attacker whose information is the probability distributions $\{\theta[z]\}_{z \in \mathcal{Z}}$ generated by the target model θ and

sample x . Therefore it is an informative privacy assessment metric. Moreover, it is significantly more practical to deploy than machine learning-based discriminators.

2.6 Threat Model

As an attacker, we consider an omniscient observer that at regular time intervals recovers the current models of all nodes $\{\theta_i\}_{i \in \mathcal{V}}$ and performs \mathcal{A}_{MPE} on each one of them, targeting each data sample of each node. This attacker is similar to the passive attacker defined in [50] to assess the vulnerability of static decentralized learning against MIA. It allows us to evaluate the vulnerability of all nodes in the system wherever their position is in the communication graph. A more detailed explanation of the metrics we used to compute the vulnerability to MIA is provided in Section 3.2.

3 Experimental Study

In this section, we present the findings of our study, which investigates the factors influencing the tradeoff between utility and privacy in Gossip learning. Specifically, we address the following research questions:

- **RQ1:** What is the impact of using the SAMO protocol compared to Base Gossip?
- **RQ2:** Is there an advantage to using a dynamic setting compared to a static one?
- **RQ3:** What is the impact of the view size on both static and dynamic settings?
- **RQ4:** What is the impact of non-i.i.d. data distribution?
- **RQ5:** Is the link between generalization error and MIA vulnerability maintained in our studied setting?

3.1 Experimental Setup & Configurations

Implementation of Gossip Learning Protocol. We consider two gossip learning protocols: the Base Gossip protocol and SAMO. Both protocols are implemented using our fork of the GossiPy python framework [51] with Python 3.11 and Pytorch.

To simulate the asynchronous nature of the gossip protocol, the execution is divided into discrete time units called ticks. Each round of communication is represented by 100 ticks and each node $i \in \mathcal{V}$ waits Δ_i ticks between wake-ups. The waiting time Δ_i is sampled from a normal distribution $\mathcal{N}(\mu, \sigma^2)$ with $\mu = 100$ and $\sigma^2 = 100$ at the beginning of the execution.

Communication Topology. In our setup we simulate 150 nodes that are connected together in an initial k -regular graph topology with the view size $k \in \{2, 5, 10, 25\}$. We consider both static and dynamic settings.

Datasets and Partitioning. The efficacy of the proposed protocol is assessed on a number of datasets described in

Table 1. For experiments on independent and identically distributed (i.i.d.) data, we utilize standard image classification datasets: CIFAR-10, CIFAR-100 [34] and FashionMNIST [64], alongside the Purchase100 dataset [55], which is commonly used in privacy attack research for tabular data. In this configuration, the training data is distributed uniformly across all nodes in an equal partition, while the global test set is maintained separately to assess the global test accuracy of each model.

Training Setup. For our experiments, we trained each model with Gossip Learning (base algorithm and our variant), varying hyperparameters such as the learning rate, the number of local steps, or the used model for the dataset depending on its task complexity. Table 2 shows the training setup for all our datasets, including the used model and the number of parameters. For light image datasets such as CIFAR-10 or Fashion-MNIST, we use a light CNN architecture. For Purchase100 we follow the implementation from Nasr et al. [46], by using a 4-layer fully-connected neural network. For more challenging image datasets such as CIFAR-100, we opt to use ResNet models [20]. We initialize each node’s model using Kaiming normal initialization function [19].

3.2 Metrics

Utility: Accuracy. In all experiments, we assess the utility of the models using top-1 accuracy. For a model θ , we define its accuracy on a dataset \mathcal{D} as follow:

$$\text{acc}(\theta, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(z, y) \in \mathcal{D}} \mathbb{I}(\arg \max \theta(z) = y), \quad (5)$$

where \mathbb{I} is the indicator function.

Privacy: MIA Vulnerability. We measure MIA vulnerability using the accuracy of the MIA attack. Let θ be the victim model, \mathcal{A}_{MPE} be the attack described in Section 2.5 and $\mathcal{D}_{\text{att}} \subset \mathcal{Z} \times \mathcal{Y} \times \{0, 1\}$ a set of member and non-member data. Membership in \mathcal{D}_{att} is defined with label m (1 for member and 0 for non-member) and usually sampled equally from the local train and test set respectively. We then define the MIA vulnerability as below:

$$\text{acc}_{\mathcal{A}}(\theta, \mathcal{D}_{\text{att}}) = \frac{1}{|\mathcal{D}_{\text{att}}|} \sum_{(z, y, m) \in \mathcal{D}_{\text{att}}} \mathbb{I}(\mathcal{A}_{\text{MPE}}(\theta, (z, y)) = m). \quad (6)$$

Generalization error. We define a generalization error metric as the gap between the local train accuracy and the local test accuracy. Formally, for node i that trains model θ_i with its local dataset \mathcal{D}_i , we define the generalization error as below:

$$\text{gen_error}(\theta_i, \mathcal{D}_i) = \text{acc}(\theta_i, \mathcal{D}_{i, \text{train}}) - \text{acc}(\theta_i, \mathcal{D}_{i, \text{test}}), \quad (7)$$

with $\mathcal{D}_{i, \text{train}}$ and $\mathcal{D}_{i, \text{test}}$ being respectively the local training and test dataset splits of node i .

Table 1. Dataset Characteristics

Dataset	Train Set	Test Set	Input Size	Classes	Model	Description
CIFAR-10	50,000	10,000	$32 \times 32 \times 3$	10	CNN	Color images across 10 classes including animals, vehicles
CIFAR-100	50,000	10,000	$32 \times 32 \times 3$	100	ResNet-8	Fine-grained color images with 100 object classes
Fashion-MNIST	60,000	10,000	$28 \times 28 \times 1$	10	CNN	Grayscale images of clothing and fashion accessories
Purchase-100	157,859	39,465	600	100	MLP	A tabular dataset of customer purchases to classify their buying behavior

Table 2. Training Configuration

Dataset	Model	Parameters	Learning Rate	Momentum	Weight Decay	Local Epochs	Rounds
CIFAR-10	CNN	124k	0.01	0	5e-4	3	250
CIFAR-100	ResNet-8	1.2M	0.001	0.9	5e-4	5	500
Fashion-MNIST	CNN	81k	0.01	0	5e-4	3	250
Purchase-100	MLP	1.3M	0.01	0.9	5e-4	10	250

3.3 Comparing SAMO to Base Gossip (RQ1)

To start our evaluation, we want to emphasize why we propose the protocol SAMO (described in Algorithm 2). To do that, we show in Figure 2 the tradeoff between average MIA vulnerability and average test accuracy for models trained on the CIFAR-10, CIFAR-100, Fashion-MNIST, and Purchase100 datasets. Each point in the plot represents an evaluation during a given round, with the x-axis showing the average model test accuracy and the y-axis showing the corresponding average MIA vulnerability.

