**Circuits and Systems**
Mekelweg 4,
2628 CD Delft
The Netherlands
https://sps.ewi.tudelft.nl/

**SPS-2025-5177855**

# M.Sc. Thesis

# Privacy Robustness Trade-Off Analysis in Decentralised Federated learning

**Zarè Palanciyan**

**Abstract**

Federated learning (FL) enables collaborative model training across multiple clients without sharing raw data, offering a promising solution for privacy-sensitive applications. However, as FL becomes more decentralised, balancing data privacy with resilience against adversarial attacks remains a fundamental challenge. This thesis investigates the interplay between privacy-preserving mechanisms such as Differential Privacy, Secure Multi-Party Computation (SMPC), and Subspace Perturbation, and the robustness of adversarial detection in fully decentralised FL networks. By extending information-theoretic bounds and conducting comprehensive experiments under a variety of attack scenarios, we show that stronger privacy guarantees often come at the cost of reduced detection capability. Notably, mechanisms that increase noise or mask updates to protect data privacy tend to obscure the test statistics that detectors rely on, resulting in higher false alarm rates and missed detections. Our results highlight that while privacy and robustness cannot be maximised simultaneously, careful tuning of system parameters and defence strategies can help achieve a practical balance. This work provides theoretical insights and empirical evidence to inform the deployment of privacy-preserving and robust federated learning systems.

# Privacy Robustness Trade-Off Analysis in Decentralised Federated learning

Thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science

in

Electrical Engineering

by

Zarè Palanciyan
born in Amsterdam, The Netherlands

This work was performed in:

Circuits and Systems Group
Department of Microelectronics
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

**Delft University of Technology**

Delft University of Technology
Department of
Microelectronics

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled **"Privacy Robustness Trade-Off Analysis in Decentralised Federated learning"** by **Zarè Palanciyan** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: My Graduation Date

Chairman: _____
prof.dr.ir. R. Heusdens

Advisor: _____
dr. Q. Li

Committee Members: _____
dr. Z. Erkin

_____

# Acknowledgments

I would like to thank my advisors prof.dr.ir. R. Heusdens and dr. Q. Li for their assistance during the writing of this thesis, and my family and friends for their support; without them, this would not have been possible.

Zarè Palanciyan
Delft, The Netherlands
My Graduation Date

*"KISS – Keep It Simple Stupid."*
—Kelly Johnson

# Contents

# List of Figures

# List of Tables

# 1 Introduction

In recent years, there has been a growing interest in developing methods for distributed computations, which facilitate collaborative data processing across decentralised nodes without relying on a centralised coordinator [2]. Such methods have found extensive applications in fields including wireless sensor networks [3], optimisation [4], and federated learning [5]. Distributed optimisation techniques, notably the Alternating Direction Method of Multipliers (ADMM) [6] and the Primal-Dual Method of Multipliers (PDMM) [7, 8, 9], have increasingly been adopted due to their effectiveness and scalability. As these distributed computation algorithms are deployed more broadly, they inevitably process sensitive data, raising critical privacy concerns [10]. Traditional privacy-preserving methods, such as Differential Privacy (DP) [11, 12] and Secure Multi Party Computation (SMPC) [13], offer protection but often incur a significant trade-off with accuracy or computational efficiency [14, 15]. To mitigate these drawbacks, new frameworks, like Subspace Perturbation (SP) [16, 17, 18], have been proposed, which strive to balance privacy preservation with minimal impact on computational performance and accuracy. However, privacy preservation is not the sole challenge in decentralised systems. Adversarial threats, where malicious nodes inject corrupt data or disruptive updates into the network, pose significant risks to the integrity and reliability of these distributed systems [19]. Such threats include a variety of attack strategies, including Gaussian noise attacks, model poisoning, and label flipping attacks [20, 21, 22]. Consequently, robust detection mechanisms, such as Krum [23], Kardam [24], and deviation based detection methods [25], have been developed to identify and mitigate malicious influences within decentralised networks. In distributed computation environments, achieving optimal outcomes requires mechanisms that simultaneously maintain privacy and effectively detect adversarial behaviours. Detection algorithms typically rely on distinguishing individual data updates from collective network behaviour, highlighting deviations that indicate adversarial activity. However, this detection capability fundamentally conflicts with privacy preservation methods, such as noise injection, which obscure node specific details essential for accurate adversarial detection.

## 1.1 Research Question and Outline

This thesis investigates the critical trade-off between privacy preservation and robust adversarial detection within decentralised federated learning frameworks. It analyses how the application of privacy preserving mechanisms influences adversarial detection effectiveness and explores the associated implications through comprehensive simulations. The goal is to provide insights into the inherent conflict between privacy and robustness, assisting informed design choices for secure and effective decentralised learning systems.

The remainder of this thesis is structured as follows: Chapter 2 provides foundational background on decentralised federated learning. Chapter 3 introduces privacy preserving mechanisms evaluated in this research. Chapter 4 details robustness mechanisms and adversarial threats relevant to decentralised environments, and theoretically analyses the privacy-robustness trade-off. Chapter 5 presents simulation results validating the trade-off under realistic conditions. Finally, Chapter 6 concludes with a summary of findings and proposes directions for future research.

# 2 Background

This chapter establishes the fundamental concepts and terminology that describe the analysis of federated learning in adversarial environments. We begin by formalising the network topology and notation used to describe decentralised systems, introducing the key roles played by honest, adversarial, and eavesdropping nodes. The adversarial model is specified to distinguish between passive and active threats, providing a clear threat landscape for privacy and robustness considerations.

## 2.1 Topology and Notation

Federated learning can be used in many different decentralised settings. These decentralised settings, such as wireless networks, can be modelled with a graph. We present a simple undirected connected graph $G$ as $G = (\mathcal{V}, \mathcal{E})$, where the set of nodes in the network is represented by $\mathcal{V} = \{1, 2, \ldots, n\}$ and the set of edges is represented by $\mathcal{E} = \{e_1, \ldots, e_m\} \subseteq \mathcal{V} \times \mathcal{V}$. The neighbourhood of node $i$ is denoted as the set $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$. The degree of node $i$ is then given by $d_i = |\mathcal{N}_i|$.

### 2.1.1 Participating Nodes

Let $\mathcal{V}_h$ denote the set of honest nodes and $\mathcal{V}_c$ the set of adversarial nodes such that $\mathcal{V} = \mathcal{V}_h \cup \mathcal{V}_c$ and $\mathcal{V}_h \cap \mathcal{V}_c = \emptyset$. In addition, let $\mathcal{V}_{c,p}$ denote the set of passive adversarial nodes and $\mathcal{V}_{c,a}$ the set of active adversarial nodes so that $\mathcal{V}_c = \mathcal{V}_{c,p} \cup \mathcal{V}_{c,a}$ and $\mathcal{V}_{c,p} \cap \mathcal{V}_{c,a} = \emptyset$. In addition to the nodes within the graph, we consider a set of external eavesdropping nodes $\mathcal{B}$, for which $\mathcal{B} \cap \mathcal{V} = \emptyset$. The behaviour of these mentioned nodes will be described as shown below.

- **Honest nodes** ($\mathcal{V}_h$): Follow the federated learning protocol correctly without malicious intent.
- **Passive adversarial nodes** ($\mathcal{V}_{c,p}$): (Honest-but-curious) Follow the federated learning protocol, but infer private information from updates.
- **Active adversarial nodes** ($\mathcal{V}_{c,a}$): (Malicious) Disrupt training via faulty updates, data poisoning, or backdoor attacks.
- **Eavesdropping nodes** ($\mathcal{B}$): Infer information from public transmissions without participating in federated learning.

### 2.1.2 Primal Dual Method of Multipliers (PDMM) for Decentralised Federated Learning

Distributed machine learning faces significant challenges when data is across decentralised nodes with statistical heterogeneity (non-IID data) and communication constraints. Centralised approaches like federated learning [5] rely on server based aggregation, which introduces bottlenecks and privacy risks. Agent to agent alternatives, such as PDMM address these limitations [26] for decentralised deep learning. The Primal Dual method of multipliers (PDMM), has been widely used for many different distributed optimisation problems [27, 28]. It has also been shown that PDMM can be implemented to train machine learning models in a distributed fashion [29].

## 2.2 Federated Learning FL

Federated learning (FL) is a term introduced by McMahan. It is a direct application of the principle of data minimisation proposed by a 2012 report on the privacy of consumer data [30] [31]. In FL, a global model is collaboratively trained across $K$ edge devices, each with their own local private data. These devices then aggregate their trained models with each other to generate a global model based on all the private data of the edge devices. Rather than transmitting raw data, each client performs local model updates based on its own dataset, and only the resulting updates (model parameters) are shared with the aggregator. These updates are then combined to form an improved global model. This way, a model can be trained and distributed on data that is not allowed to be transmitted due to restrictions. This subset of machine learning can thus be applied to sectors where local institutions have a low amount of training data, but simultaneously are not allowed to transmit, due to restrictions, to other institutions with the same goal. Each client $k \in \{1, \ldots, K\}$ maintains a local dataset $(x, y)_k$, where $x_k$ represents the observed input features and $y_k$ the corresponding labels. The client defines a local loss function $f_k(x, y; w)$, which evaluates the model's performance on its own data. Each client's goal is to minimise their local loss function by optimising the shared model parameters $w \in \mathbb{R}^d$ by minimising this local objective.

(a) Centralised (star) topology

(b) Decentralized topology

Figure 2.1: Two topologies for federated learning showing the model parameter communication.

Depending on the topology of the network of the devices, the FL framework can be separated into 2 sections. If the network has a star topology, meaning a single central aggregator, this is considered to be centralised federated learning (CFL). But if the network does not have a central aggregator, it is considered to be decentralised federated learning (DFL), as is shown in Figure 2.1.

### 2.2.1 DFL: FedAVG

DFL comprises a class of protocols that enable distributed nodes to train a machine learning model without a central server collaboratively. Among these protocols, the first and most widely used is the FedAVG protocol [1]. After local training, the nodes exchange their model parameters with their neighbours and perform an averaging step to update their local models. This local training and parameter exchange process is repeated iteratively. At each communication round, the model on each node becomes an admixture of its own and its neighbour's knowledge, converging to a global consensus. The learning continues until a convergence criterion is met.

**Algorithm 1** Decentralised Federated Averaging (DFL-FedAVG)

**Require:** Initial models $\{w_i^{(0)}\}_{i=1}^{K}$, learning rate $\eta$, local epochs $H$, communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
**for** each round $t = 0, 1, 2, \ldots, T-1$ **do**
  **for** each node $i \in \mathcal{V}$ **in parallel do**
    Initialize $w_i^{(t,0)} \leftarrow w_i^{(t)}$
    **for** epoch $h = 1$ to $H$ **do**
      Sample a mini-batch $\mathcal{B}_i$ from local data
      Compute gradient: $g_i \leftarrow \nabla \ell(w_i^{(t,h-1)}; \mathcal{B}_i)$
      Local update: $w_i^{(t,h)} \leftarrow w_i^{(t,h-1)} - \eta g_i$
    **end for**
    Set $\tilde{w}_i^{(t+1)} \leftarrow w_i^{(t,H)}$ ▷ Local model after training
  **end for**
  **for** each node $i \in \mathcal{V}$ **in parallel do**
    Receive $\tilde{w}_j^{(t+1)}$ from all $j \in \mathcal{N}_i$
    Update model: $w_i^{(t+1)} \leftarrow \sum_{j \in \mathcal{N}_i \cup \{i\}} a_{ij} \tilde{w}_j^{(t+1)}$
  **end for**
**end for**

FedAVG is popular due to its simplicity, scalability, and ability to reduce communication costs by performing multiple local updates before synchronisation. However, its performance in a decentralised setting heavily depends on the topology of the communication graph, the heterogeneity of local data distributions (non-IID), and the choice of local update parameters like learning rate and epoch count [32, 33].

[1] FedAVG can also be applied in a centralised topology, where updates are aggregated by a central server.

### 2.2.2 DFL: Optimisation-Based

Another method of federated learning, which has been gaining traction, is the optimisation-based decentralised federated learning [29]. Instead of using model averaging like in FedAVG, these methods treat the global learning objective as a constrained optimisation problem and solve it using distributed optimisation algorithms such as ADMM or PDMM.

In these protocols, each node maintains local copies of the model and dual variables and iteratively minimizes its local objective function augmented with consensus-enforcing terms. The updates involve solving a regularized local subproblem, followed by a dual variable update through message passing with neighbours. The pseudocode below outlines the synchronous version of ADMM/PDMM for decentralised optimisation.

**Algorithm 2** Synchronous ADMM/PDMM

**Initialization** of $z^{(0)}$
**for** $t = 0, 1, \ldots$ **do**
  **for each** node $i \in \mathcal{V}$ **in parallel do**
    $w_i^{(t+1)} = \arg\min_{w_i} \left( f_i(w_i) + \sum_{j \in \mathcal{N}_i} {z_{i|j}^{(t)}}^\top B_{i|j} w_i + \frac{\rho d_i}{2} w_i^2 \right)$
    **for each** $j \in \mathcal{N}_i$ **do**
      $y_{i|j}^{(t+1)} = z_{i|j}^{(t)} + 2\rho B_{i|j} w_i^{(t+1)}$
    **end for**
  **end for**
  **for each** $i \in \mathcal{V}, j \in \mathcal{N}_i$ **do**
    $\text{Node}_j \leftarrow \text{Node}_i(y_{i|j}^{(t+1)})$
  **end for**
  **for each** $i \in \mathcal{V}, j \in \mathcal{N}_i$ **do**
    $z_{j|i}^{(t+1)} = (1-\theta) z_{j|i}^{(t)} + \theta y_{i|j}^{(t+1)}$
  **end for**
**end for**

Optimisation-based methods are particularly useful in scenarios with heterogeneous data distributions or where stronger convergence properties are required. They also enable incorporating constraints or regularisation more naturally than averaging-based methods like FedAVG.

# 3 Preliminary: Privacy

This chapter provides a foundation for privacy in distributed systems. We begin by formalising privacy by introducing mutual information as a quantifiable metric for privacy loss. The chapter then introduces widely used privacy preserving mechanisms and highlighting their formal guarantees and inherent trade-off between privacy and utility.

## 3.1 Mutual Infomration

Because privacy depends on how much information can be inferred from available data, a key metric is mutual information (MI) from information theory, which measures how much uncertainty of a random variable is reduced by the knowledge of another. As a privacy metric, mutual information quantifies the dependence between a node $i$'s private data ($X_i$) and the observations ($\mathcal{O}$) accessible to an adversary. It is given by Equation 3.1.

$$I(X_i; \mathcal{O}) = H(X_i) - H(X_i|\mathcal{O}) \tag{3.1}$$

Here, the entropy of $X_i$ is quantified by $H(X_i)$, which is the uncertainty of the private data $X_i$, based on its probability distribution $p(x)$. The entropy of $X_i$ is given by Equation 3.2.

$$H(X_i) = -\sum_{x \in X_i} p(x) \log p(x), \tag{3.2}$$

The conditional entropy is denoted by $H(X_i|\mathcal{O})$, which is the uncertainty of $X_i$ given the adversary's observations $\mathcal{O}$. Thus, MI is the reduction in uncertainty about $X_i$ gained from $\mathcal{O}$. The conditional entropy is given by Equation 3.3.

$$H(X_i|\mathcal{O}) = -\sum_{o \in \mathcal{O}} p(o) \sum_{x \in \mathcal{X}} p(x|o) \log p(x|o). \tag{3.3}$$

When $I(X_i; \mathcal{O}) = 0$, it implies that the adversary gains no information about $X_i$ from their observations, ensuring perfect privacy. Conversely, if $I(X_i; \mathcal{O}) = H(X_i)$, the adversary has full knowledge of the private data, indicating complete privacy loss. Therefore, a larger mutual information value corresponds to a higher degree of privacy leakage. For high-dimensional $\mathcal{O}$ ($\mathcal{O} \in \mathbb{R}^d$ where $d \gg 1$), the computation of $I(X_i; \mathcal{O})$ is intractable due to:

- The curse of dimensionality: Density estimation for $p(x, o)$ and $p(o)$ requires exponentially many samples[34].
- Integration over large spaces: Summing over all $o \in \mathcal{O}$ becomes infeasible[34].

This is commonly the case when trying to use the mutual information metric in a federated learning environment, as the transmitted data is commonly high-dimensional. For this reason, heuristic metrics, such as the private data reconstruction error, are used to prove the privacy of a framework in federated learning. This way it is still possible to asses how much privacy a framework has based on the magnitude of the reconstruction error. If this error converges to the optimal value, the adversary can perfectly reconstruct the private data, meaning that the privacy has been violated. In contrast, when this reconstruction error diverges, the privacy of the framework has not been violated, and the private data can still be considered private. A commonly used metric for federated learning to asses the reconstruction error in practice, when the private data are images which will be the norm in this thesis, is the structural similarity (SSIM) between the original private images and the reconstructed images [35].

## 3.2 Privacy Preserving Mechanisms

### 3.2.1 Differential Privacy

Differential privacy is a privacy-preserving mechanism that ensures that individual data points in a dataset cannot be identified or extracted. This is achieved by injecting noise into the data before releasing the output of any analysis. The introduction of this noise enhances privacy but inherently introduces a trade-off between accuracy and privacy [36]. The formal definition of differential privacy is given in Equation 3.4. There exists a randomised mechanism $\mathcal{M}$ which guarantees $\epsilon$-differential privacy for any pair of neighbouring datasets[1], $D$ and $D'$. The mechanism satisfies differential privacy if, for any measurable subset $S$ of possible outputs, the probability of producing an output in $S$ does not differ by more than a factor of $e^{\epsilon}$:

$$\forall S \subseteq \text{Range}(\mathcal{M}): \quad \frac{\Pr[\mathcal{M}(D) \in S]}{\Pr[\mathcal{M}(D') \in S]} \leq e^{\epsilon} \tag{3.4}$$

A smaller value of the privacy budget $\epsilon$ implies a stronger privacy guarantee, as the output distributions for $D$ and $D'$ become more similar and harder to distinguish. In practice, differential privacy is implemented by adding noise to the query outputs. The scale of this noise depends on the sensitivity of the query and the desired privacy level. A common distribution, to create noise, used in differential privacy is the Laplace distribution. The noise that is to be inserted is dependent on how much the output of the function can change when a single entry has been changed. This is called the $\ell_1$-sensitivity and is defined as shown in Equation 3.5.

$$\Delta f = \max_{D,D'} \|f(D) - f(D')\|_1 \tag{3.5}$$

For a function $f(D)$ with $\ell_1$-sensitivity $\Delta f$, the differential privacy mechanism outputs:

$$\mathcal{M}(D) = f(D) + \eta, \quad \text{where } \eta \sim \text{Laplace}\left(0, \frac{\Delta f}{\epsilon}\right) \tag{3.6}$$

[1]Neighbouring datasets are datasets that differ in at most one element.

The expected error introduced by this mechanism, also referred to as the mean squared error (MSE), is proportional to the variance of the Laplace noise. When applying this to an average over $n$ users with values bounded in $[\alpha, \alpha + M]$, the variance of each noise term is:

$$\mathrm{Var}(\eta_i) = \frac{2M^2}{\epsilon^2} \tag{3.7}$$

Hence, the overall MSE of the noisy average is given by:

$$e_{\mathrm{DP}} = \frac{1}{n^2} \sum_{i=1}^{n} \mathrm{Var}(\eta_i) = \frac{2M^2}{n\epsilon^2} \tag{3.8}$$

This shows that as the privacy budget $\epsilon$ decreases (stronger privacy), the error increases, reflecting the fundamental trade-off between privacy and accuracy in differentially private mechanisms.

**Information-Theoretic Proof of Perfect Privacy for Differential Privacy** Let $D$ and $D'$ be any neighbouring data sets and let $\mathcal{M}$ be an $\epsilon$-differentially private mechanism. For $\epsilon = 0$ the defining inequality in Equation 3.4 reduces to

$$\forall S \subseteq \mathrm{Range}(\mathcal{M}) : \qquad \Pr\big[\mathcal{M}(D) \in S\big] = \Pr\big[\mathcal{M}(D') \in S\big].$$

Hence $\mathcal{M}(D)$ has the same distribution for every database; it is statistically independent of $D$. Let $X := D$ and $Z := \mathcal{M}(D)$. Independence yields

$$I(X; Z) = H(Z) - H(Z \mid X) = H(Z) - H(Z) = 0,$$

so the adversary's mutual information about any individual entry is zero. Thus $\epsilon = 0$ achieves perfect privacy, at the expense of a higher MSE.

### 3.2.2 Secure Multi-Party Computation (SMPC)

Secure multi-party computation (SMPC) allows $n$ mutually distrustful parties to jointly evaluate a public function $F$ over private inputs while revealing no information beyond the prescribed output [37, 38]. A degree $t-1$ polynomial is uniquely determined by its values at any $t$ distinct points, which underlines Shamir's Secret Sharing (SSS) [37, 39]. The dealer encodes the secret as the constant term of such a polynomial $f(x) \in \mathbb{F}_p[x]$ and delivers to each party $P_i$ the point $(x_i, f(x_i))$.

- **Reconstruction:** Any coalition of at least $t$ parties holds $t$ distinct points, so there is exactly one degree $t-1$ polynomial consistent with them; evaluating that polynomial at $x = 0$ reveals the secret [37].
- **Secrecy:** With fewer than $t$ points, the coalition sees an *entire affine space* of candidate polynomials that all differ in their constant term, leaving the secret information-theoretically hidden [40].

Additive secret sharing, as used in SPDZ [41], represents $s$ as

$$s \equiv \sum_{i=1}^{n} s_i \pmod{p}, \tag{3.9}$$

enabling linear operations with a single round of local communication, at the cost of requiring encrypted channels [41]. In the dealerless neighbour masking variant of additive sharing, each party chooses pairwise random masks and a prime $p$ is fixed such that $p$ exceeds the numerical range of the secrets.

Connectivity constraints limit secrecy. Lemma 1 in [42] shows that, with $n-1$ corrupt parties, perfect correctness forces maximal leakage:

$$I(s_i;\, F(\mathbf{s}), \{s_j\}_{j\in\mathcal{V}_c}) = I(s_i; s_i).$$

Proposition 1 of the same work further states that if the honest parties form disconnected components $G_{h,1}, \ldots, G_{h,k_h}$, any perfectly correct protocol must leak the evaluation of $F$ on each component. For the canonical linear function $F(\mathbf{s}) = \sum_i s_i$, this leakage reduces to the set of partial sums

$$\left\{ \sum_{i\in\mathcal{V}_{h,k}} s_i \right\}_{k=1}^{k_h}.$$

Accordingly, the adversary's ideal-world observation is

$$O_{\text{re-ideal},\mathcal{V}_c} = \{s_j\}_{j\in\mathcal{V}_c} \cup \left\{ \sum_{i\in\mathcal{V}_{h,k}} s_i \right\}_{k=1}^{k_h}, \tag{3.10}$$

so privacy within each component is quantified by

$$I\left( s_i;\, \sum_{j\in\mathcal{V}_{h,k}} s_j \right).$$

Perfect secrecy is attainable only when the honest parties form a single connected component ($k_h = 1$); otherwise, partial-sum leakage is possible.

**Information-Theoretic Privacy** For the case of $k_h = 1$, we assume all honest parties lie in one connected component. Let $S = F(\mathbf{s})$ denote the public output. For an adversary controlling $\mathcal{V}_c$, its ideal-world view is

$$O_{\text{re-ideal},\mathcal{V}_c} = \{s_j\}_{j\in\mathcal{V}_c} \cup \{S\}.$$

Because additive shares satisfy Equation 3.9, each honest share $s_i$ is uniformly distributed over $\mathbb{F}_p$ conditioned on the sum of honest shares. Consequently,

$$H\left( s_i \,\middle|\, S - \sum_{j\in\mathcal{V}_c} s_j \right) = H(s_i) \quad \Longrightarrow \quad I(s_i;\, O_{\text{re-ideal},\mathcal{V}_c}) = 0.$$

Thus, SMPC with additive secret sharing achieves perfect information-theoretic privacy whenever the honest parties remain connected.

### 3.2.3 Subspace Perturbation (SP)

The basis of subspace perturbation is a property of PDMM, which is the decomposition of the auxiliary variable $\mathbf{z}$.

$$\mathbf{z}^{(t+1)} = (1-\theta)\mathbf{z}^{(t)} + \theta\left(P\mathbf{z}^{(t)} + 2cPC\mathbf{w}^{(t)}\right), \tag{3.11}$$

where $C = \left[B_+^\top, B_-^\top\right]^\top$ and $B_+$ and $B_-$ contain the positive and negative entries of $B$, respectively. Additionally, $P$ is a permutation matrix that interchanges the upper half rows and lower half rows of the matrix it multiplies, leading to $PC = \left[B_-^\top, B_+^\top\right]^\top$. Denote $\Psi = \mathrm{ran}(C) + \mathrm{ran}(PC)$, its orthogonal complement is denoted by $\Psi^\perp = \ker(C^\top) \cap \ker((PC)^\top)$. Let $\Pi_\Psi$ represent the orthogonal projection. We can then decompose $\mathbf{z}$ into components within $\Psi$ and $\Psi^\perp$ as $\mathbf{z}^{(t)} = \mathbf{z}_\Psi^{(t)} + \mathbf{z}_{\Psi^\perp}^{(t)}$. Note that the component $\mathbf{z}_{\Psi^\perp}^{(t)}$ is not null, requiring that the number of edges should not be smaller than the number of nodes, which is not met in a star topology [18]. The main idea of the subspace perturbation technique is to exploit the auxiliary variable.

$$\mathbf{z}_{\Psi^\perp}^{(t)} = \frac{1}{2}\left(\mathbf{z}_{\Psi^\perp}^{(0)} + P\mathbf{z}_{\Psi^\perp}^{(0)}\right) + \frac{1}{2}(1-2\theta)^t\left(\mathbf{z}_{\Psi^\perp}^{(0)} - P\mathbf{z}_{\Psi^\perp}^{(0)}\right). \tag{3.12}$$

Thus, for a given graph structure and $\theta$, $\mathbf{z}_{\Psi^\perp}^{(t)}$ depends solely on the initialization of the auxiliary variable $\mathbf{z}^{(0)}$. Consequently, if $z_{i|j}^{(0)}$ is not known by the adversary, so does $z_{i|j}^{(t)}$ for subsequent iterations. This only holds for synchronous updates. For asynchronous updates, such a subspace does not exist. However, it is still proven that privacy can be achieved.

**Information-Theoretic Privacy for Subspace Perturbation** To keep $z_{i|j}^{(0)}$ unknown by adversaries, it must be transmitted in an encrypted channel for subspace perturbation based privacy to succeed. The privacy guarantee of subspace perturbation is based on the non-convergent component $\mathbf{z}_{\Psi^\perp}^{(t)}$, which is unknown due to the encrypted initialisation $\mathbf{z}^{(0)}$. To formalise this, consider the adversary's residual observation:

$$Y_i^{(t)} \in \partial f_i(\mathbf{x}_i^{(t+1)}) + \sum_{j\in\mathcal{N}_{i,h}} \mathbf{B}_{i|j}\mathbf{z}_{j|i}^{(t)},$$

where $\mathcal{N}_{i,h}$ is the set of honest neighbours of $i$, and $\mathbf{z}_{j|i}^{(t)}$ decomposes into:

$$\mathbf{z}_{j|i}^{(t)} = \underbrace{\Pi_\Psi \mathbf{z}_{j|i}^{(t)}}_{\text{Convergent}} + \underbrace{(I-\Pi_\Psi)\mathbf{z}_{j|i}^{(0)}}_{\text{Non-Convergent}}.$$

The non-convergent term $\mathbf{z}_{j|i}^{(0)}$ injects irreducible noise into $Y_i^{(t)}$, as it is statistically independent of $\partial f_i(\mathbf{x}_i^{(t+1)})$ and encrypted during initialization. The leakage of the sensitive gradient is quantified by:

$$I\left(\partial f_i(\mathbf{x}_i^{(t+1)}); Y_i^{(t)}\right) \le \frac{1}{2}\log\left(1 + \frac{\mathrm{Var}(\partial f_i)}{\mathrm{Var}\left(\sum_{j\in\mathcal{N}_{i,h}} \mathbf{B}_{i|j}(I-\Pi_\Psi)\mathbf{z}_{j|i}^{(0)}\right)}\right).$$

To enforce $I(\cdot) \leq \delta$, the variance of the non-convergent noise must satisfy:

$$\mathrm{Var}\left(\sum_{j\in\mathcal{N}_{i,h}} \mathbf{B}_{i|j}(I-\Pi_\Psi)\mathbf{z}_{j|i}^{(0)}\right) \geq \frac{\mathrm{Var}(\partial f_i)}{2^{2\delta}-1}.$$

Perfect privacy ($\delta = 0$) is achieved by choosing the initial non-convergent noise with arbitrarily large variance, the mutual information $I\big(\partial f_i(\mathbf{x}_i^{(t+1)}); Y_i^{(t)}\big)$ can be driven to zero, so the adversary gains no information about the gradient.

# Preliminary: Federated Learning Adversarial Robustness

# 4

This chapter introduces foundational concepts in Byzantine-robust aggregation for federated learning, focusing on the difference between *weak* and *strong* robust aggregators. We analyse their theoretical guarantees and how mitigation techniques interact with adversarial strategies.

## 4.1 Weak and Strong Aggregators

Robust aggregation mechanisms are generally classified into two types: weak robust aggregators and strong robust aggregators. The primary distinction lies in the dimensionality-dependent bias that an adversary can introduce.

Let $X = \{x_j \in \mathbb{R}^d \mid j \in \mathcal{N}_i\}$ represent the set of samples or updates node $i$ receives from its neighbours. Assume an adversary corrupts an $\epsilon$-fraction of these updates. We denote $|\Sigma|_2$ as the spectral norm of the covariance matrix of benign data. The bias bounds differ significantly based on aggregator type.[1]

**Weak Robust Aggregators.** Weak aggregators analyse the updates *coordinate-wise*, independently considering each dimension. Thus, an attacker can accumulate bias along each dimension. The induced bias for weak aggregators typically scales as:

$$\text{Bias}_{\text{weak}} = \widetilde{O}\big(\sqrt{\epsilon d}\big) \cdot \|\Sigma\|_2^{\frac{1}{2}},$$

growing with model dimension $d$ [43, 44].

**Strong Robust Aggregators.** Strong aggregators consider updates as full vectors, detecting anomalies across all directions simultaneously. Hence, the bias is dimension-independent:

$$\text{Bias}_{\text{strong}} = \widetilde{O}\big(\sqrt{\epsilon}\big) \cdot \|\Sigma\|_2^{\frac{1}{2}}.$$

Strong aggregators provide superior theoretical guarantees, especially in high-dimensional federated learning settings [43, 44].

## 4.2 Robustness Mechanisms

In machine and federated learning, a robustness mechanism is a training or aggregation time modification that prevents adversarially perturbed inputs while keeping honest

[1] We use Big-$O$ notation to express asymptotic upper bounds: $f(n) = O(g(n))$ means that $f(n)$ grows at most on the order of $g(n)$, up to a constant factor.

inputs' accuracy at its original level [45]. The present works realise this concept of robustness mechanisms at the communication layer, each client treats the incoming transmitted variable $\mathbf{y}_{j|i}$ as an indication to determine whether a neighbouring node $j$ of the node $i$ is an active adversary or not. Robustness, in this setting, is achieved by filtering these messages according to a local detection rule.

### 4.2.1 Trimmed Mean (Median)

One of the most well known and basic robustness mechanisms is the Trimmed mean[2]. The Trimmed mean mechanism collects a set of neighbour values. It sorts the values based on their magnitude, and discards values that have the largest distance from the measure of centrality per dimension, based on the $\alpha\%$ of the largest distances. The remaining neighbour values, then aggregated according to the aggregation algorithm. The Trimmed Mean is a rule used in many decentralised federated learning settings to obstruct adversarial attacks to prevent convergence [46, 47, 48].

**Bias Bound.** As a coordinate-wise method, the trimmed mean (or median) computes a robust aggregation separately for each dimension. This allows an adversary to inject small, undetectable biases into every coordinate, causing the total bias to accumulate across all $d$ dimensions. As a result, the worst-case bias scales as $\widetilde{O}(\sqrt{\epsilon d})\|\Sigma\|_2^{1/2}$, growing with the model's dimension [44, 43].