Our results show that given a target test accuracy the SAMO protocol outperforms Base Gossip in most settings, especially in the regimes where the test accuracy is maximal. More precisely, with SAMO, the maximum test accuracy obtained ranges from 35.4% and 88.4% relative to an MIA vulnerability ranging from 62.7% to 92.9%. In comparison, in the Base Gossip setting the maximum test accuracy obtained ranges from 29.9% and 82.6% relative to an MIA vulnerability ranging from 63.5% and 94.1%. We explain this advantage by the fact that SAMO favors model mixing, as each user receives models from multiple neighbors and delays the aggregation in order to hide his or her models within more models.

Takeaway: Favoring model mixing by incorporating the SAMO protocol offers a better privacy/utility trade-off.

Given these results, we favor using the SAMO protocol for the remaining experiments.

3.4 Comparing Static and Dynamic Settings (RQ2)

In this section, we present in Figure 3 the tradeoff between average MIA vulnerability and average test accuracy to compare static and dynamic topology. We observe that in all datasets, settings with dynamic topology outperform the static ones, achieving a better trade-off between MIA vulnerability and test accuracy. More precisely, in the dynamic setting, the maximum test accuracy obtained ranges from 30.4% and 80.9% relative to an MIA vulnerability ranging from 67.0% to 94.6%. While, in the static setting, the maximum test accuracy obtained ranges from 15.8% and 76.4% relative to an MIA vulnerability ranging from 72.0% and 98.8%. We observe that models trained with dynamic topologies maintain lower MIA vulnerability at comparable accuracy levels.

Takeaway: dynamic topologies have a positive impact on both the MIA vulnerability and model utility

3.5 Impact of the View Size on the Privacy/Utility Tradeoff (RQ3)

In this experiment, we aim to study the impact of the view size on the trade-off between utility and privacy. The results are presented in Figure 4 with the maximum average MIA vulnerabilities and the maximum average test accuracy on the CIFAR-10 dataset with various view sizes. The results show that the dynamic settings outperform the static settings in both utility and privacy. Interestingly, upon increasing the view size the differences between dynamic and static

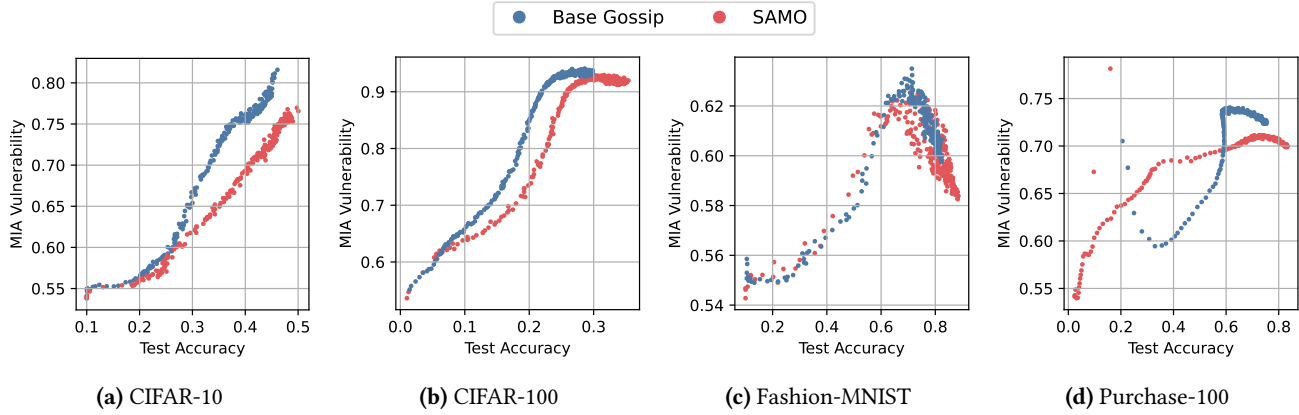


Figure 2. Trade-off between MIA vulnerability and Test Accuracy across different datasets, comparing Base Gossip and SAMO learning protocols, with a static topology on a 5-regular graph with 150 nodes (60 nodes on CIFAR100).

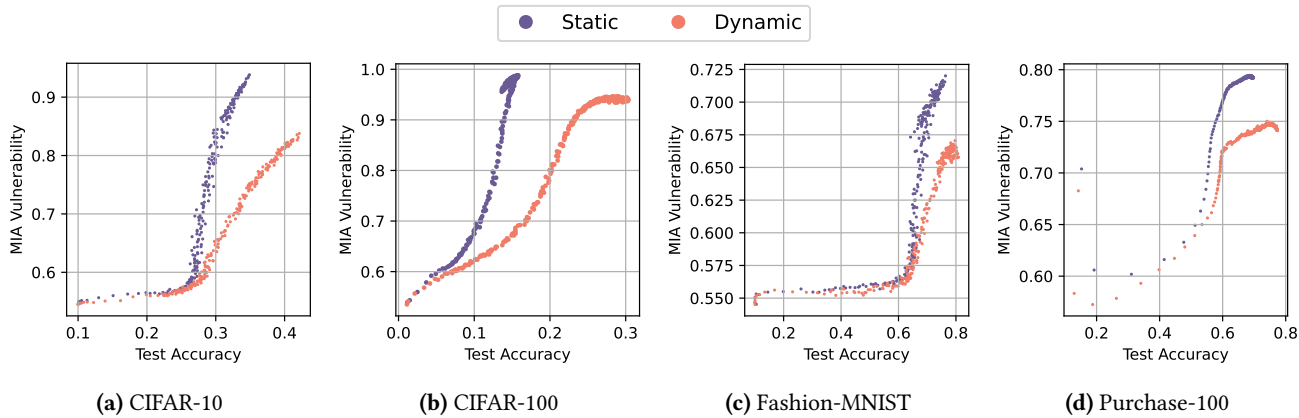


Figure 3. Trade-off between MIA vulnerability and Test Accuracy across different datasets, comparing static and dynamic topology setups, on a 2-regular graph with 150 nodes (60 nodes on CIFAR100).

gets smaller. This is anticipated as increasing the view size gets the communication graph closer to a complete graph where both settings are equivalent. Also, the increase in view size is overall a strict benefit for the privacy/utility trade-off but this is at the cost of increasing the number of models sent per user (i.e., messages). Overall, profiting from both dynamic topology and increased view size there is a better privacy/utility tradeoff to be found. In particular, using a dynamic setting with a view size of 10 is quasi-similar to a static setting with a view size of 25 (with 2.5 times fewer models sent).