### 4.2.2 SignGuard

SignGuard is a defence mechanism against Byzantine adversaries in federated learning, proposed in [49]. The key idea behind SignGuard is to leverage the signs of model updates rather than their magnitudes. By focusing on the direction of the updates, SignGuard reduces the impact of any large but misleading gradients introduced by Byzantine agents. The aggregator collects the sign vectors from all participating agents and computes the element-wise majority sign $M$ as shown in Equation 4.1

$$M = \text{Majority}\left(\{\text{sign}(g_p)\}_{p=1}^{P}\right), \tag{4.1}$$

where $P$ is the total number of participating agents, $g_p$ denotes the gradient update from agent $p$, and $\text{sign}(g_p)$ represent the sign vector of $g_p$. The majority function selects the sign ($+$ or $-$) that appears most frequently for each element across all agents. The aggregated update $\tilde{g}$ is then computed by combining the majority sign with a scaling factor $\eta$:

$$\tilde{g} = \eta \cdot M, \tag{4.2}$$

where $\eta$ is a predefined learning rate or scaling parameter. By utilising majority voting on the signs, SignGuard effectively filters out anomalous updates that deviate from the consensus, enhancing robustness against adversarial attacks.

[2]If the mean is not a reliable measure of centrality, the median is often used instead.

**Bias Bound.** SignGuard aggregates only the sign of each update's coordinates using majority vote. While this mitigates the impact of large adversarial values, it still operates per dimension, so adversaries can bias the aggregated direction in every coordinate where they constitute a local majority. Consequently, the worst-case bias grows with the number of dimensions: $\widetilde{O}(\sqrt{\epsilon d})\|\Sigma\|_2^{1/2}$ [49].

### 4.2.3 Self–Centered Clipping (SCC)

Self–Centered Clipping operates in two stages:

**(i) Data–dependent clipping radius.** For every node $i$ a *single* radius $\tau_i > 0$ is computed from the observed dissimilarities between the local vector $x_i$ and its neighbours:

$$\tau_i \;=\; \sqrt{\frac{1}{\delta_i}\sum_{j\,\in\,\mathcal{N}_i} A_{ij}\,\mathbb{E}\big[\,\|\,x_i - x_j\,\|_2^2\big]}, \tag{4.3}$$

where $A_{ij}$ is the edge weight in the communication graph and $\delta_i = \sum_j A_{ij}$ the weighted degree of node $i$. Intuitively, $\tau_i$ is a robust estimate of the typical internode spread around $x_i$.

**(ii) Vector-norm clipping.** Given any update difference $z = x_j - x_i$, apply the operator

$$\mathrm{CLIP}(z,\tau_i) \;:=\; \min\Big(1,\,\frac{\tau_i}{\|z\|_2}\Big)\,z, \tag{4.4}$$

which scales $z$ down whenever it exceeds $\tau_i$. Finally node $i$ aggregates clipped increments from all neighbours:

$$\mathrm{SCCLIP}_i(x_1,\dots,x_n) \;=\; \sum_{j=1}^{n} A_{ij}\Big(x_i + \mathrm{CLIP}(x_j - x_i,\tau_i)\Big). \tag{4.5}$$

**Bias Bound.** SCC makes its aggregation decision based on the full $\ell_2$ norm of each neighbour's update relative to the local centre, clipping outliers in all directions at once. Because adversarial updates can only cause bias by exceeding the global radius, and any excess is shrunk isotropically, the total bias introduced is independent of the dimension $d$ and scales as $\widetilde{O}(\sqrt{\epsilon})\|\Sigma\|_2^{1/2}$ [**?**].

### 4.2.4 SMAD-Detection Mechanism

The Scaled Median Absolute Deviation (SMAD) method provides a robust statistical approach for detecting and isolating adversarial updates in decentralized federated learning. By measuring the deviation of each neighbour's update from the coordinate-wise median, SMAD effectively identifies outliers while maintaining resilience against a minority of corrupted agents.

**(i) Coordinatewise Median Estimation:** At each round $t$, agent $i$ receives neighbour updates $y_{j|i} \in \mathbb{R}^d$. The per dimension median

$$m_{i,k}(t) = \text{med}\{|x_{i,j,k}(t)| : j \in \mathcal{N}_i\}$$

robustly estimates the central tendency.

**(ii) Scaled Median Absolute Deviation (SMAD):** Dispersion is measured by

$$\text{SMAD}_i(t) = \tau \ \text{med}\{\| \, |x_{i,j}(t)| - m_i(t)\|_\infty : j \in \mathcal{N}_i\}$$

with scale $\tau > 0$. Neighbours with deviation above this threshold are flagged and excluded from aggregation.

**(iii) Majority-Voting Filter:** Flags are accumulated over $L$ rounds, with a majority vote used to quarantine persistently suspicious neighbours and stabilize detection.

**Bias Bound.** SMAD aggregates updates using per-coordinate statistics, making it susceptible to adversaries who spread small biases across many dimensions. As with other coordinate-wise methods, the bias accumulates proportionally to the square root of the product $\epsilon d$, yielding a total bias of order $\widetilde{O}(\sqrt{\epsilon d})\|\Sigma\|_2^{1/2}$ [43, 44].

Table 4.1: Comparison of Robust Aggregators in Federated Learning

| **Aggregator** | **Type** | **Bias Bound** | **Limitation** |
|---|---|---|---|
| Trimmed Mean / Median | Weak | $\widetilde{O}\big(\sqrt{\epsilon d}\big)$ | Dimension–dependent bias, coordinate–wise only |
| SignGuard | Weak | $\widetilde{O}\big(\sqrt{\epsilon d}\big)$ | Majority-vote needs $> 50\%$ benign, ignores magnitude, can slow convergence |
| Self–Centered Clipping (SCC) | Strong | $\widetilde{O}\big(\sqrt{\epsilon}\big)$ | Moderate computational cost, requires $\ell_2$ norm computation |
| SMAD | Weak | $\widetilde{O}\big(\sqrt{\epsilon d}\big)$ | Vulnerable to sparse, subtle attacks |

## 4.3 Adversarial Attacks in Federated Learning

Attackers may aim to prevent the convergence of federated learning algorithms instead of introducing backdoors, thereby disrupting the collaborative training process and degrading the performance of the global model. By injecting malicious updates, adversaries can cause the model to fail to reach an optimal solution or to converge to a suboptimal one. Several types of attacks can be employed to prevent convergence, including Gaussian attacks, sample-duplicating attacks, A-Little-Is-Enough (ALIE) attacks, and sign-flipping attacks.

### 4.3.1 Gaussian Noise Attack

In distributed and federated learning, adversaries often disrupt convergence by replacing their true updates with carefully crafted noise. A classic example is the Gaussian attack: a corrupt client sends gradients or parameters whose entries are drawn from a zero mean Gaussian distribution with an arbitrarily large variance, $\mathcal{N}(0, \sigma_i^2)$. The resulting high-variance fluctuations overwhelm the aggregation step, destabilising the optimisation process and potentially preventing convergence altogether [19]. Different adversaries can even choose different variances, $\mathcal{N}(0, x), \mathcal{N}(0, y), \mathcal{N}(0, z)$ to make detection harder.

**Additive Gaussian Noise Attack** A more concealed strategy than message replacement is to perturb otherwise genuine updates with zero mean Gaussian noise. Each malicious client sends

$$\theta_i' \;=\; \theta_i \;+\; \epsilon_i, \qquad \epsilon_i \sim \mathcal{N}\big(0, \sigma^2\big) \tag{4.6}$$

Because the noise is centred at zero on legitimate gradients, defences that look for outliers cab fail to flag the attack, making additive Gaussian noise a subtle yet potent threat in distributed and federated learning.

### 4.3.2 Copycat Attack

A coalition of Byzantine clients can undermine learning by cloning the update of one benign participant and submitting it en masse. Concretely, each malicious client replaces its own parameters with those of a chosen target client,

$$\theta_i' \;\leftarrow\; \theta_{\text{target}} \tag{4.7}$$

thereby flooding the aggregator with identical copies of the same message [50]. The over-representation of this single update biases the aggregation step toward the target's information, the global model learns disproportionately from that client's data, reducing diversity in the updates, and increasing the risk of overfitting [51]. Because the copied message itself is benign, simple outlier detection or variance-based defences often fail to spot the attack, it is the multiplicity of the duplicates that skews the optimisation process, not the magnitude.

### 4.3.3 Little-Is-Enough (LIE) Attack

The Little-Is-Enough (LIE) attack targets federated learning networks, aiming to subtly shift the aggregated result in the attacker's desired direction without detection by defence mechanisms. Instead of submitting values that deviate significantly from the mean, attackers craft malicious updates close to the distribution boundary of benign updates to remain undetected:

$$\theta_i' \leftarrow \mu_{\text{benign}} + z\sigma_{\text{benign}}, \quad 0 < z \ll 1, \tag{4.8}$$

where $\mu_{\text{benign}}$ and $\sigma_{\text{benign}}$ denote the empirical mean and standard deviation of benign updates, respectively. A common choice is $z \approx 0.08$, strategically positioning the malicious update just inside typical variance thresholds used by robust aggregators like Krum, Trimmed Mean, or Median [1].

The attack assumes benign clients' parameter values follow a normal distribution around the true mean. To shift the global model, the attacker must control a majority of nodes. When Byzantine nodes are in the minority ($m < \lfloor n/2 + 1 \rfloor$), they seduce the defence mechanism to classify benign clients as Byzantine [1]. As shown in Figure 4.1, attackers create values between the true mean and the supporters, causing the mean to shift toward Byzantine workers. This makes defences remove opposing benign nodes as perceived outliers.

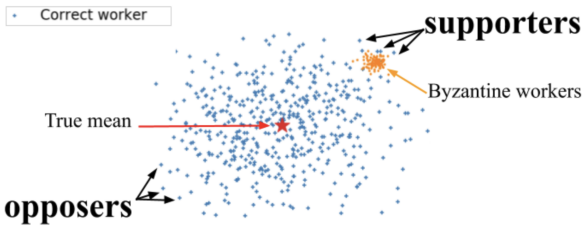


Figure 4.1: Byzantine workers seducing benign nodes to shift the mean [1].

### 4.3.4 Sign-Flipping Attack

Malicious clients reverse the update direction and optionally scale or perturb it before sending it to the aggregator:

$$\theta_i' \;=\; -s\,\theta_i + \epsilon_i, \qquad s > 0,\ \epsilon_i. \tag{4.9}$$

By inverting the sign (with scale $s$) of either their local gradient or a mean computed from benign updates, the attackers push the global optimiser in the opposite direction; added noise can further mask the manipulation [52, 53].

### 4.3.5 Label-Flipping (Dirty-Label) Attack

A data-poisoning client corrupts learning by altering the target labels of its local examples while keeping the inputs unchanged. In a $C$-class problem the attacker inverts each label,

$$y_i' \;=\; C - 1 - y_i, \tag{4.10}$$

so training proceeds on $(x_i, y_i')$ and the global objective becomes

$$\mathcal{L}(\theta) \;=\; \frac{1}{N}\sum_{i} \ell\big(f_\theta(x_i), y_i'\big), \tag{4.11}$$

driving the model to misclassify one or more classes [54, 55].

| **Attack Type** | **Impact on Convergence** | **Detectability** | **References** |
|---|---|---|---|
| Gaussian Noise | Severe disruption | Variance-based detection | [44, 56] |
| Copycat | Biased convergence | Requires redundancy check | [50, 51] |
| LIE | Subtle convergence disruption | Avoids robust methods | [1] |
| Sign-Flipping | Severe disruption | High Norm-based detection | [52, 56] |
| Label-Flipping | Misclassification | Local validation required | [54, 55] |

Table 4.2: Comparison of adversarial attacks in federated learning.

## 4.4 Trade-Off Formulation

To characterise the trade-off between robustness and privacy, we make the following assumptions:

1. The additive noise $Y$ is independent of both the private signal $X$ and the honesty $D$.
2. The detector only observes the noisy signal $Z = X + Y$, never directly accessing $X$.

We define $D \in \{0, 1\}$ as the honesty, with $D = 0$ for an honest node and $D = 1$ for an adversarial node. Following [18], let $X$ and $Y$ be independent random variables, where $X$ represents the private gradient with variance $\sigma_X^2$, and $Y$ is Gaussian noise with variance $\sigma_Y^2$. Given a privacy budget $\delta > 0$, the information-theoretic privacy requirement is formulated as:

$$I(X; Z) = H(Z) - H(Z \mid X) \leq \delta. \tag{4.12}$$

From this privacy constraint, the minimum noise variance required can be explicitly bounded by:

$$\sigma_Y^2 \geq \frac{\sigma_X^2}{2^{2\delta} - 1}. \tag{4.13}$$

When a neighbour is honest ($D = 0$), its gradient follows the legitimate distribution $P_{X|D=0}$. In contrast, adversarial neighbours ($D = 1$) produce gradients from an alternative distribution $P_{X|D=1}$. Observing only the noisy signal $Z$, the conditional independence assumption implies:

$$Z \perp\!\!\!\perp D \mid X \quad \Rightarrow \quad I(D; Z \mid X) = 0. \tag{4.14}$$

We now expand the mutual information $I(X; Z)$ using standard identities:

$$\begin{aligned} I(X; Z) &= H(Z) - H(Z \mid X) \\ &= [H(Z \mid D) + I(D; Z)] - [H(Z \mid X, D) + I(D; Z \mid X)]. \end{aligned} \tag{4.15}$$

Considering conditional independence ($I(D; Z \mid X) = 0$) and $H(Z \mid X, D) = H(Y)$ (since $Y$ is independent), we simplify Equation 4.15 to:

$$I(X; Z) = I(X; Z \mid D) + I(D; Z). \tag{4.16}$$

This decomposition clarifies how information about $X$ can be partitioned into components conditional on the honesty $D$. Furthermore, we have:

$$I(D; Z) \leq I(X; Z) \leq \delta. \tag{4.17}$$

Since $I(D; Z) = H(D) - H(D \mid Z)$ directly measures how much information $Z$ reveals about node honesty, Equation 4.17 yields a fundamental limit for detection under a given privacy budget $\delta$:

$$I(D; Z) \leq \delta. \tag{4.18}$$

This inequality indicates no detector can infer more than $\delta$ bits of information about honesty from the noisy observation $Z$.

## 4.5 Binary Detection and Fano's Inequality

To quantify detection accuracy, we use Fano's inequality for binary classification. For any estimator $\hat{D}$ attempting to infer the discrete random variable $D$ from observation $Z$, Fano's inequality states:

$$H(D \mid Z) \leq H_b(P_e) + P_e \log(|D| - 1), \tag{4.19}$$

where $P_e = P(\hat{D} \neq D)$ is the error probability. With $|D| = 2$, the term $\log(1) = 0$ vanishes, simplifying the inequality to:

$$H(D \mid Z) \leq H_b(P_e). \tag{4.20}$$

Substituting $H(D \mid Z) = H(D) - I(D; Z)$ gives:

$$H(D) - I(D; Z) \leq H_b(P_e). \tag{4.21}$$

Assuming a worst-case uniform prior $P(D = 0) = P(D = 1) = 1/2$, the entropy is maximal:

$$H(D) = H_b\left(\frac{1}{2}\right) = 1 \text{ bit}. \tag{4.22}$$

Thus, we have:

$$1 - I(D; Z) \leq H_b(P_e). \tag{4.23}$$

Because $H_b(\cdot)$ is strictly increasing on $[0, 1/2]$, we obtain:

$$P_e^\star \geq H_b^{-1}(1 - I(D; Z)) \geq H_b^{-1}(1 - \delta). \tag{4.24}$$

**Interpretation:**

- **Strong Privacy** ($\delta \to 0$): Detection error approaches random guessing ($P_e^\star \to 1/2$).
- **Weak Privacy** ($\delta \to 1$): Detection error can vanish ($P_e^\star \to 0$), allowing potentially perfect detection.

## 4.6 Influence of Adversarial Variance

Consider an adversary manipulating its gradient variance, setting:

$$\sigma_{X,a}^2 = \kappa\, \sigma_X^2, \quad \kappa > 1. \tag{4.25}$$

The honest nodes select noise $\sigma_Y^2$ to satisfy (4.13). Thus, the effective leakage against adversarial variance becomes:

$$\delta_a = \frac{1}{2} \log_2 \left(1 + \frac{\sigma_{X,a}^2}{\sigma_Y^2}\right) = \frac{1}{2} \log_2 \left(1 + \kappa \frac{\sigma_X^2}{\sigma_Y^2}\right) > \delta. \tag{4.26}$$

Since $\delta_a > \delta$, the detector effectively has more information about the adversary, improving detection capability. Applying Fano's inequality with this increased leakage, we have a tighter detection error bound:

$$P_{e,a}^\star \geq H_b^{-1}(1 - \delta_a) < H_b^{-1}(1 - \delta). \tag{4.27}$$

Thus, increasing variance may inadvertently help detection by reducing the adversary's stealth.

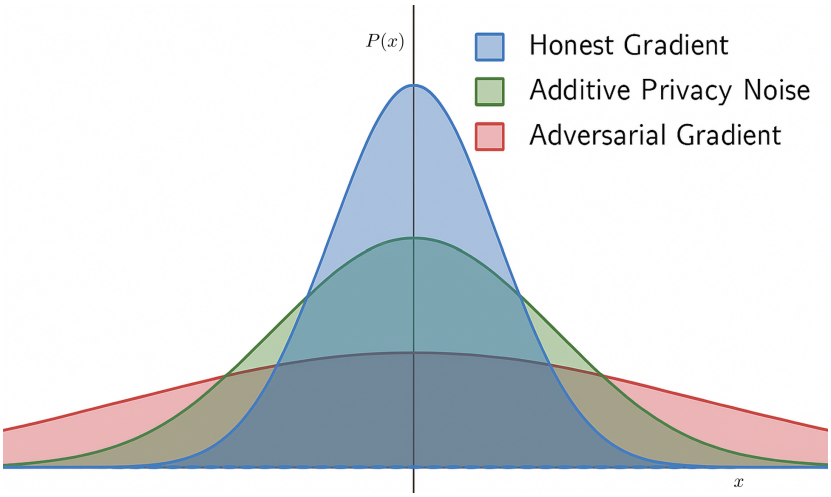


Figure 4.2: Example illustration of distributions under variance attack: Honest ($D = 0$), Noise ($Y$), and Adversarial ($D = 1$) gradients. Increased adversarial variance $\sigma^2_{X,a}$ leads to enhanced detectability.

## 4.7 Summary

The analysis illustrates the trade-off between privacy and robustness. Tightening privacy (smaller $\delta$) restricts the effectiveness of adversarial detection. Conversely, adversarial manipulation aimed at privacy violation can enhance detectability, emphasising careful design of combined robustness-privacy mechanisms in federated learning.

# 5 Simulation Set-Up

## 5.1 Network Set-Up

Figure 5.1 illustrates our decentralized peer-to-peer network, $G$. This network is modelled as a two-dimensional Random Geometric Graph (RGG) comprising $N = 20$ nodes [57].

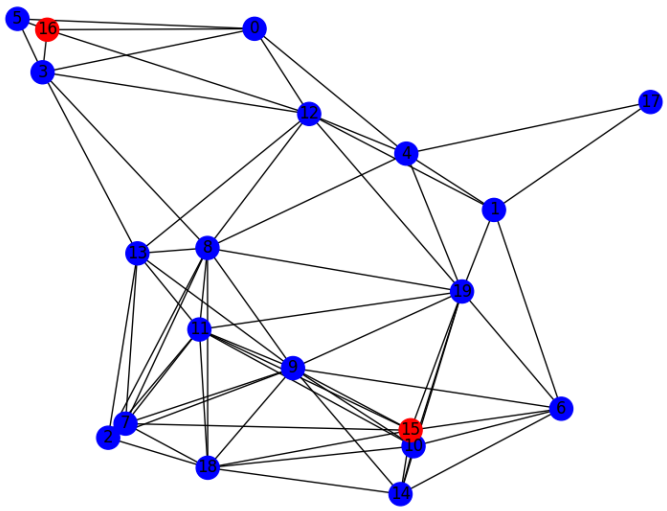


Figure 5.1: Graph representation illustrating honest and adversarial nodes.

Each node is positioned independently and uniformly at random within a unit square. An undirected edge (communication link) connects two nodes if their Euclidean distance is less than or equal to a radius. It is well-established that if the connection radius satisfies:

$$r \geq \sqrt{\frac{\log N}{N}}, \tag{5.1}$$

Then the network $G$ is connected with a high probability. Specifically, for large $N$, the probability of connectivity is at least $(1 - \frac{1}{N^2})$ [58]. Thus, we have:

$$\Pr\{G \text{ is connected}\} \geq 1 - \frac{1}{N^2}$$

Furthermore, we simulate a scenario where 2 out of the $N$ nodes behave maliciously. These corrupted nodes, the active adversarial nodes, are highlighted in red in Figure 5.1.

The remaining nodes, highlighted in blue, are honest and follow the prescribed protocol accurately. Malicious nodes, however, may deviate arbitrarily from the protocol.We categorise nodes into three types:

- **Honest nodes** ($V_h$): Follow the averaging protocol precisely and do not attempt to infer private data.
- **Passive adversarial nodes** ($V_{c,p}$): Follow the protocol correctly but collaborate among themselves to infer as much private data as possible from honest nodes.
- **Active adversarial nodes** ($V_{c,a}$): Inject malicious updates intentionally to disrupt the consensus process, aiming either to cause divergence or to steer the network toward an adversarially chosen state.

We denote the complete set of adversarial nodes as $V_c = V_{c,p} \cup V_{c,a}$, ensuring:

$$V = V_h \cup V_c, \quad V_h \cap V_c = \varnothing, \quad V_c = V_{c,p} \cup V_{c,a}, \quad V_{c,p} \cap V_{c,a} = \varnothing.$$

We additionally impose two structural assumptions on the network:

**Assumption 1.** ***Limited Adversarial Neighbours****. For every honest node $i$, the number of active adversarial neighbours is strictly less than half of its total degree:*

$$\frac{d_i}{2} > |\{j \in \mathcal{N}_i \cap V_{c,a} \mid (i,j) \in \mathcal{E}\}|.$$

**Assumption 2.** ***Residual Connectivity****. After removing all active adversarial nodes $V_{c,a}$, the remaining subgraph formed by honest and passive-adversarial nodes remains connected:*

$$G[V \setminus V_{c,a}] \quad \textit{is connected.}$$

This residual connectivity assumption is not only crucial for the feasibility of SMPC, but also for the reliable operation of distributed detection algorithms. Suppose the network becomes fragmented due to adversarial removals or overly aggressive detection responses. In that case, honest nodes may become isolated, which in turn compromises both the collaborative learning process and the effectiveness of adversarial detection. Therefore, maintaining a connected, honest subgraph is a key prerequisite for both privacy and robustness in federated learning.

## 5.2 Node Set-Up

We study the trade-off between privacy and adversarial-behaviour detection with the fully decentralised federated-learning FedAVG scheme. Each of the $N$ clients holds a disjoint shard of the training dataset (MNIST) and is connected to its neighbours by a fixed connected peer-to-peer graph. All clients train the same two-layer perceptron, MLP($784 \to 128 \to 10$) with ReLU activation and optimise the cross-entropy loss over their local data. The information exchange is done with the synchronous PDMM algorithm. In the synchronous PDMM–FedAvg setup, each client performs 5 local epochs of mini-batch SGD (batch size $B = 64$, learning rate $\eta = 0.1$) on its full dataset per round, before starting the PDMM averaging process. This procedure is repeated for 50 global rounds.

**Privacy mechanisms evaluated:**

a) Differential privacy via Gaussian noise,

b) Sub-space perturbation,

c) SMPC with pairwise additive masking.

**Adversarial attacks injected:**

a) Pure Gaussian model poisoning,

b) Gaussian additive noise attack,

c) ALIE outlier attack,

d) Label-flipping attack,

e) Sign-flipping attack.

Detection is performed with the scaled median absolute deviation (scaled-MAD) rule presented in Section 4.2.4, applied to PDMM messages to flag and isolate malicious neighbours.

## 5.3 Attack setup

To rigorously evaluate the robustness of the detection and aggregation schemes, we simulate several canonical adversarial attacks drawn from the federated learning literature. For all experiments, we fix the number of active adversarial nodes to two out of $N = 20$ total clients (i.e., 10% Byzantine proportion), in line with standard benchmarks [59, 60, 61]. Each attack is instantiated with carefully selected parameter values to balance detectability and impact, as recommended by recent studies [61, 62, 63].

- **Random Model Poisoning:** Corrupted clients replace their model updates with vectors sampled from a Gaussian distribution of zero mean and variance $\sigma^2 = 100$. This level of variance produces adversarial updates that are disruptive but not trivially detectable, as recommended in [60, 62].

- **Additive Gaussian Noise Attack:** Malicious clients add Gaussian noise with zero mean and variance $\sigma^2 = 10$ to their model updates. This models a more subtle attack, designed to avoid simple anomaly detection by introducing a stochastic perturbation [62, 60].
- **ALIE Attack:** Each adversarial update is crafted as $\mu_{\text{honest}} + z \cdot \sigma_{\text{honest}}$, where $\mu_{\text{honest}}$ and $\sigma_{\text{honest}}$ are the mean and standard deviation of the honest clients' updates, and $z = 0.08$. This is the recommended value for strong but not easily flagged outliers, following the "A Little Is Enough (ALIE)" paradigm from [61].
- **Sign-Flipping Attack:** Adversarial clients submit the negative of their computed model update (i.e., all parameters multiplied by $-1$). This classic Byzantine strategy seeks to directly counteract the progress of honest nodes [59, 60].
- **Label-Flipping Attack:** In this data poisoning attack, adversarial clients systematically permute the labels of all their training samples according to a fixed mapping:

  $$0 \mapsto 3, \quad 1 \mapsto 4, \quad 2 \mapsto 7, \quad 3 \mapsto 5, \quad 4 \mapsto 8, \quad 5 \mapsto 0, \quad 6 \mapsto 9, \quad 7 \mapsto 6, \quad 8 \mapsto 2, \quad 9 \mapsto 1.$$

  This constant permutation corrupts the ground truth during local training, propagating misleading gradients to the global model and degrading its accuracy [64, 63]. The mapping is fixed throughout the experiment to ensure reproducibility.

The parameter choices (see Table 5.1) are motivated by literature benchmarks and preliminary tuning to ensure that attacks are neither trivially detectable nor completely ineffective. Each attack is run with two adversarial clients per experiment.

Table 5.1: Attack types and parameter values used in experiments.

| **Attack** | **Parameter(s)** | **Value(s)** |
|---|---|---|
| Random Model Poisoning | Variance ($\sigma^2$) | 100 |
| Additive Gaussian Noise | Variance ($\sigma^2$) | 100 |
| ALIE (outlier) | $z$ | 0.08 |
| Sign-Flipping | Scaling factor | $-1$ |
| Label-Flipping | Labels | Permuted |

All attack implementations follow the protocols established in [59, 60, 61, 64, 63], ensuring a strong, diverse threat model for empirical evaluation.

## 5.4 Evaluation Metrics

### 5.4.1 Training Accuracy and Loss

During training, the training loss and accuracy are computed and recorded at every communication round. After each round in which all clients have locally updated their models for a fixed number of epochs, the average training loss and accuracy across all clients are calculated.

### 5.4.2 Test Accuracy and Loss

Due to computational constraints, the evaluation frequency on the test dataset is not uniform throughout training. Instead, we adopt an adaptive evaluation schedule: the model is tested more frequently in the early stages, every 5th communication round for the first 50 rounds, to capture the rapid improvements typically observed during the early phase. As training progresses and model performance stabilises, the evaluation interval increases to every 10 rounds between rounds the first 50 and 200, and every 20 rounds after the 200th round.

### 5.4.3 FAR and MDR

We evaluate the effectiveness of the defence mechanisms using two key metrics: False Alarm Rate (FAR) and MisDetection Rate (MDR), defined as follows:

**False Alarm Rate** (FAR) quantifies the tendency of the detection mechanism to incorrectly classify honest nodes as malicious. For a time frame of $L$ samples, let $D(i,j)$ denote the number of times node $i$ flagged node $j$ as malicious. The FAR at round $kL$ is defined by:

$$\mathrm{FAR}(k) = \frac{1}{|E_H|} \sum_{(i,j)\in E_H} \mathbb{I}\left(D(i,j)(kL) > \frac{L}{2}\right),$$

where $E_H \subseteq (V \setminus V_{c,a}) \times (V \setminus V_{c,a})$ denotes the set of edges between honest nodes, and $\mathbb{I}(\cdot)$ is the indicator function.

**MisDetection Rate** (MDR) captures the failure to identify truly adversarial nodes. Specifically, it measures how often honest nodes fail to detect adversaries. MDR at round $kL$ is computed as:

$$\mathrm{MDR}(k) = \frac{1}{|E_A|} \sum_{(i,j)\in E_A} \mathbb{I}\left(D(i,j)(kL) < \frac{L}{2}\right),$$

where $E_A \subseteq (V \setminus V_{c,a}) \times V_{c,a}$ represents the set of directed edges from honest to adversarial nodes.

To visualise MDR and FAR without producing separate plots for each communication round, we compute the average metric value across all PDMM iterations within each communication round. This results in a single representative MDR and FAR value per round.

## 5.5 Results

### 5.5.1 No Attack

Figure 5.2 shows heatmaps of the False Alarm Rate (FAR) and MisDetection Rate (MDR) for differential privacy, SMPC, and subspace perturbation when no attack is present. For differential privacy and subspace perturbation, FAR is low at small noise levels but increases rapidly with higher privacy parameters, indicating more frequent false alarms. SMPC shows a consistently high FAR across all settings. In all cases, MDR remains low as there are no adversaries. This demonstrates that increasing privacy can reduce the reliability of adversarial detection, even when the network is attack free.

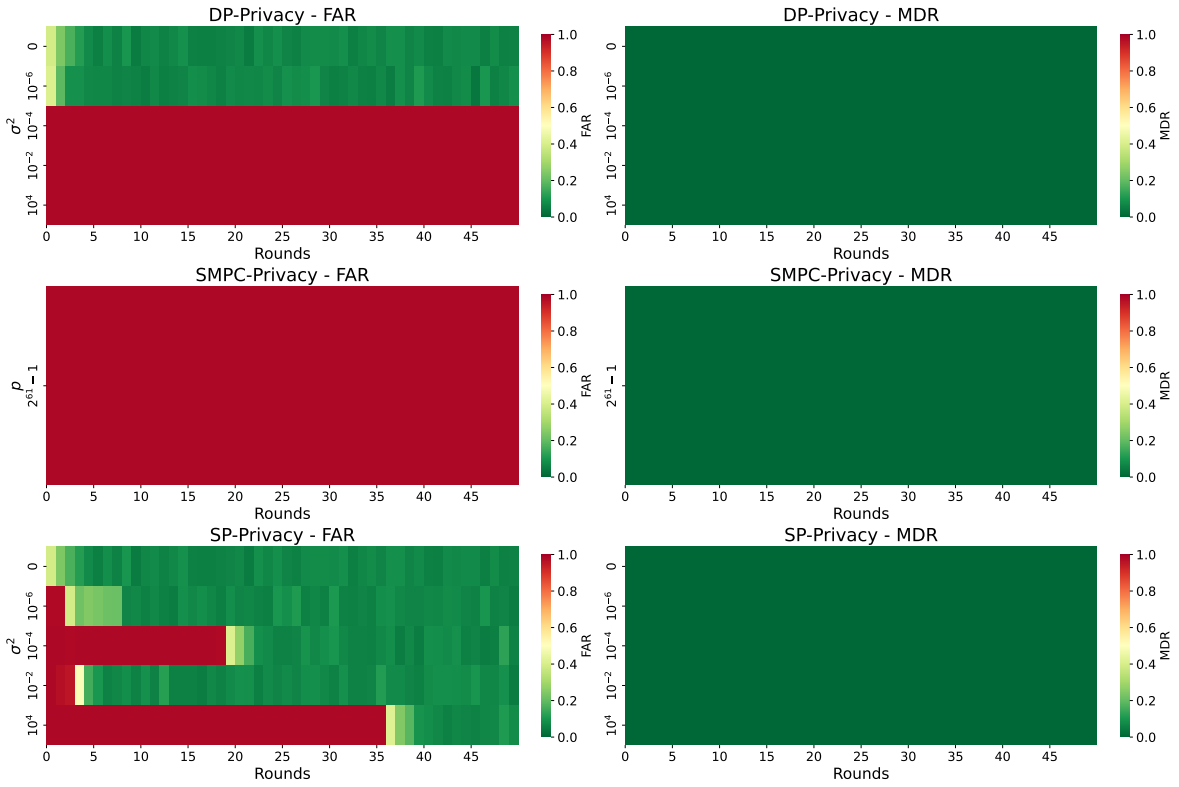


Figure 5.2: MDR and FAR for all privacy methods with no attack.