We conclude that both dynamicity and increased view size have a positive impact on MIA vulnerability as they both favor the model mixing. This intuition is further investigated in Section 4

Takeaway: Increasing the view size closes the gap of dynamic over static but at the cost of more models sent.

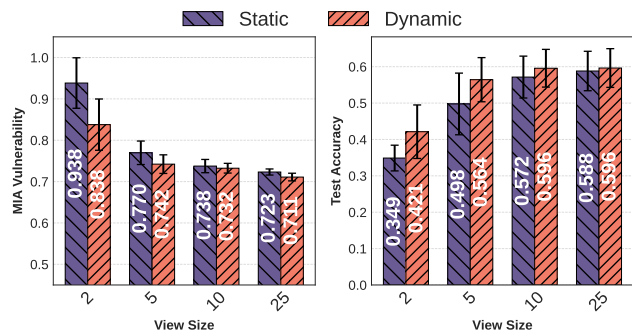


Figure 4. Comparison of maximum average MIA vulnerabilities and according test accuracy across different network view sizes and topology setups on the CIFAR-10 dataset using SAMO, with regular graph with 150 nodes.

3.6 Impact of non-i.i.d Data Distribution (RQ4)

In this experiment, our goal is to examine the impact of non-iidness in Gossip Learning. To achieve this, we use the classical method for enforcing a non-iid setting [36], applying the Dirichlet function to apply label imbalance between each node. Specifically, using the Dirichlet function with parameter β , we sample the proportion $p_k \sim \text{Dir}_N(\beta)$ of records with label k across the training sets of the N nodes. The parameter $\beta > 0$ here controls the degree of heterogeneity, where a lower value of β (≤ 0.1) yields a higher label imbalance among nodes. In contrast, a higher β value yields a distribution closer to the i.i.d distribution.

The results are presented in Figure 5, showing the tradeoff between test accuracy and MIA vulnerability. As expected, the overall test accuracy (utility) decreases with a higher non-iid setting from a maximum of 77.4% for the iid setting until a maximum of 52.4% for non-iid with $\beta = 0.1$, as the non-iid setting is known to make the collaborative learning process more challenging. However, more notably, the MIA vulnerability increases significantly across all training rounds. This is because the learned models tend to align a global model with data distribution close to the overall global distribution, making local updates more prone to leaking membership information.

Regarding the difference between static and dynamic settings, we observe similar conclusions as in the previous experiment, with dynamic settings consistently outperforming static ones. However, the increase in MIA vulnerability due to non-iidness remains prominent regardless of the setting. Often, the dynamic setting successfully approaches the results of the static setting with lower non-iidness but never completely bridges the gap.

Takeaway: Non-iid data distribution among nodes significantly increases MIA vulnerability across all training rounds, and dynamic setting alone is insufficient to completely bridge this gap.

3.7 Rethinking the Link Between Generalization Error and MIA Vulnerability (RQ5)

The relationship between generalization error (Equation (7)) and MIA vulnerability has been extensively studied in the literature, particularly in the centralized setting [55, 58]. This focus is understandable, as most MIA attacks rely on training overfitting, which is exposed through loss function evaluations on specific examples. The distinction between data points seen during training and those unseen is a crucial factor enabling the separation of members from non-members in MIA attacks. Intuitively, a model that generalizes well should be more resilient to MIA attacks. In our setting, however, we are not dealing with a centralized framework involving a single model. Instead, as demonstrated in previous experiments, models undergo various stages of mixing with

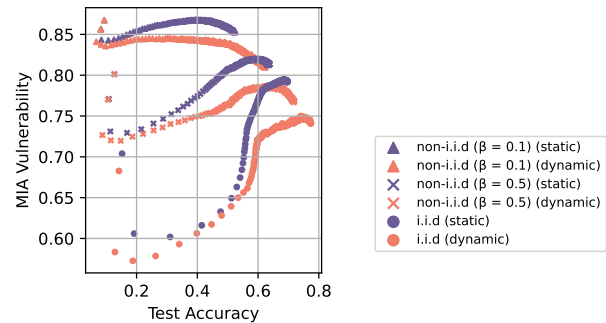


Figure 5. Trade-off between MIA vulnerability and Test Accuracy on the Purchase100 with varying degree of heterogeneity (iid, dirichlet $\beta = 0.5$ and $\beta = 0.1$), comparing static and dynamic topology setups, on a 2-regular graph with 150 nodes

others trained on different node data. This is why, we investigate the connection between generalization error and MIA vulnerability in our fully decentralized context. The results, depicted in Figure 6, illustrate the tradeoff between average MIA vulnerability and average generalization error. Each point in the plot represents an evaluation during a fixed round, with the x-axis showing the average generalization error and the y-axis showing the corresponding average MIA vulnerability. The findings reveal dataset-specific behaviors. For example, CIFAR-10 demonstrates an intuitive relationship between generalization error and MIA vulnerability. However, CIFAR-100 and Fashion-MNIST exhibit scenarios where, for a fixed generalization error, MIA vulnerability varies significantly. This phenomenon is even more pronounced for Purchase100, where distinct regimes of MIA vulnerability are observed for the same generalization error.

To delve deeper, Figure 7 highlights the tradeoff across training rounds. We observe that leakage introduced during an earlier round cannot be mitigated simply by improving the generalization error in subsequent rounds. Once a model overfits, increasing test accuracy and reducing generalization error do not eliminate the existing MIA vulnerability. Our interpretation suggests that the peak generalization error, typically occurring early in the training process, determines the maximum MIA vulnerability observed later.

Takeaway: Early overfitting drives persistent MIA vulnerability, which generalization improvements cannot fully mitigate.

This observation aligns with the concept of the Critical Learning Period phenomenon explored in FL [65]. This work emphasizes that changes during the early learning phases significantly impact the best accuracy achievable by the end of training. In our work, we find experimentally a similar

pattern, but in terms of MIA vulnerability rather than accuracy.

4 Mixing with Dynamic Graphs

In Section 3, we demonstrated the benefits of dynamic settings and increased view sizes in improving both privacy and utility. Our intuition is that the mixing properties of the learning protocol play a crucial role in reducing MIA vulnerability. Therefore, in this section, we aim to provide a deeper understanding of how dynamic settings and larger view sizes enhance the mixing rate of models. To investigate the mixing properties, we leverage existing findings on the influence of the spectral properties of the weighted adjacency matrix associated with gossip exchanges [7]. To conduct the analysis, we assume an idealized scenario with synchronous iterations where all nodes simultaneously wake up to exchange their models and no local updates are performed.

The simplified analysis considers essentially a consensus task. Let $T \in \mathbb{N}$ be a fixed number of iterations. At each (synchronous) iteration $t \in \{1, \dots, T\}$, each node $i \in \mathcal{V}$ performs the following mixing step:

$$\theta_i^{(t+1)} \leftarrow \frac{1}{k+1} \sum_{j \in \mathcal{N}_i^{(t)} \cup \{i\}} \theta_j^{(t)} \quad (8)$$

Where $\theta_i^{(t)} \in \mathcal{M}$ is the model of node i at iteration t and $\theta_i^{(0)}$ is equal to the initial model $\theta_0 \in \mathcal{M}$. While it is already evident that graphs $G^{(1)}, \dots, G^{(T)}$ defined by neighborhoods $\mathcal{N}_i^{(t)}$ for all $i \in \mathcal{V}$ and $t \in \{1, \dots, T\}$ impact convergence of consensus, this is also holds for decentralized learning [13, 32].

We now analyze mixing properties. For any iteration $t \in \{1, \dots, T\}$, let $\theta^{(t)} \in \mathcal{M}^n$ be the vector whose elements are the models of each node, and $W^{(t)} \in \mathbb{R}^{n \times n}$ be the weighted adjacency matrix (or mixing matrix) of the graph used at iteration t . Specifically, for our k -regular graphs $W_{i,j}^{(t)} = 1/(k+1)$ if and only if i and j are neighbors or $j = i$, and $W_{i,j}^{(t)} = 0$ else.

Exchanges at Equation (8) can be summarized by

$$\theta^{(t+1)} \leftarrow W^{(t)} \theta^{(t)}. \quad (9)$$

It has been proven in the work of Boyd et al. [7] that if a mixing matrix $W \in \mathbb{R}^{n \times n}$ is symmetric and doubly stochastic (that is, the sum of the elements of any column or row are equal to 1), then for any vector $\theta \in \mathbb{R}^n$ we have that

$$\|W\theta - \mathbb{1}\tilde{\theta}\|_2 \leq \lambda_2(W) \|\theta - \mathbb{1}\tilde{\theta}\|_2, \quad (10)$$

where $\tilde{\theta} = \frac{1}{n} \sum_{i \in \mathcal{V}} \theta_i$ is the average of the elements of the vector, $\lambda_2(W)$ is the second largest eigenvalue of W and $\mathbb{1}$ is the vector of all ones. Essentially, [7] shows that $\lambda_2(W)$ quantifies the reduction of the distance between values (in our

case, models³) of each node and the average of all values. The smaller $\lambda_2(W)$, the better the mixing. Given that we use undirected k -regular graphs and the weights of the exchanges is equal for all edges, we have that for all $t \in \{1, \dots, T\}$, $W^{(t)}$ is doubly stochastic and symmetric and hence we can use Equation (10) in our simplified consensus protocol.

We focus on the mixing properties of the matrix W^* obtained by performing the matrix product $W^{(T)} \dots W^{(1)}$. For any vector $\theta \in \mathbb{R}^n$, $W^*\theta$ is the result of performing the sequence of mixing exchanges determined by $W^{(1)}, \dots, W^{(T)}$ over θ .

An immediate bound given by

$$\|W^*\theta - \mathbb{1}\tilde{\theta}\|_2 \leq \prod_{t=1}^T \lambda_2(W^{(t)}) \|\theta - \mathbb{1}\tilde{\theta}\|_2. \quad (11)$$

can be obtained from sequentially applying Equation (10) to each factor $W^{(t)}$ of W^* . However, Equation (11) only takes into account individual mixing properties of each $W^{(t)}$ in isolation, ignoring the effects of varying the communication graph.

Since W^* is also doubly-stochastic when $W^{(1)}, \dots, W^{(T)}$ are doubly-stochastic, a more accurate bound can be obtained by directly applying that Equation (10) using $\lambda_2(W^*)$. This captures the overall mixing properties of the graph sequence. In the *static setting*, $W^{(t)} = W$ for all $t \in \{1, \dots, T\}$ and some W initially chosen. In the *dynamic setting* all nodes are randomly permuted at each iteration. It is easy to see that for the static case $\lambda_2(W^*) = \lambda_2(W)^T$, while in the dynamic case the closed form of $\lambda_2(W^*)$ is not evident.

In Figure 8, we empirically show how $\lambda_2(W^*)$ decreases as iterations increase. We show it for the same topologies used in Section 3. All points in the curve are reported over an average of 50 runs of gossip simulation together with their standard deviation. We can see that for the same degree k , $\lambda_2(W^*)$ decreases much faster for dynamic graphs than for static graphs. Moreover, the standard deviation is negligible in the dynamic case, indicating that bad mixing scenarios occur with negligible probability.

From this analysis, we can better understand why the dynamic setting is favorable for reducing MIA vulnerability. Models in the dynamic setting tend to align more closely with consensual models and exhibit less bias toward the peculiar model of an individual client, which would otherwise increase MIA vulnerability.