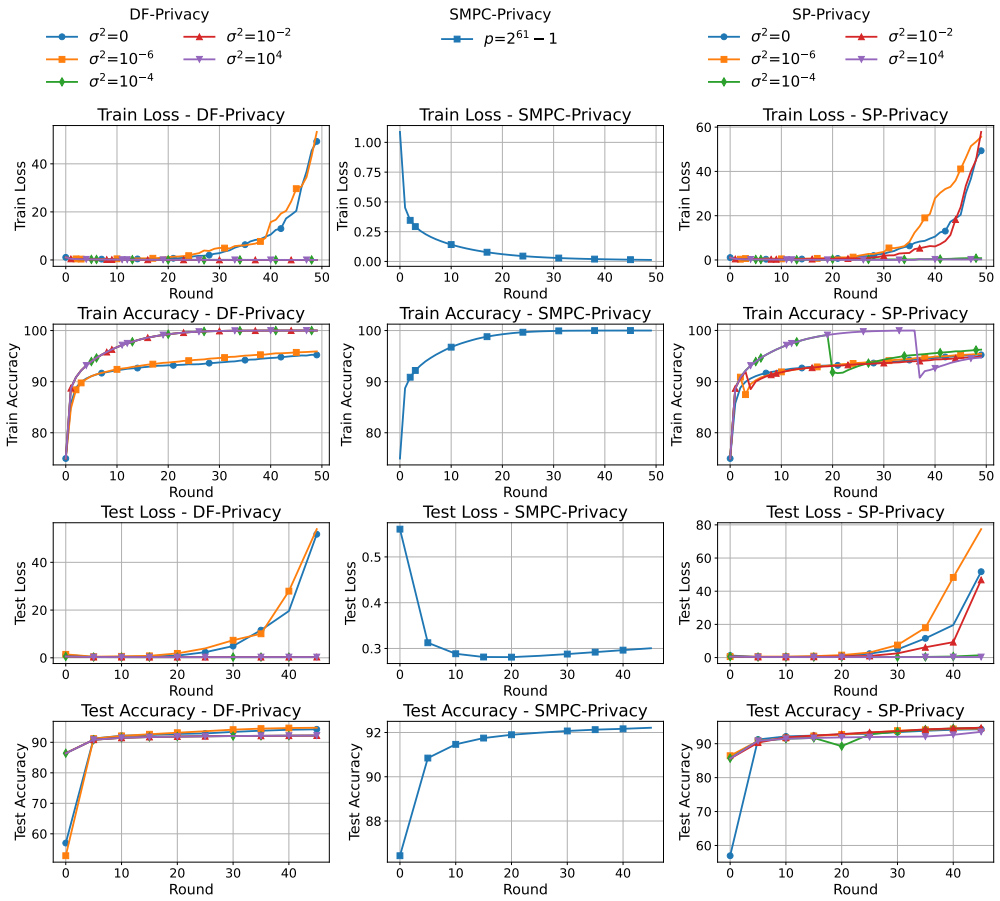


Figure 5.3: Testing and training losses and accuracies for all privacy methods with no attack.

| Privacy | Param | train_loss | train_accuracy | test_loss | test_accuracy |
|---|---|---|---|---|---|
| DF-Privacy | 0 | 49.335912 | 95.223000 | 51.774869 | 94.266000 |
| DF-Privacy | $10^{-6}$ | 53.270019 | 95.917667 | 53.909140 | 94.776000 |
| DF-Privacy | $10^{-4}$ | 0.012820 | 100.000000 | 0.300516 | 92.205000 |
| DF-Privacy | $10^{-2}$ | 0.012818 | 100.000000 | 0.300480 | 92.214000 |
| DF-Privacy | $10^{4}$ | 0.012819 | 100.000000 | 0.300492 | 92.214000 |
| SMPC-Privacy | $2^{61}-1$ | 0.012818 | 100.000000 | 0.300480 | 92.214000 |
| SP-Privacy | 0 | 49.335912 | 95.223000 | 51.774869 | 94.266000 |
| SP-Privacy | $10^{-6}$ | 55.608769 | 95.437333 | 77.418474 | 94.476000 |
| SP-Privacy | $10^{-4}$ | 0.803917 | 96.243000 | 1.397796 | 94.701500 |
| SP-Privacy | $10^{-2}$ | 57.829862 | 94.920000 | 46.817626 | 94.696500 |
| SP-Privacy | $10^{4}$ | 0.283121 | 94.704333 | 0.313000 | 93.494500 |

Table 5.2: Final performance of each privacy mechanism under no attack.

### 5.5.2 Pure Gaussian model poisoning

Figure 5.4 shows heatmaps of the False Alarm Rate (FAR) and MisDetection Rate (MDR) for differential privacy, SMPC, and subspace perturbation during a pure Gaussian model poisoning attack. For differential privacy and subspace perturbation, FAR increases with higher privacy parameters, while MDR also rises in later rounds and with stronger privacy, meaning more adversarial nodes are missed. SMPC starts with high FAR and MDR at lower privacy settings, but both rates decrease as privacy increases. These results show that increasing privacy makes it more difficult to reliably detect adversarial nodes when the network is under attack.

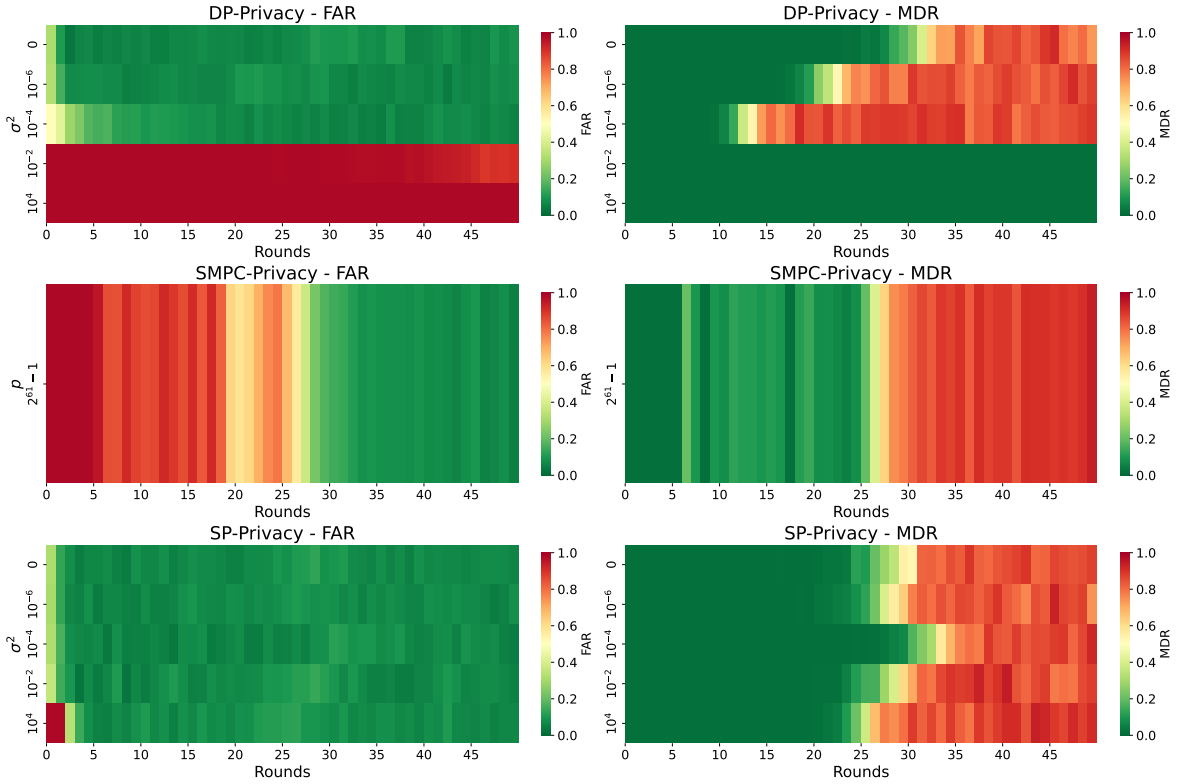


Figure 5.4: MDR and FAR for all privacy methods with the pure Gaussian model poisoning attack.

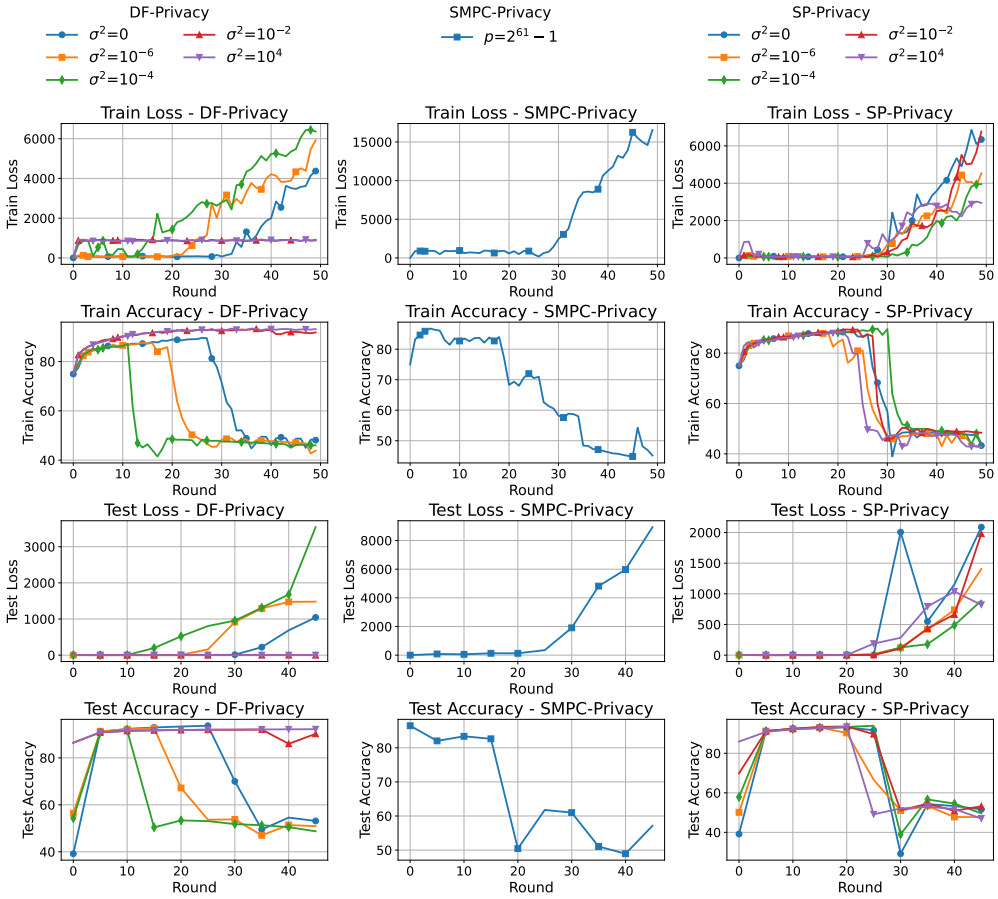


Figure 5.5: Testing and training losses and accuracies for all privacy methods with the pure Gaussian model poisoning attack.

| Privacy | Param | train_loss | train_accuracy | test_loss | test_accuracy |
|---|---|---|---|---|---|
| DF-Privacy | 0 | 4372.496814 | 48.121000 | 1043.888931 | 53.142222 |
| DF-Privacy | $10^{-6}$ | 5921.509142 | 43.869667 | 1480.388771 | 50.918333 |
| DF-Privacy | $10^{-4}$ | 6361.811268 | 45.989333 | 3541.667434 | 48.726111 |
| DF-Privacy | $10^{-2}$ | 908.848027 | 91.831667 | 0.513369 | 90.241667 |
| DF-Privacy | $10^{4}$ | 872.436032 | 93.171000 | 0.299554 | 92.223333 |
| SMPC-Privacy | $2^{61}-1$ | 16556.168516 | 45.149667 | 8934.592069 | 57.126667 |
| SP-Privacy | 0 | 6340.501866 | 43.287333 | 2086.971098 | 51.632222 |
| SP-Privacy | $10^{-6}$ | 4530.671095 | 42.803000 | 1403.488760 | 47.718333 |
| SP-Privacy | $10^{-4}$ | 3951.323499 | 44.126667 | 889.187631 | 49.618333 |
| SP-Privacy | $10^{-2}$ | 6765.672985 | 48.389333 | 1979.638480 | 53.195556 |
| SP-Privacy | $10^{4}$ | 2946.687319 | 42.796667 | 820.932079 | 47.001111 |

Table 5.3: Final performance of each privacy mechanism under the pure Gaussian model poisoning attack.

### 5.5.3 Gaussian additive Noise

Figure 5.6 shows heatmaps of the False Alarm Rate (FAR) and MisDetection Rate (MDR) for differential privacy, SMPC, and subspace perturbation under the additive Gaussian model poisoning attack. For differential privacy and subspace perturbation, FAR remains low at small noise levels but increases with higher privacy, while MDR becomes significant only in later rounds and at high privacy settings, indicating that adversarial nodes are increasingly missed. SMPC starts with high FAR and low MDR, but as privacy increases, FAR drops and MDR rises in the later rounds. These results show that increasing privacy again reduces the ability to reliably detect adversarial nodes, especially as the attack persists.

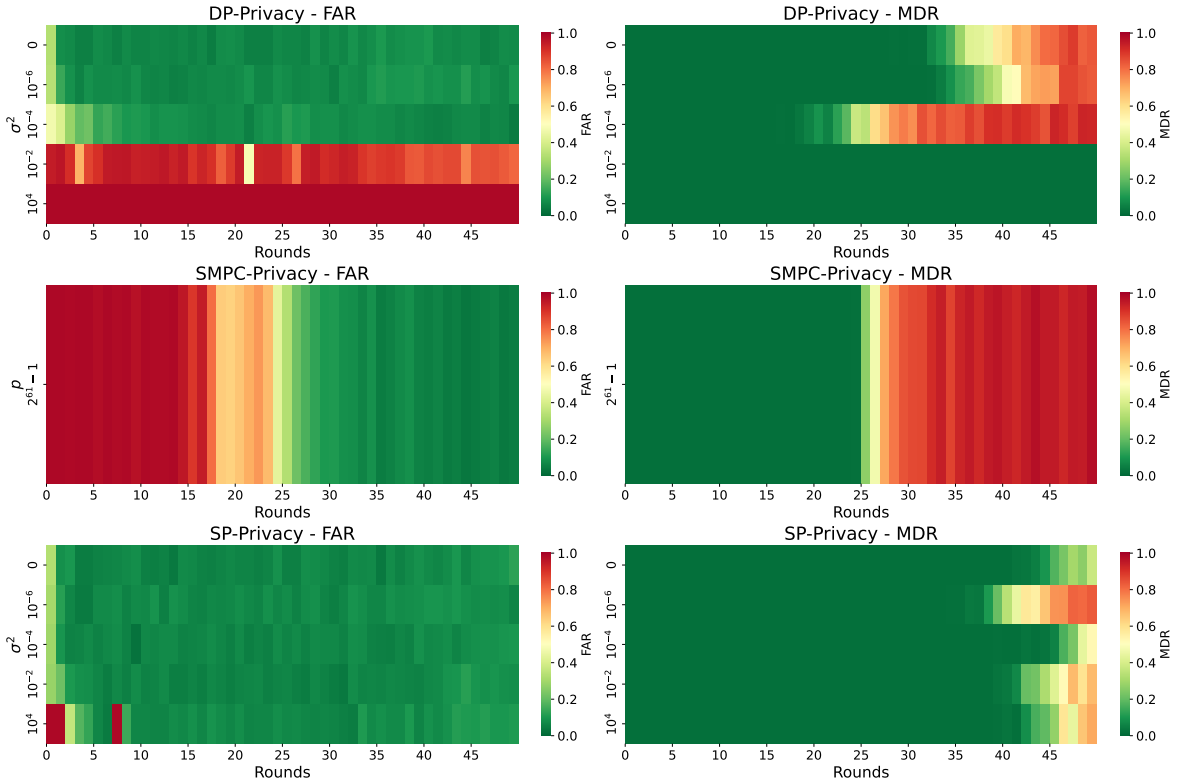


Figure 5.6: MDR and FAR for all privacy methods with the additive Gaussian model poisoning attack.