5 Summary of Takeaways and Actionable Recommendations

Our experimental analysis provides several important insights into the factors influencing MIA vulnerability and

³For simplicity, we perform our analysis treating models as scalars. However, if we consider models as vectors of scalars (i.e., $\mathcal{M} = \mathbb{R}^d$ where d is the number of parameters of the model), the current analysis holds for each model parameter.

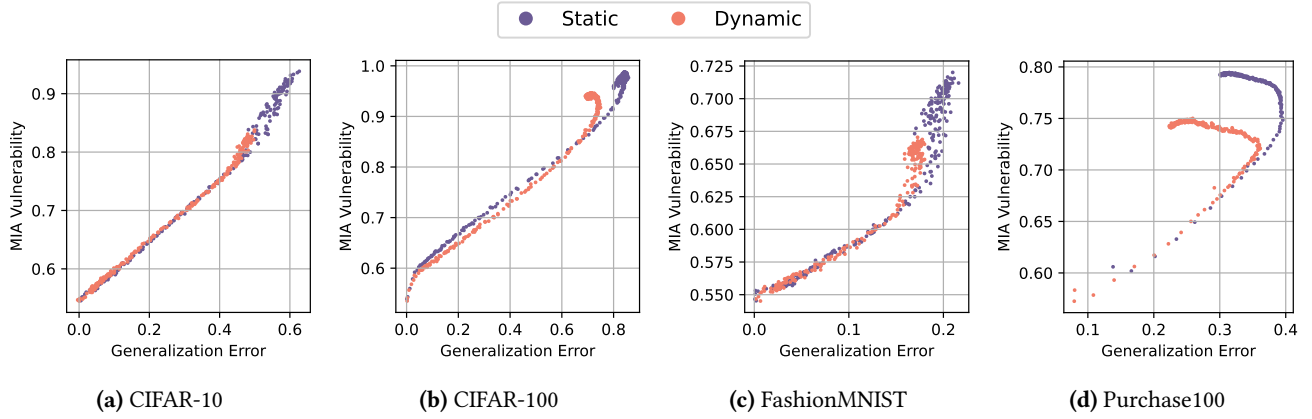


Figure 6. Trade-off between MIA vulnerability and Generalization Error across different datasets, comparing Base Gossip and SAMO learning protocols.

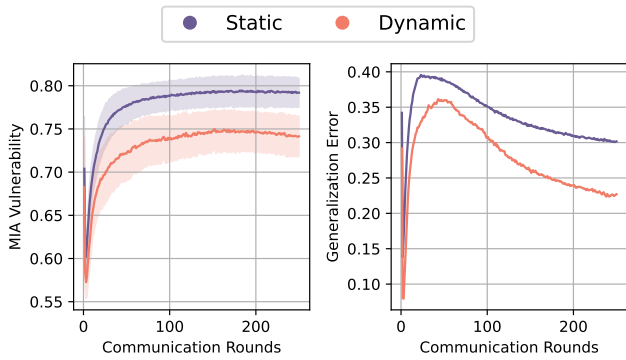


Figure 7. Average MIA Vulnerability and Generalization Error over communication rounds on Purchase100 dataset using the Merge Once Protocol, with a 2-regular graph of 150 nodes.

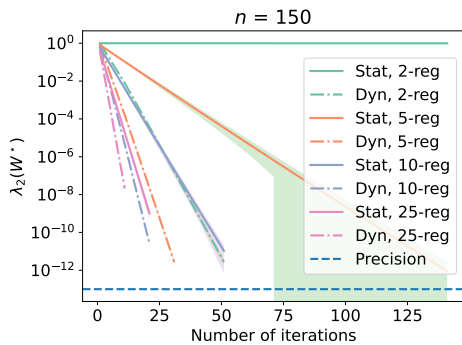


Figure 8. Values of $\lambda_2(W^*)$ depending on the number of iterations for k -regular graphs (k -reg) of different degrees in the dynamic (Dyn) and static (Stat) settings.

utility in Gossip learning. Below, by summarizing the above

takeaways, we offer actionable recommendations for designing privacy-aware decentralized learning architectures:

- **Static vs Dynamic Graph Topologies:** From our study, we recommend using dynamic graph structures to improve model mixing and mitigate privacy risks, especially as the communication network becomes large and sparse. These topologies should be paired with robust peer-sampling protocols.
- **Model Mixing:** Our study shows that incorporating advanced mixing strategies like SAMO reduces individual local biases in models and achieves better privacy-utility tradeoffs.
- **View Size:** While increasing view size is generally beneficial, it should be balanced against the increased communication cost. For practical implementations, we recommend considering dynamic settings with moderate view sizes to maximize efficiency.
- **Addressing Non-i.i.d Data:** Our study shows the high impact of non-i.i.d data distribution which asks for stronger protection mechanisms. We also recommend exploring adaptive learning protocols that account for data heterogeneity, such as adapting the ones introduced in FL [66]. For instance, using nodes pre-clustering [8] or knowledge distillation using public data [14].
- **Early Overfitting:** From our study, we recommend focusing on strategies to prevent early overfitting, such as regularization, dynamic learning rates, or delayed aggregation, to limit the persistent impact of initial vulnerabilities.

By implementing these recommendations in completion to privacy-preserving mechanisms (see Section 6.2), decentralized learning systems can achieve a more favorable balance between utility and privacy, paving the way for safer and more effective collaborative learning environments.

6 Related Works

6.1 Privacy in Decentralized and Gossip Learning.

From location-hiding [15] to source anonymity [4, 28], The inherent privacy properties of gossip protocols have captivated the research community over the past decade. However, studying how these properties translate to a learning protocol remains relatively under-explored. To the best of our knowledge, [50] is the first endeavor in this direction. In this work, the authors demonstrated that decentralized learning is at least as vulnerable as its federated counterpart to reconstruction and membership inference attacks. However, their approach relies on partial model aggregation, which we show leads to worse model mixing and, consequently, to more vulnerable models. Moreover, their results rely on assumptions such as adversaries possessing power equivalent to an FL parameter server in some contexts, or the notion that honest nodes have no honest neighbors within a two-hop radius ([50], Section 4).

These assumptions are particularly unrealistic for dynamic networks such as ours. Additionally, important characteristics of gossip learning, such as asynchronous communications and the presence of a random peer sampling protocol, are not considered. The threat model of Pasquini et al. [50] was recently extended in [45], where the authors consider adversaries operating beyond their immediate neighborhoods. While their findings are compelling, they remain limited to static and synchronous settings and further assume i.i.d data distributions. Concurrently, a recent study [26] examines the vulnerability of decentralized learning through a mutual information framework. While their theoretical results regarding the resilience of decentralized learning align with our findings, their analysis is constrained. Specifically, they exclusively consider an i.i.d setting, with static topologies and synchronous communications. Our work alleviates the assumptions made in these works.

6.2 Defenses and Protection Mechanisms

Numerous attempts have been made to enhance privacy in decentralized/gossip learning using various mechanisms. For instance, Cyffers and Bellet [10] proposes a novel formulation of differential privacy (DP) that accounts for the position of nodes within the network. In such an approach, nodes send noisy model updates with formal privacy guarantees. Then Sabater et al. [53] considers a stronger model where adversarial parties can collude and share their view. Moreover, it substantially reduces the perturbations required to achieve privacy using a correlated noise approach. While DP-based solutions are holistic from several perspectives (e.g., adversary knowledge, types of attacks), they often result in significant utility loss [9, 31] and can impede other robustness objectives [16].

Other works explore secure multi-party computation primitives. For example, Dekker et al. [12] proposes a framework

where each node performs a sequence of secure aggregations with its neighbors. Although effective, these solutions are often expensive to adapt to more general settings (e.g., Byzantine settings [56]). Finally, another line of research investigates hardware-based approaches, namely, Trusted Execution Environments (TEEs). For instance, GradSec [44] prevents privacy attacks in FL by incorporating sensitive model layers within secure compartments. However, this approach has demanding hardware requirements and has been shown to be vulnerable to side-channel attacks [27]. Overall, while we consider each of these research directions promising, they are orthogonal to our work, which focuses specifically on understanding the vulnerabilities of models in gossip learning, particularly in relation to dynamicity and mixing rate.

6.3 Studying Dynamic Communication Topologies

There are mainly two lines of work that investigate dynamic topologies in gossip learning: (i) works that attempt to quantify the impact of dynamicity on the mixing rate and convergence properties, and (ii) works that propose practical protocols leveraging peer sampling algorithms [3, 22, 23]. In the mixing rate and convergence line of work, several studies have examined the use of dynamic topologies in decentralized learning, with the majority focusing on the synchronous case [47, 59]. A noteworthy example is Epidemic Learning [11], which employs a randomly formed communication topology where, in each round, a node transmits its model to a randomly selected set of peers without coordination between them. The study showed that randomness in peer sampling positively impacts convergence speed, aligning with previous results [59].

Corroborating these findings, many practical peer-sampling protocols have been proposed. For instance, Hegedús et al. [23] demonstrate empirically that gossip learning, enhanced with a peer sampling protocol, can systematically compete with federated settings across various configurations and hyperparameters (e.g., message loss, data distributions). Similarly, Belal et al. [3] observe that a carefully designed peer sampling protocol can improve model personalization. Overall, while our work aligns to some extent with both research directions, it introduces a novel perspective by establishing a crucial connection between the mixing properties of a practical dynamic protocol and its vulnerability to membership inference attacks. To the best of our knowledge, such a connection has not been explored in prior research.

7 Conclusion

In this work, we presented a comprehensive study of MIA vulnerabilities in decentralized learning systems. By analyzing the interplay between model mixing properties, graph dynamics, and data distributions, we provided new insights into the privacy risks inherent in these systems.

Our findings revealed that mixing properties are critical in shaping MIA vulnerabilities. Dynamic graph topologies, coupled with increased view sizes, enhance mixing rates, leading to better privacy-utility tradeoffs. However, the effectiveness of these approaches is constrained when faced with non-i.i.d data distributions, which amplify MIA risks across all training rounds. Importantly, we demonstrated that early overfitting creates persistent vulnerabilities that cannot be fully mitigated by improving generalization later in the training process.

These insights underline the need for careful design in decentralized learning protocols. By favoring dynamicity and improving model mixing, systems can achieve a better balance between utility and privacy. Future work should focus on developing adaptive mechanisms to address the challenges posed by heterogeneous data distributions and early overfitting, as well as exploring complementary defense techniques, such as differential privacy and secure computation.

Through our analysis and experiments, we provide actionable recommendations for building more privacy-resilient decentralized learning systems, paving the way for safer collaborative learning environments.

Acknowledgments

This work was supported by the French government managed by the Agence Nationale de la Recherche (ANR) through France 2030 program with the reference ANR-23-PEIA-005 (REDEEM project).