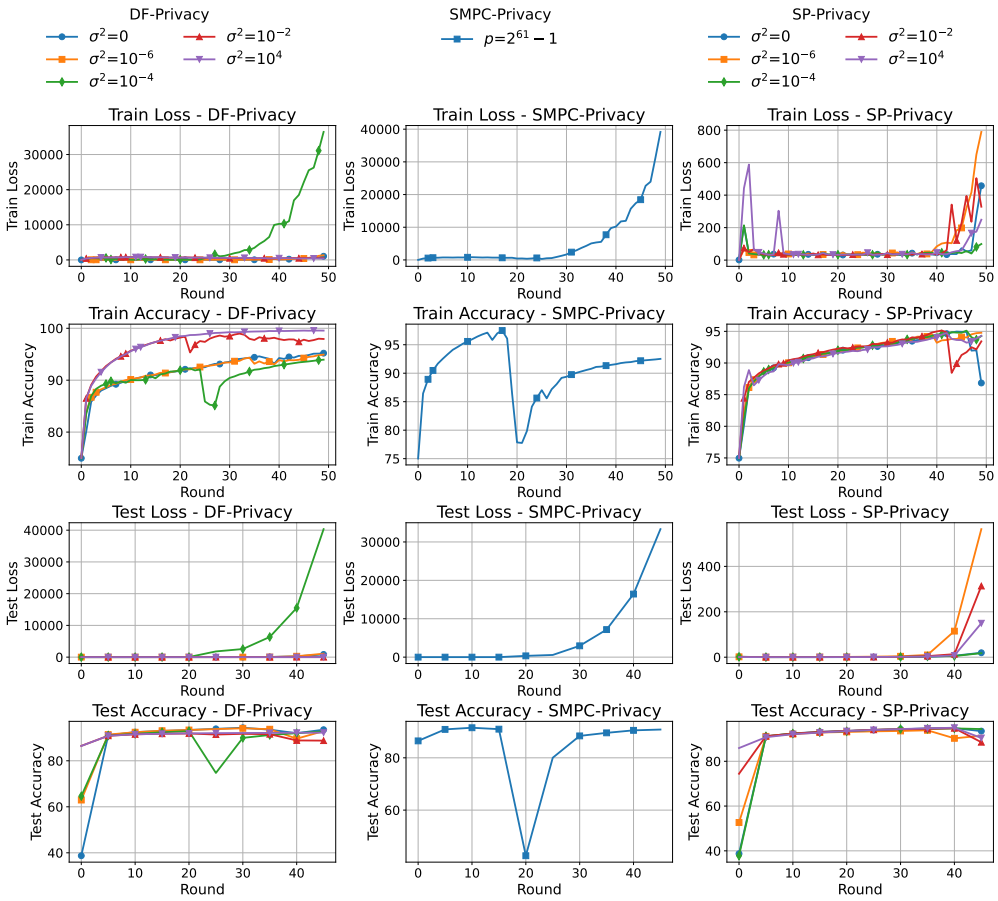


Figure 5.7: Testing and training losses and accuracies for all privacy methods with the additive Gaussian model poisoning attack.

| Privacy | Param | train_loss | train_accuracy | test_loss | test_accuracy |
|---|---|---|---|---|---|
| DF-Privacy | 0 | 1043.707227 | 95.204667 | 852.059209 | 93.512778 |
| DF-Privacy | $10^{-6}$ | 1316.406611 | 94.922333 | 1090.068989 | 92.906667 |
| DF-Privacy | $10^{-4}$ | 36427.493763 | 93.949333 | 40327.301017 | 92.775556 |
| DF-Privacy | $10^{-2}$ | 509.986998 | 97.938000 | 0.609012 | 88.765000 |
| DF-Privacy | $10^{4}$ | 398.929011 | 99.551000 | 0.299554 | 92.223333 |
| SMPC-Privacy | $2^{61}-1$ | 39155.961656 | 92.516333 | 33335.904828 | 90.812222 |
| SP-Privacy | 0 | 457.685248 | 86.841333 | 19.573742 | 93.481111 |
| SP-Privacy | $10^{-6}$ | 789.077356 | 94.776667 | 564.063987 | 91.417778 |
| SP-Privacy | $10^{-4}$ | 99.024798 | 94.273333 | 17.764089 | 94.397222 |
| SP-Privacy | $10^{-2}$ | 328.796953 | 93.418333 | 313.522617 | 88.518333 |
| SP-Privacy | $10^{4}$ | 248.582525 | 94.279333 | 149.563033 | 90.417222 |

Table 5.4: Final performance of each privacy mechanism under the additive Gaussian poisoning attack.

### 5.5.4 ALIE

Figure 5.8 shows heatmaps of the False Alarm Rate (FAR) and MisDetection Rate (MDR) for differential privacy, SMPC, and subspace perturbation under the ALIE attack. For differential privacy and SMPC, FAR is high across most privacy settings, while MDR is also high for differential privacy at low noise levels, meaning many adversarial nodes are not detected. SMPC shows high FAR but maintains a low MDR. For subspace perturbation, FAR stays low at small noise but rises with increased privacy, while MDR is consistently high except at the lowest privacy settings. These results show that the ALIE attack makes adversarial detect harder, especially when privacy is strong.

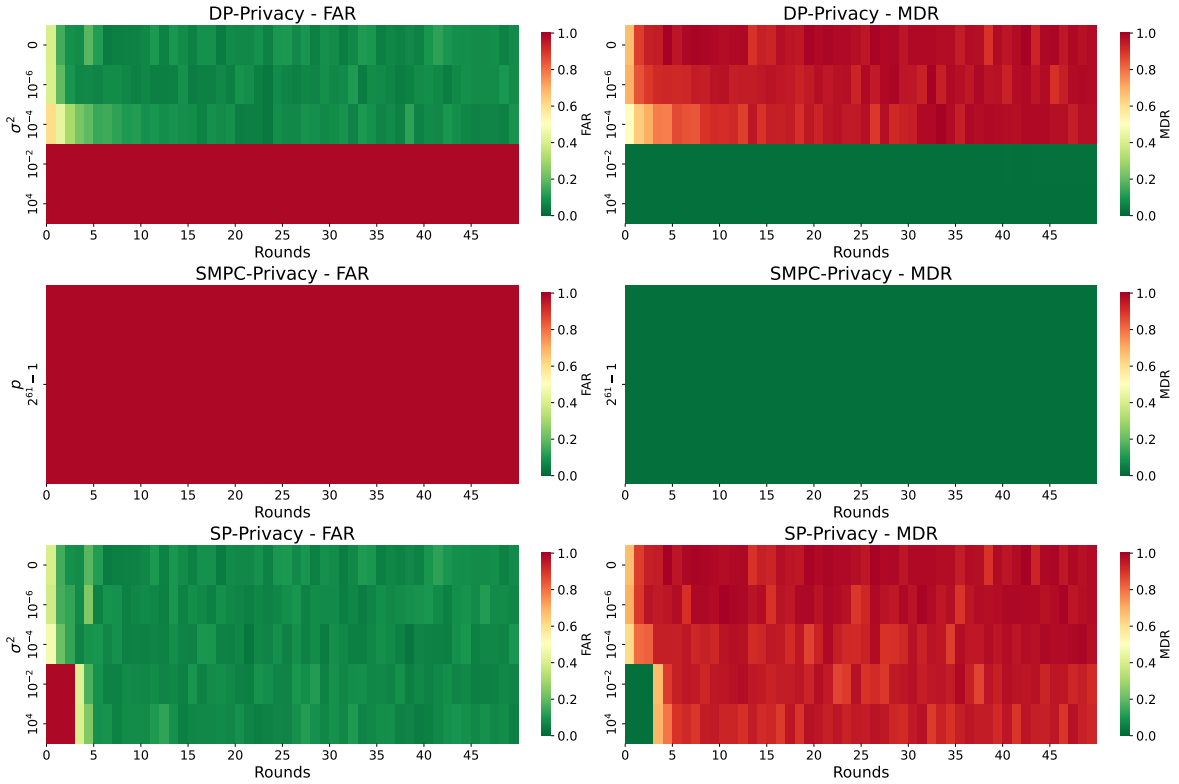


Figure 5.8: MDR and FAR for all privacy methods with the ALIE attack.

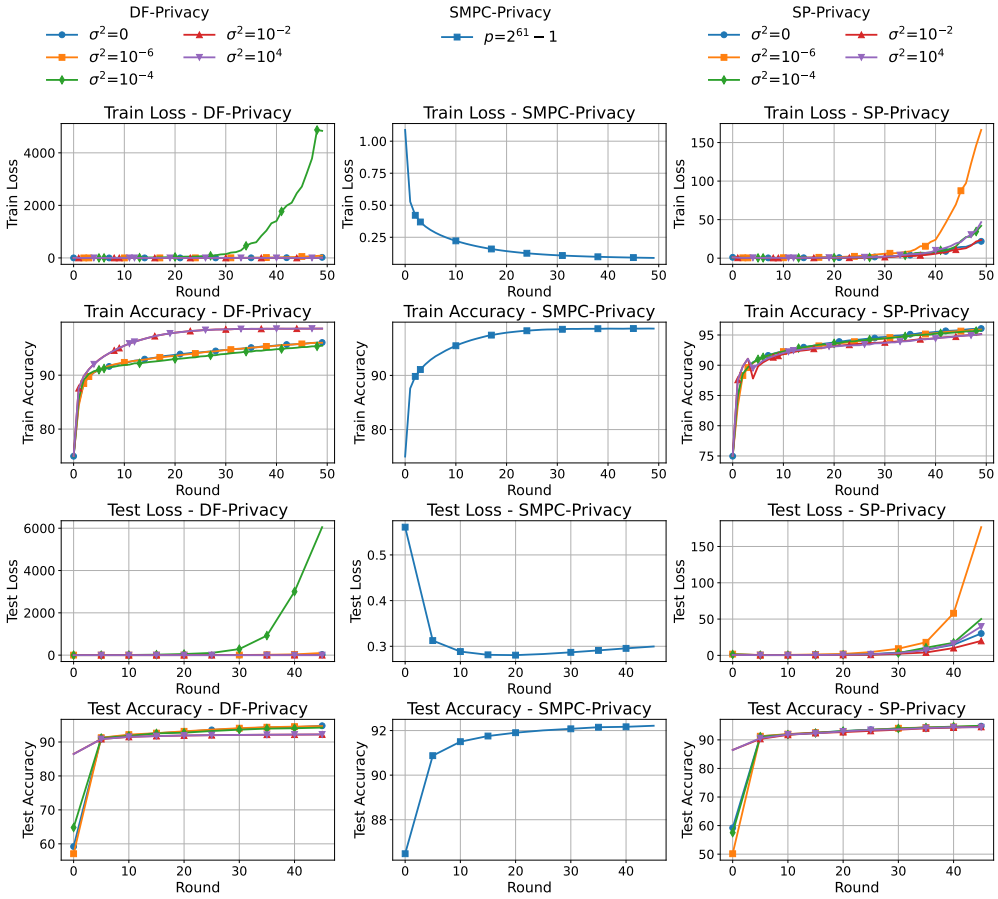


Figure 5.9: Testing and training losses and accuracies for all privacy methods with the ALIE attack.

| Privacy | Param | train_loss | train_accuracy | test_loss | test_accuracy |
|---|---|---|---|---|---|
| DF-Privacy | 0 | 21.824384 | 96.067333 | 30.140204 | 94.785556 |
| DF-Privacy | $10^{-6}$ | 89.555000 | 96.015333 | 102.251220 | 94.685000 |
| DF-Privacy | $10^{-4}$ | 4837.107944 | 95.473667 | 6050.724498 | 94.242222 |
| DF-Privacy | $10^{-2}$ | 0.088760 | 98.639000 | 0.299552 | 92.215556 |
| DF-Privacy | $10^{4}$ | 0.088734 | 98.657000 | 0.299554 | 92.223333 |
| SMPC-Privacy | $2^{61}-1$ | 0.088760 | 98.639000 | 0.299552 | 92.215556 |
| SP-Privacy | 0 | 21.824384 | 96.067333 | 30.140204 | 94.785556 |
| SP-Privacy | $10^{-6}$ | 166.365982 | 95.894667 | 176.641025 | 94.768889 |
| SP-Privacy | $10^{-4}$ | 41.930449 | 95.747333 | 50.093372 | 94.943333 |
| SP-Privacy | $10^{-2}$ | 23.602709 | 95.162333 | 19.913434 | 94.548333 |
| SP-Privacy | $10^{4}$ | 46.679535 | 95.115333 | 39.770453 | 94.413333 |

Table 5.5: Final performance of each privacy mechanism under ALIE attack.

### 5.5.5 Sign-Flipping

Figure 5.10 shows heatmaps of the False Alarm Rate (FAR) and MisDetection Rate (MDR) for differential privacy, SMPC, and subspace perturbation under the sign-flipping attack. For differential privacy and SMPC, FAR is high across most privacy settings, while MDR is high for differential privacy at low noise and for subspace perturbation at higher privacy levels. SMPC shows high FAR but maintains a low MDR. For subspace perturbation, FAR is low at small noise but increases with higher privacy, and MDR is generally high throughout the rounds. These results show that the sign-flipping attack is difficult to detect, especially when privacy protection is strong.

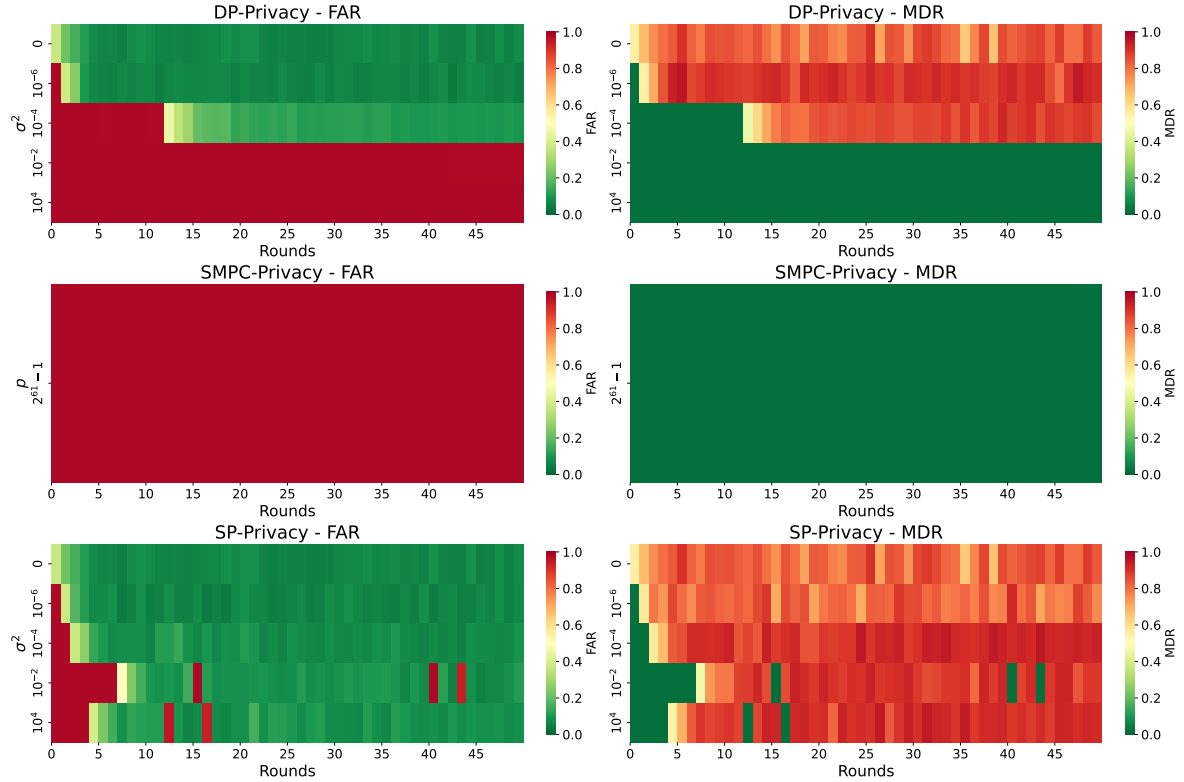


Figure 5.10: MDR and FAR for all privacy methods with the sign-flipping attack.