References

- [1] Syreen Banabilah, Moayad Aloqaily, Eitaa Alsayed, Nida Malik, and Yaser Jararweh. 2022. Federated learning review: Fundamentals, enabling technologies, and future applications. *Information processing & management* 59, 6 (2022), 103061.
- [2] Paul Baran. 1964. On distributed communications networks. *IEEE transactions on Communications Systems* 12, 1 (1964), 1–9.
- [3] Yacine Belal, Aurélien Bellet, Sonia Ben Mokhtar, and Vlad Nitu. 2022. PEPPER: Empowering User-Centric Recommender Systems over Gossip Learning. <https://doi.org/10.48550/arXiv.2208.05320> arXiv:2208.05320
- [4] Aurélien Bellet, Rachid Guerraoui, and Hadrien Hendrikx. 2020. Who started this rumor? Quantifying the natural differential privacy of gossip protocols. In *34th International Symposium on Distributed Computing (DISC 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
- [5] Aurélien Bellet, Rachid Guerraoui, Mahsa Taziki, and Marc Tommasi. 2017. *Fast and differentially private algorithms for decentralized collaborative machine learning*. Ph. D. Dissertation. INRIA Lille.
- [6] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. 2019. Analyzing federated learning through an adversarial lens. In *International conference on machine learning*. PMLR, 634–643.
- [7] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. 2006. Randomized gossip algorithms. *IEEE transactions on information theory* 52, 6 (2006), 2508–2530.
- [8] Christopher Briggs, Zhong Fan, and Peter Andras. 2020. Federated learning with hierarchical clustering of local updates to improve training on non-IID data. In *2020 international joint conference on neural networks (IJCNN)*. IEEE, 1–9.
- [9] Alycia N Carey, Minh-Hao Van, and Xintao Wu. 2024. Evaluating the impact of local differential privacy on utility loss via influence functions. In *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–10.
- [10] Edwige Cyffers and Aurélien Bellet. 2022. Privacy amplification by decentralization. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 5334–5353.
- [11] Martijn De Vos, Sadegh Farhadkhani, Rachid Guerraoui, Anne-marie Kermarrec, Rafael Pires, and Rishi Sharma. 2023. Epidemic Learning: Boosting Decentralized Learning with Randomized Communication. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., New Orleans, LA, USA, 36132–36164. https://proceedings.neurips.cc/paper_files/paper/2023/file/7172e147d916eef4cb1eb30016ce725f-Paper-Conference.pdf
- [12] Florine W Dekker, Zekeriya Erkin, and Mauro Conti. 2023. Topology-Based Reconstruction Prevention for Decentralised Learning. *arXiv preprint arXiv:2312.05248* (2023).
- [13] Mathieu Even, Anastasia Koloskova, and Laurent Massoulié. 2024. Asynchronous sgd on graphs: a unified framework for asynchronous decentralized and federated optimization. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 64–72.
- [14] Xiuwen Fang and Mang Ye. 2022. Robust federated learning with noisy and heterogeneous clients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10072–10081.
- [15] Karol Gotfryd, Marek Klonowski, and Dominik Pająk. 2017. On location hiding in distributed systems. In *International Colloquium on Structural Information and Communication Complexity*. Springer, 174–192.
- [16] Rachid Guerraoui, Nirupam Gupta, Rafael Pinot, Sébastien Rouault, and John Stephan. 2021. Differential Privacy and Byzantine Resilience in SGD: Do They Add Up? *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing* (2021). <https://api.semanticscholar.org/CorpusID:231934098>
- [17] Rachid Guerraoui, Anne-Marie Kermarrec, Anastasiia Kucherenko, Rafael Pinot, and Marijn de Vos. 2024. PeerSwap: A Peer-Sampler with Randomness Guarantees. *ArXiv abs/2408.03829* (2024). <https://api.semanticscholar.org/CorpusID:271745114>
- [18] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604* (2018).
- [19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV) (ICCV '15)*. IEEE Computer Society, USA, 1026–1034. <https://doi.org/10.1109/ICCV.2015.123>
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (Las Vegas, NV, USA) (CVPR '16)*. IEEE, 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- [21] István Hegedűs, Gábor Danner, and Márk Jelasity. 2019. Gossip Learning as a Decentralized Alternative to Federated Learning. In *Distributed Applications and Interoperable Systems (Lecture Notes in Computer Science, Vol. 11665)*. IFIP, Springer, Copenhagen, Denmark, 74–90. https://doi.org/10.1007/978-3-030-22496-7_5
- [22] István Hegedűs, Gábor Danner, and Márk Jelasity. 2019. Gossip learning as a decentralized alternative to federated learning. In *Distributed Applications and Interoperable Systems: 19th IFIP WG 6.1 International Conference, DAIS 2019, Held as Part of the 14th International Federated Conference on Distributed Computing Techniques, DisCoTec 2019, Kongens Lyngby, Denmark, June 17–21, 2019, Proceedings 19*. Springer, 74–90.

- [23] István Hegedűs, Gábor Danner, and Márk Jelasity. 2021. Decentralized Learning Works: An Empirical Comparison of Gossip Learning and Federated Learning. *J. Parallel and Distrib. Comput.* 148 (Feb. 2021), 109–124. <https://doi.org/10.1016/j.jpdc.2020.10.006>
- [24] Hadrien Hendrikx, Francis Bach, and Laurent Massoulié. 2019. Accelerated decentralized optimization with local updates for smooth and strongly convex objectives. In *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 897–906.
- [25] Márk Jelasity, Spyros Voulgaris, Rachid Guerraoui, Anne-Marie Ker-marrec, and Maarten Van Steen. 2007. Gossip-based peer sampling. *ACM Transactions on Computer Systems (TOCS)* 25, 3 (2007), 8–es.
- [26] Changlong Ji, Stephane Maag, Richard Heusdens, and Qiongxiu Li. 2024. Re-Evaluating Privacy in Centralized and Decentralized Learning: An Information-Theoretical and Empirical Study. *arXiv preprint arXiv:2409.14261* (2024).
- [27] Jianyu Jiang, Claudio Soriente, and Ghassan Karamé. 2022. On the challenges of detecting side-channel attacks in SGX. In *Proceedings of the 25th International Symposium on Research in Attacks, Intrusions and Defenses*. 86–98.
- [28] Richeng Jin, Yufan Huang, Zhaoyang Zhang, and Huaiyu Dai. 2023. On the privacy guarantees of gossip protocols in general networks. *IEEE Transactions on Network Science and Engineering* 10, 6 (2023), 3114–3130.
- [29] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and trends® in machine learning* 14, 1–2 (2021), 1–210.
- [30] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*. PMLR, 5132–5143.
- [31] Muah Kim, Onur Günlü, and Rafael F Schaefer. 2021. Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication. In *ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2650–2654.
- [32] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. 2020. A unified theory of decentralized sgd with changing topology and local updates. In *International Conference on Machine Learning*. PMLR, 5381–5393.
- [33] Noam Koren, Abigail Goldstein, Guy Amit, and Ariel Farkash. 2024. Membership Inference Attacks Against Time-Series Models. In *The 16th Asian Conference on Machine Learning (Conference Track)*.
- [34] Alex Krizhevsky and Geoffrey Hinton. 2009. *Learning multiple layers of features from tiny images*. Technical Report 0. University of Toronto, Toronto, Ontario. <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>
- [35] Chengxi Li, Gang Li, and Pramod K. Varshney. 2022. Decentralized Federated Learning via Mutual Knowledge Transfer. *IEEE Internet of Things Journal* 9, 2 (Jan. 2022), 1136–1147. <https://doi.org/10.1109/JIOT.2021.3078543>
- [36] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. 2022. Federated Learning on Non-IID Data Silos: An Experimental Study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE Computer Society, Los Alamitos, CA, USA, 965–978. <https://doi.org/10.1109/ICDE53745.2022.00077>
- [37] Qinbin Li, Bingsheng He, and Dawn Song. 2021. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10713–10722.
- [38] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine* 37, 3 (2020), 50–60.
- [39] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. 2017. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (Long Beach, California, USA) (NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 5336–5346.
- [40] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. 2017. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. *Advances in neural information processing systems* 30 (2017).
- [41] NA Lynch. 1996. *Distributed Algorithms*. Morgan Kaufmann Publishers.
- [42] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera Y. Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 54)*, Aarti Singh and Jerry Zhu (Eds.). PMLR, Fort Lauderdale Florida, USA, 1273–1282.
- [43] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)*. IEEE, 691–706.
- [44] Aghiles Ait Messaoud, Sonia Ben Mokhtar, Vlad Nitu, and Valerio Schiavoni. 2021. GradSec: a TEE-based Scheme Against Federated Learning Inference Attacks. In *Proceedings of the First Workshop on Systems Challenges in Reliable and Secure Federated Learning (Virtual Event, Germany) (ResilientFL '21)*. Association for Computing Machinery, New York, NY, USA, 10–12. <https://doi.org/10.1145/3477114.3488763>
- [45] Abdellah El Mrini, Edwige Cyffers, and Aurélien Bellet. 2024. Privacy Attacks in Decentralized Learning. *arXiv preprint arXiv:2402.10001* (2024).
- [46] 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)*. IEEE, 739–753.
- [47] Angelia Nedic and Alex Olshevsky. 2014. Distributed Optimization over Time-Varying Directed Graphs. <https://doi.org/10.48550/arXiv.1303.2289> arXiv:1303.2289 [math]
- [48] Julien Nicolas, César Sabater, Mohamed Maouche, Sonia Ben Mokhtar, and Mark Coates. 2024. Differentially private and decentralized randomized power method. *arXiv preprint arXiv:2411.01931* (2024).
- [49] Róbert Ormándi, István Hegedűs, and Márk Jelasity. 2013. Gossip learning with linear models on fully distributed data. *Concurrency and Computation: Practice and Experience* 25, 4 (2013), 556–571. <https://doi.org/10.1002/cpe.2858> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.2858>
- [50] Dario Pasquini, Mathilde Raynal, and Carmela Troncoso. 2023. On the (In)security of Peer-to-Peer Decentralized Machine Learning. <http://arxiv.org/abs/2205.08443> arXiv:2205.08443 [cs].
- [51] Mirko Polato. 2024. makgyver/gossipy. <https://github.com/makgyver/gossipy>
- [52] Pian Qi, Diletta Chiaro, Antonella Guzzo, Michele Ianni, Giancarlo Fortino, and Francesco Piccialli. 2024. Model aggregation techniques in federated learning: A comprehensive survey. *Future Generation Computer Systems* 150 (2024), 272–293.
- [53] César Sabater, Aurélien Bellet, and Jan Ramon. 2022. An accurate, scalable and verifiable protocol for federated differentially private averaging. *Machine Learning* 111, 11 (2022), 4249–4293.
- [54] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2019. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models.. In *NDSS*. The Internet Society. <http://dblp.uni-trier.de/db/conf/ndss/ndss2019.html#Salem0HBF019>
- [55] 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)*. IEEE, IEEE

- Computer Society, San Jose, CA, USA, 3–18. <https://doi.org/10.1109/SP.2017.41>
- [56] Jinhyun So, Başak Güler, and A Salman Avestimehr. 2020. Byzantine-resilient secure federated learning. *IEEE Journal on Selected Areas in Communications* 39, 7 (2020), 2168–2181.
- [57] Liwei Song and Prateek Mittal. 2020. *Systematic Evaluation of Privacy Risks of Machine Learning Models*. <https://doi.org/10.48550/arXiv.2003.10595> arXiv:2003.10595
- [58] Liwei Song, Reza Shokri, and Prateek Mittal. 2019. Membership Inference Attacks Against Adversarially Robust Deep Learning Models. In *2019 IEEE Security and Privacy Workshops (SPW)*. IEEE Computer Society, Los Alamitos, CA, USA, 50–56. <https://doi.org/10.1109/SPW.2019.00021>
- [59] Zhuoqing Song, Weijian Li, Kexin Jin, Lei Shi, Ming Yan, Wotao Yin, and Kun Yuan. 2022. Communication-Efficient Topologies for Decentralized Learning with $\mathcal{O}(1)$ Consensus Rate. 35 (2022), 1073–1085. https://proceedings.neurips.cc/paper_files/paper/2022/hash/0790ef700dd0072f4940abda9b7d0005-Abstract-Conference.html
- [60] Robbert Van Renesse, Kenneth P Birman, and Werner Vogels. 2003. Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining. *ACM transactions on computer systems (TOCS)* 21, 2 (2003), 164–206.
- [61] Spyros Voulgaris, Daniela Gavidia, and Maarten Van Steen. 2005. Cyclon: Inexpensive membership management for unstructured p2p overlays. *Journal of Network and systems Management* 13 (2005), 197–217.
- [62] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. 2019. Adaptive federated learning in resource constrained edge computing systems. *IEEE journal on selected areas in communications* 37, 6 (2019), 1205–1221.
- [63] Pengjun Wu, Zhanzhi Zhang, Xueyi Peng, and Ran Wang. 2024. Deep learning solutions for smart city challenges in urban development. *Scientific Reports* 14, 1 (2024), 5176.
- [64] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. arXiv. arXiv:cs.LG/1708.07747 [cs.LG]
- [65] Gang Yan, Hao Wang, and Jian Li. 2022. Seizing Critical Learning Periods in Federated Learning. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 8 (Jun. 2022), 8788–8796. <https://doi.org/10.1609/aaai.v36i8.20859>
- [66] Mang Ye, Xiuwen Fang, Bo Du, Pong C Yuen, and Dacheng Tao. 2023. Heterogeneous federated learning: State-of-the-art and research challenges. *Comput. Surveys* 56, 3 (2023), 1–44.
- [67] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. <https://doi.org/10.48550/arXiv.1709.01604> arXiv:1709.01604
- [68] Jingwen Zhang, Jiale Zhang, Junjun Chen, and Shui Yu. 2020. GAN Enhanced Membership Inference: A Passive Local Attack in Federated Learning. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. 1–6. <https://doi.org/10.1109/ICC40277.2020.9148790>