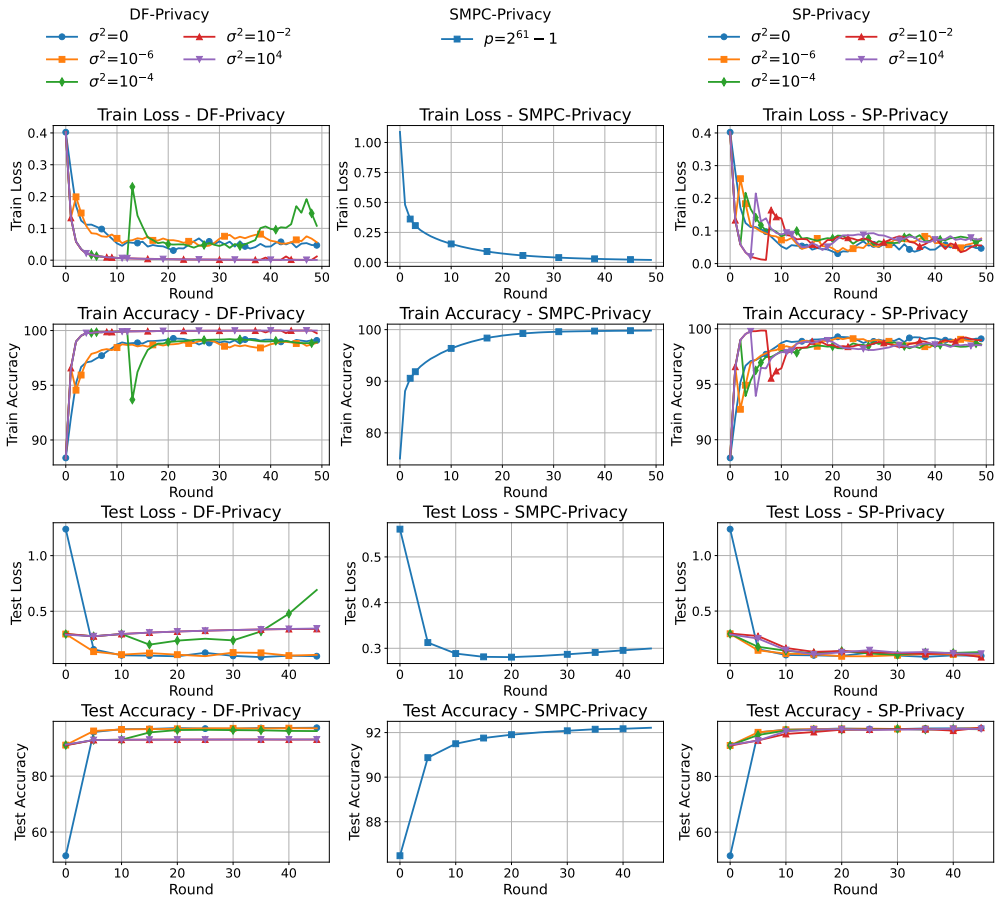


Figure 5.11: Testing and training losses and accuracies for all privacy methods with the sign-flipping attack.

| Privacy | Param | train_loss | train_accuracy | test_loss | test_accuracy |
|---|---|---|---|---|---|
| DF-Privacy | 0 | 0.046813 | 99.097000 | 0.096447 | 97.314444 |
| DF-Privacy | $10^{-6}$ | 0.056393 | 98.876667 | 0.108465 | 97.115000 |
| DF-Privacy | $10^{-4}$ | 0.108145 | 98.936000 | 0.691567 | 96.121111 |
| DF-Privacy | $10^{-2}$ | 0.011646 | 99.751000 | 0.341160 | 93.097778 |
| DF-Privacy | $10^{4}$ | 0.000807 | 99.997333 | 0.345165 | 93.107778 |
| SMPC-Privacy | $2^{61}-1$ | 0.020690 | 99.822000 | 0.299552 | 92.216111 |
| SP-Privacy | 0 | 0.046813 | 99.097000 | 0.096447 | 97.314444 |
| SP-Privacy | $10^{-6}$ | 0.077181 | 98.575000 | 0.103127 | 97.357222 |
| SP-Privacy | $10^{-4}$ | 0.071827 | 98.587000 | 0.130480 | 96.932778 |
| SP-Privacy | $10^{-2}$ | 0.055484 | 98.972000 | 0.085067 | 97.462778 |
| SP-Privacy | $10^{4}$ | 0.075264 | 98.516000 | 0.115446 | 97.007778 |

Table 5.6: Final performance of each privacy mechanism under the sign-flipping attack.

### 5.5.6 Label-Flipping

Figure 5.12 shows heatmaps of the False Alarm Rate (FAR) and MisDetection Rate (MDR) for differential privacy, SMPC, and subspace perturbation under the label-flipping attack. For differential privacy and SMPC, FAR is high for most privacy settings, while MDR is high for differential privacy at low noise and for subspace perturbation at higher privacy. SMPC shows high FAR but maintains low MDR across all rounds. For subspace perturbation, FAR is low at small noise but increases with higher privacy, while MDR rises quickly as privacy strengthens. These results indicate that label-flipping attacks are also difficult to detect, especially when privacy protection is strong.

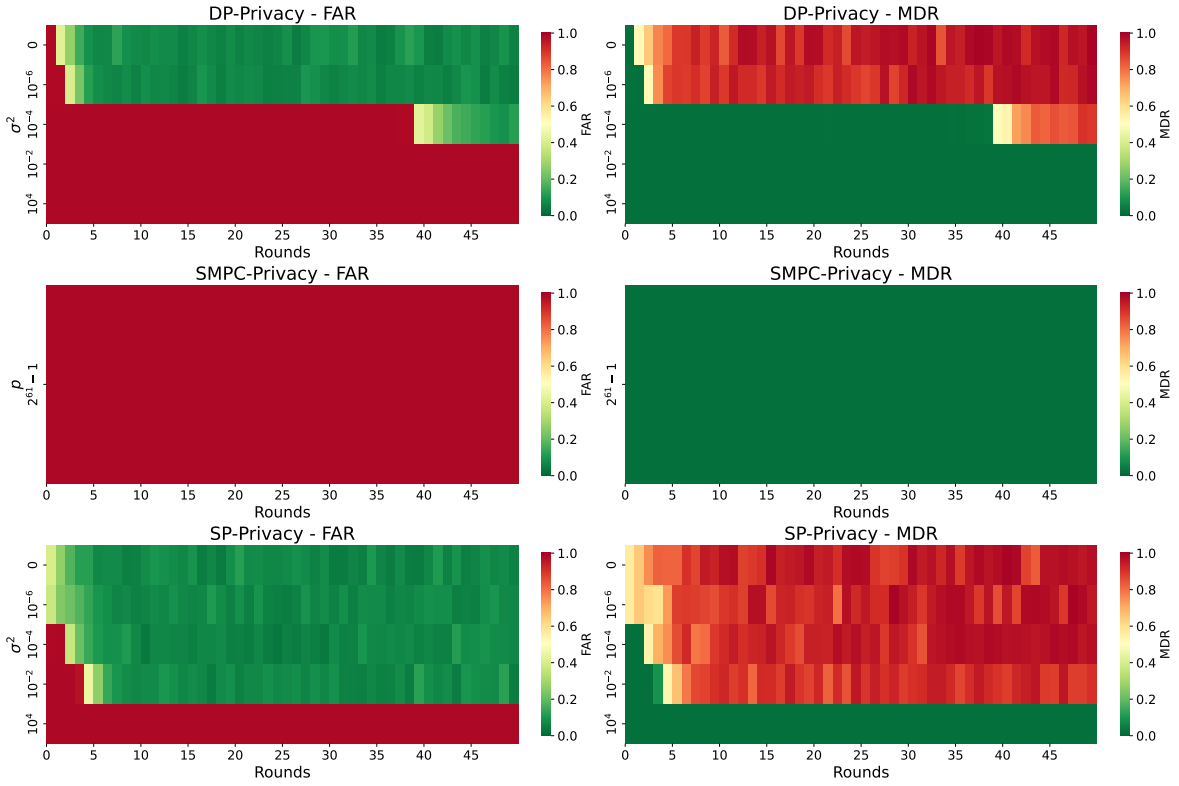


Figure 5.12: MDR and FAR for all privacy methods with the label-flipping attack.

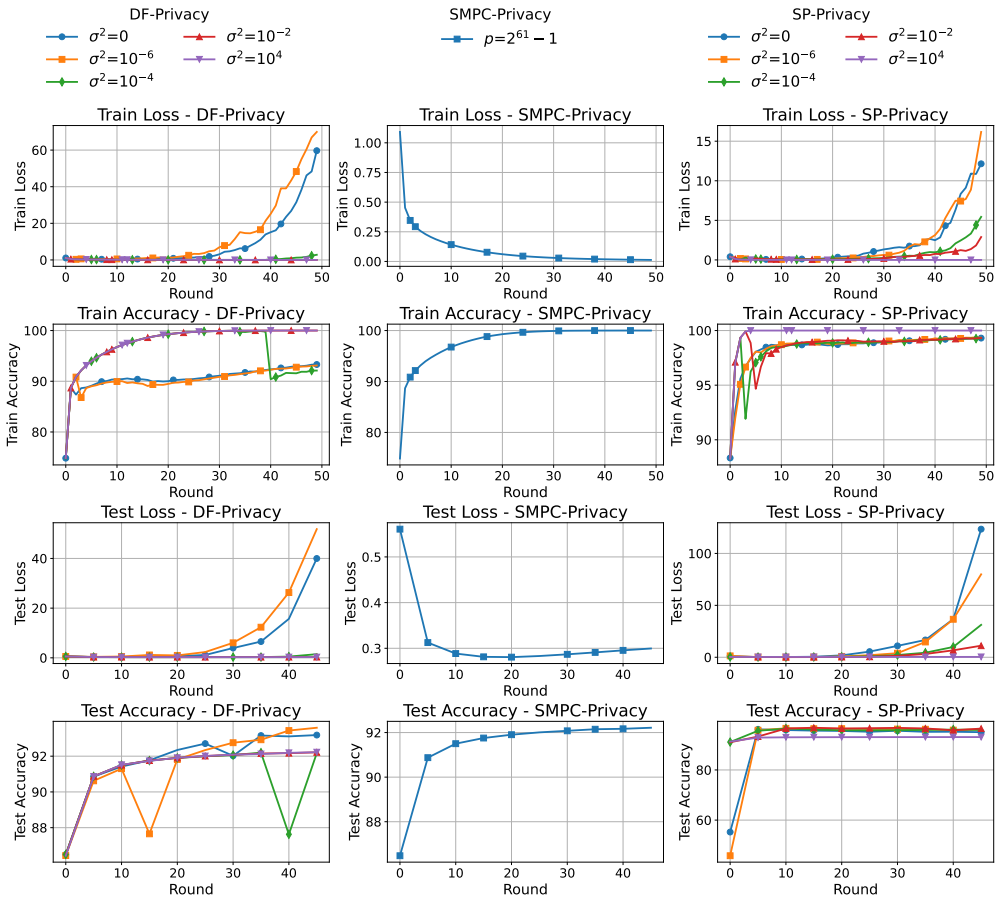


Figure 5.13: Testing and training losses and accuracies for all privacy methods with the label-flipping attack.

| Privacy | Param | train_loss | train_accuracy | test_loss | test_accuracy |
|---|---|---|---|---|---|
| DF-Privacy | 0 | 59.681298 | 93.304333 | 39.984239 | 93.181111 |
| DF-Privacy | $10^{-6}$ | 69.913095 | 93.110000 | 51.792063 | 93.590556 |
| DF-Privacy | $10^{-4}$ | 2.821447 | 92.109333 | 1.466957 | 92.168333 |
| DF-Privacy | $10^{-2}$ | 0.012811 | 100.000000 | 0.299552 | 92.215556 |
| DF-Privacy | $10^{4}$ | 0.012808 | 100.000000 | 0.299554 | 92.223333 |
| SMPC-Privacy | $2^{61}-1$ | 0.012811 | 100.000000 | 0.299552 | 92.215556 |
| SP-Privacy | 0 | 12.147219 | 99.307667 | 123.389141 | 95.066667 |
| SP-Privacy | $10^{-6}$ | 16.170015 | 99.346333 | 79.873362 | 96.193333 |
| SP-Privacy | $10^{-4}$ | 5.439566 | 99.284333 | 31.209541 | 95.951667 |
| SP-Privacy | $10^{-2}$ | 2.891573 | 99.310000 | 11.123221 | 96.483889 |
| SP-Privacy | $10^{4}$ | 0.000468 | 100.000000 | 0.345165 | 93.107778 |

Table 5.7: Final performance of each privacy mechanism under the label-flipping attack.

### 5.5.7 Privacy Preserving Mechanisms Trade-Off Performance

While the trade-off between privacy and accuracy in the shown mechanisms is well documented in the literature, the trade-off between privacy and robustness is less documented. The results in this chapter however, do give an insight into the performances of the mechanisms with the most known Byzantine attacks. Differential Privacy offers privacy robustness tuning, yet at high privacy budgets, both adversarial detection and model accuracy can suffer. SMPC provides strong, persistent privacy guarantees and stable accuracy but suffers from consistently high false alarm rates, limiting its practical robustness in adversarial settings. Subspace Perturbation often delivers the best balance under moderate privacy, especially against additive and label-flipping attacks, yet it too struggles as privacy requirements become more stringent. Notably, no mechanism achieves both low FAR and MDR rates and high accuracy under all attack types when privacy is strong, highlighting the inherent limitations imposed by the privacy robustness trade-off. These findings underscore the importance of carefully tuning privacy parameters and combining multiple defence strategies to effectively balance privacy and robustness in decentralised federated learning systems.

# 6 Conclusion, and Future Work

## 6.1 Conclusion

In this work, we investigated the balance between privacy preservation and adversarial robustness in decentralised federated learning. We studied the impact of several core privacy mechanisms, including Differential Privacy, Secure Multi-Party Computation, and Subspace Perturbation, within a decentralised federated learning protocol. Our analysis extended known information-theoretic bounds and showed how privacy-preserving mechanisms change the way adversarial detection works.

Through experiments, we found that increasing privacy protection directly limits the effectiveness of adversarial detection. This confirms the fundamental limit that no single mechanism can achieve both perfect privacy and perfect robustness at the same time. As a result, system designers must carefully adjust privacy parameters and detection strategies to meet the needs and threat models of their applications.

We also found that privacy mechanisms differ in their practical effects. Differential Privacy allows for flexible trade-offs through its privacy budget. SMPC offers strong protection but adds more complexity. Subspace Perturbation uses decentralized structures to improve privacy. In all these methods, adding noise or masking during training can disrupt adversarial inference, especially without causing a major loss in model accuracy.

These results highlight the importance of a careful approach to privacy and robustness in federated learning. Careful parameter choices, thoughtful network design, and extra defence strategies such as early stopping, inexact updates, or quantisation are all recommended for better protection. Overall, this study provides useful guidance for deploying privacy-preserving federated learning systems and managing the ongoing challenge of balancing privacy with robustness.

## 6.2 Future Work

Several directions emerge from this work that warrant further investigation. One natural extension is to broaden the scope of evaluated mechanisms by including approaches such as homomorphic encryption, trusted execution environments, or hybrid schemes that combine multiple privacy-preserving and robustness-enhancing techniques. Another promising direction is the development of adaptive frameworks that can dynamically adjust privacy parameters and detection thresholds in response to changing network conditions, observed attack patterns, and performance feedback.Finally, more sophisticated threat models should be explored, including colluding, adaptive, and multi-stage adversaries, together with countermeasures that can evolve alongside these threats.

# Bibliography


[1] G. Baruch, M. Chen, and M. Chain, "A little is enough: Circumventing defenses for distributed learning," in *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019, pp. 2026–2038.

[2] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. Tsitsiklis, "On distributed averaging algorithms and quantization effects," *Automatic Control, IEEE Transactions on*, vol. 54, 2009.

[3] Z. Chen, M. Dahl, and E. G. Larsson, "Decentralized learning over wireless networks: The effect of broadcast with random access," in *2023 IEEE 24th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2023, pp. 316–320.

[4] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, vol. 54, pp. 1273–1282, 2017.

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine learning, vol. 3, no. 1, pp. 1–122*, 2011.

[7] G. Zhang and R. Heusdens, "Distributed optimization using the primal-dual method of multipliers," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 173–187, 2017.

[8] T. W. Sherson, R. Heusdens, and W. B. Kleijn, "Derivation and analysis of the primal-dual method of multipliers based on monotone operator theory," *IEEE transactions on signal and information processing over networks*, vol. 5, no. 2, pp. 334–347, 2018.

[9] R. Heusdens and G. Zhang, "Distributed optimisation with linear equality and inequality constraints using pdmm," *IEEE Transactions on Signal and Information Processing over Networks*, 2024.

[10] M. Mageshwari and R. Naresh, "Decentralized data privacy protection and cloud auditing security management," in *2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 2022, pp. 103–109.

[11] C. Dwork, F. McSherry, K. Nissim, A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory of Cryptography Conf. , pp. 265-284*, 2006.

[12] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[13] R. Cramer, I. B. Damgård, and J. B. Nielsen, *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.

[14] E. Nozari, P. Tallapragada, and J. Cortés, "Differentially private distributed convex optimization via functional perturbation," *IEEE Trans. Control Netw. Syst., vol. 5, no. 1, pp 395-408*, 2018.

[15] Q. Li, J. S. Gundersen, R. Heusdens and M. G. Christensen, "Privacy-preserving distributed processing: Metrics, bounds, and algorithms," in *IEEE Trans. Inf. Forensics Secur.*, vol. 16, 2021, pp. 2090–2103.

[16] Q. Li, R. Heusdens, and M. G. Christensen, "Convex optimisation-based privacy-preserving distributed average consensus in wireless sensor networks," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 5895–5899.

[17] Q. Li, R. Heusdens and M. G. Christensen, "Convex optimization-based privacy-preserving distributed least squares via subspace perturbation," in *Proc. Eur. Signal Process. Conf.*, 2020.

[18] Q. Li, R. Heusdens, and M. G. Christensen, "Privacy-preserving distributed optimization via subspace perturbation," *IEEE Transactions on Signal Processing*, vol. 68, pp. 5983–5996, 2020.

[19] X. Dong, Z. Wu, Q. Ling, and Z. Tian, "Byzantine-robust distributed online learning: Taming adversarial participants in an adversarial environment," *IEEE Transactions on Signal Processing*, vol. 72, pp. 235–248, 2024.

[20] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, vol. 108, 2020, pp. 2938–2948.

[21] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2018, p. 6106–6116.

[22] Y. Li, X. Wei, Y. Li, Z. Dong, and M. Shahidehpour, "Detection of false data injection attacks in smart grid: A secure federated deep learning approach," *IEEE Transactions on Smart Grid*, vol. PP, 2022.

[23] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, 2017.

[24] G. Damaskinos, E. M. El Mhamdi, R. Guerraoui, R. Patra, and M. Taziki, "Asynchronous Byzantine machine learning (the case of SGD)," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, 2018, pp. 1145–1154.

[25] O. Shalom, A. Leshem, and A. Scaglione, "Localization of data injection attacks on distributed m-estimation," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 8, pp. 655–669, 2022.

[26] G. Zhang and R. Heusdens, "Distributed optimization using the primal-dual method of multipliers," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 173–187, 2017.

[27] Q. Li, R. Heusdens, and M. G. Christensen, "Convex optimisation-based privacy-preserving distributed average consensus in wireless sensor networks," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 5895–5899.

[28] K. Niwa, G. Zhang, and W. B. Kleijn, "Edge consensus computing for heterogeneous data sets," in *2018 IEEE Statistical Signal Processing Workshop (SSP)*, 2018, pp. 80–84.

[29] K. Niwa, N. Harada, G. Zhang, and W. B. Kleijn, "Edge-consensus learning: Deep learning on p2p networks with nonhomogeneous data," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Association for Computing Machinery, 2020, p. 668–678. [Online]. Available: https://doi.org/10.1145/3394486.3403109

[30] 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

[31] The White House, "Consumer data privacy in a networked world: A framework for protecting privacy and promoting innovation in the global digital economy," 2012, accessed: 2024-09-10. [Online]. Available: https://obamawhitehouse.archives.gov/sites/default/files/privacy-final.pdf

[32] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *International Conference on Learning Representations (ICLR)*, 2020, arXiv preprint arXiv:1907.02189.

[33] A. Bellet, A.-M. Kermarrec, and E. Lavoie, "D-cliques: Compensating for data heterogeneity with topology in decentralized federated learning," *arXiv preprint arXiv:2104.07365*, 2021.

[34] L. Paninski, "Estimation of entropy and mutual information," *Neural Computation*, vol. 15, no. 6, pp. 1191–1253, 2003, communicated by Jonathan Victor.

[35] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[36] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014. [Online]. Available: http://dx.doi.org/10.1561/0400000042

[37] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[38] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, 1988, pp. 1–10.

[39] D. R. Stinson, *Cryptography: Theory and Practice*, 3rd ed. Chapman & Hall/CRC, 2005.

[40] C. Blundo, A. D. Santis, D. R. Stinson, and U. Vaccaro, "Perfectly-secure key distribution for dynamic conferences," in *Advances in Cryptology – CRYPTO '92*, ser. Lecture Notes in Computer Science, vol. 740. Springer, 1993, pp. 471–486.

[41] I. Damgård, V. Pastro, N. P. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," in *Advances in Cryptology – CRYPTO 2012*, ser. Lecture Notes in Computer Science, vol. 7417. Springer, 2012, pp. 643–662.

[42] Q. Li, J. S. Gundersen, M. Lopuhaä-Zwakenberg, and R. Heusdens, "Adaptive differentially quantized subspace perturbation (adqsp): A unified framework for privacy-preserving distributed average consensus," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 1780–1793, 2024.

[43] S. Choudhary, A. Kolluri, and P. Saxena, "Attacking byzantine robust aggregation in high dimensions," in *2024 IEEE Symposium on Security and Privacy (SP)*, 2024, pp. 1325–1344.

[44] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research. PMLR, 2018.

[45] J. Zhang, B. Li, C. Chen, L. Lyu, S. Wu, S. Ding, and C. Wu, "Delving into the adversarial robustness of federated learning," 2023. [Online]. Available: https://arxiv.org/abs/2302.09479

[46] X. Li, Y. Luo, Y. Zhu, S. Lin, and Y. Shi, "Learn: Byzantine-robust decentralized federated learning," *arXiv preprint*, vol. abs/2307.14747, 2023, available: https://arxiv.org/abs/2307.14747.

[47] S. G. Vlaski, J. Li, and A. H. Sayed, "Robust and efficient aggregation for distributed learning," *arXiv preprint*, vol. abs/2204.00586, 2022, available: https://arxiv.org/abs/2204.00586.

[48] Y. Xie, N. Xiong, T. Li, and V. Smith, "Remove-then-clip: Byzantine-robust decentralized learning," in *Proc. 35th AAAI Conf. on Artificial Intelligence (AAAI-21)*, Virtual, 2021, pp. 4202–4209.

[49] J. Xu, S.-L. Huang, L. Song, and T. Lan, "Byzantine-robust federated learning through collaborative malicious gradient filtering," 2023. [Online]. Available: https://arxiv.org/abs/2109.05872

[50] C. F. Fung, C. J. M. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," in *Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, 2020, pp. 293–313.

[51] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, "Rsa: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," *AAAI*, vol. 33, no. 1, pp. 1544–1551, 2019.

[52] D. Alistarh, Z. Wang, J. Li, M. Pilanci, and M. Jordan, "Byzantine stochastic gradient descent," *arXiv preprint arXiv:1803.09877*, 2018.

[53] Z. Wu, T. Chen, and Q. Ling, "Byzantine-resilient decentralized stochastic optimization with robust aggregation rules," 2023. [Online]. Available: https://arxiv.org/abs/2206.04568

[54] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*, 2012, pp. 1467–1474.

[55] ——, "Poisoning attacks against support vector machines," 2013. [Online]. Available: https://arxiv.org/abs/1206.6389

[56] Z. Wu, T. Chen, and Q. Ling, "Byzantine-resilient decentralized stochastic optimization with robust aggregation rules," *IEEE Transactions on Signal Processing*, vol. 71, pp. 3179–3195, 2023.

[57] P. Blanchard, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *arXiv preprint arXiv:1705.09410*, 2017.

[58] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, 2006.

[59] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *NeurIPS*, 2017.

[60] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *ICML*, 2018.

[61] G. Baruch, M. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," in *NeurIPS*, 2019.

[62] C. Fung, C. J. Yoon, and I. Beschastnikh, "Limitations of group norms for robust aggregation in federated learning," in *NeurIPS*, 2020.

[63] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "Dba: Distributed backdoor attacks against federated learning," in *ICLR*, 2020.

[64] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in *ICML*, 2012.

# APPENDIX A

# On the Privacy-Robustness Trade-off in Distributed Average Consensus

Zarè Palanciyan
*Delft University of Technology*
the Netherlands

Qiongxiu Li
*Aalborg University*
Denmark

Richard Heusdens
*Netherlands Defence Academy*
*Delft University of Technology*
the Netherlands

***Abstract*—Distributed consensus algorithms face a dual challenge in modern networked systems: safeguarding sensitive data through privacy-preserving mechanisms while maintaining robustness against adversarial nodes (e.g., Byzantine faults). While prior work addresses these goals separately, their interplay remains poorly understood, particularly in scenarios where output accuracy must be preserved. In this work, we reconcile these objectives by integrating a subspace perturbation framework, which guarantees privacy by confining noise to redundant network subspaces, with a median absolute deviation (MAD)-based thresholding mechanism to detect active adversarial nodes transmitting corrupted data. Through in-depth analysis, we demonstrate that enhancing privacy via subspace perturbation inherently limits the discriminative power of MAD-based detection, as adversarial updates become statistically indistinguishable from privacy-preserving perturbations. Numerical simulations quantify this tension, demonstrating that as privacy guarantees strengthen, the ability to detect active adversaries diminishes. These findings highlight a core challenge in distributed consensus—achieving both strong privacy and Byzantine robustness simultaneously is inherently difficult.**

***Index Terms*—Primal-dual method of multipliers (PDMM), privacy, subspace perturbation, adversary, detection, median absolute deviation (MAD), privacy-robustness trade-off**

## I. Introduction

In recent years, a rising trend is gaining traction which aims to develop a wide range of techniques to perform distributed computations [1]. These techniques enable collaborative data processing across decentralised nodes without a centralised coordinator. For example, this can be seen in applications ranging from wireless sensor networks [2], optimisation [3], and federated learning [4]. Distributed optimisation algorithms such as the alternating direction method of multipliers (ADMM) [5] and the primal-dual method of multipliers (PDMM) [6]–[8] have gained popularity in distributed frameworks. As these algorithms are applied across more fields, they increasingly handle sensitive data, making privacy protection vital [9]. Traditional approaches, such as differential privacy (DP) [10], [11] and secure multiparty computation (SMPC) [12], aim to address these concerns by protecting sensitive data but often do so at the cost of accuracy or increased computational complexity [13], [14]. To overcome these limitations, new methods have been developed that maintain both accuracy and computational efficiency. One such approach is the subspace perturbation framework introduced in [15]–[17] and its variants [18]–[20].

Privacy is not the only concern in decentralised networks. In addition to data being extracted from the network, corrupt data can also be injected into a distributed system [21]. Various types of attacks can introduce corrupt data, such as backdoor attacks in a federated learning environment [22], [23]. Additionally, attacks can be designed to prevent the network from converging to the optimal value. This can be achieved, for example, through random Gaussian attacks [21] or by transmitting malicious data to poison the network [24]. To counter these attacks, various robust detection algorithms have been developed, such as the Krum algorithm [25]. Here, the node calculates the proximity of its neighbours and the similarity of their transmitted data. It then selects the node with the smallest distance to the others as the true update. Another method, called Kardam [26], computes the Lipschitz coefficients of its neighbours and accepts data from those neighbours whose values fall within an acceptable range around the median of the Lipschitz coefficients. Another approach, proposed in [27], detects corrupt nodes by calculating at every time instant the normalised difference of transmitted data among neighbours and determining the maximum deviation from the median. Neighbours of which the averaged distance to the median exceeds a certain threshold are identified as being malicious. When implementing a decentralised network, both privacy and adversarial robustness must be considered. In a distributed network, the optimal solution can be achieved when the appropriate algorithm is used and all nodes share the same goal. However, in the presence of an attack, the network may diverge from its optimal output if robust detection algorithms are not in place. Detection algorithms evaluate and distinguish nodes based on the data they transmit. By comparing individual data updates to the collective behaviour of neighbouring nodes, adversarial nodes can be identified and flagged. However, if the network achieves perfect secrecy, nodes may become indistinguishable from one another, undermining the fundamental principle on which detection algorithms assess corruption.

In this paper, we investigate a fundamental trade-off between privacy preservation and adversarial detection in distributed average consensus algorithms. Specifically, we demonstrate that integrating both privacy-protecting mechanisms and attack detection capabilities creates an inherent conflict: enhancing privacy preservation (e.g., through noise injection) can inadvertently diminish the system's ability to

identify malicious or compromised nodes. Our analysis reveals that as the level of privacy preservation increases, the detection accuracy for adversarial behaviour declines accordingly, highlighting a critical design challenge for secure and privacy-aware consensus frameworks. We further consolidate these findings through numerical simulations, which quantify the trade-off and validate our theoretical claims.

The paper is organised as follows. In Section II, we define the privacy preservation methods used and describe the adversarial models. Section III presents the problem setup, the network attack model, and the metrics used to demonstrate the privacy-robustness trade-off. In Section IV, we introduce the proposed detection algorithm for active adversarial nodes. Section V provides numerical validation of our findings, and finally, Section VI presents our conclusions.

## II. Preliminaries

We present a simple undirected connected graph $G$ as $G = (\mathcal{V}, \mathcal{E})$, where the set of nodes in the network is represented by $\mathcal{V} = \{1, 2, \dots, n\}$ and the set of edges is represented by $\mathcal{E} = \{e_1, \dots, e_m\} \subseteq \mathcal{V} \times \mathcal{V}$. The neighbourhood of node *i* is denoted as the set $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i,j) \in \mathcal{E}\}$. The degree of node *i* is then given by $d_i = |\mathcal{N}_i|$. The distributed average consensus algorithm aims to calculate the average of the local data each node holds

$$\mathbf{s}_{\text{ave}} = \frac{1}{n} \sum_{i \in \mathcal{V}} \mathbf{s}_i \tag{1}$$

where $\mathbf{s}_i \in \mathbb{R}^q$ is the local data each node holds and $q$ the dimension of the local data.

### A. A/PDMM approach of solving average consensus

The solution to (1) can be achieved in a distributed network by implementing PDMM and reformulating the overall setup problem as follows:

$$\begin{aligned} \min_{\{\mathbf{x}_i : i \in \mathcal{V}\}} \quad & \sum_{i \in \mathcal{V}} f_i(\mathbf{x}_i) \\ \text{subject to} \quad & \forall (i,j) \in \mathcal{E} : \mathbf{B}_{i|j}\mathbf{x}_i + \mathbf{B}_{j|i}\mathbf{x}_j = \mathbf{0}, \end{aligned} \tag{2}$$

where $f_i(\mathbf{x}_i) = \frac{1}{2}||\mathbf{x}_i - \mathbf{s}_i||_2^2$ and $\mathbf{B}_{i|j} \in \mathbb{R}^{q \times q}$ is defined as $\mathbf{B}_{i|j} = \mathbf{I}_q$ if $i < j$, and $\mathbf{B}_{i|j} = -\mathbf{I}_q$ otherwise, where $\mathbf{I}_q$ denotes the $q \times q$ identity matrix. As shown in [28], problem (2) can be solved using A/PDMM. This leads to the following set of update equations for each node:

$$\mathbf{x}_i^{(t+1)} = \arg\min_{\mathbf{x}_i} \left( f_i(\mathbf{x}_i) + \sum_{j \in \mathcal{N}_i} \mathbf{z}_{i|j}^{(t)\top} \mathbf{B}_{i|j}\mathbf{x}_i + \frac{\rho d_i}{2} \|\mathbf{x}_i\|^2 \right),$$

$$(\forall j \in \mathcal{N}_i) \quad \mathbf{y}_{i|j}^{(t+1)} = \mathbf{z}_{i|j}^{(t)} + 2\rho \mathbf{B}_{i|j}\mathbf{x}_i^{(t+1)}, \tag{3}$$

$$(\forall j \in \mathcal{N}_i) \quad \mathbf{z}_{j|i}^{(t+1)} = (1-\theta)\mathbf{z}_{j|i}^{(t)} + \theta \mathbf{y}_{i|j}^{(t+1)}, \tag{4}$$

where $\theta$ is the averaging parameter and $\rho$ controls the convergence rate. For $\theta = \frac{1}{2}$ (Douglas-Rachford splitting), $\frac{1}{2}$-averaged PDMM is achieved, which is equivalent to the classical ADMM algorithm [28].

### B. Privacy

There are various privacy-preserving distributed average consensus algorithms proposed, such as DP-based [29]–[31], SMPC based [32]–[34] and subspace-perturbation based approaches [15]–[17]. In this paper, we deploy subspace perturbation to achieve privacy preservation, motivated by its efficiency and flexibility; it has been shown to achieve similar privacy guarantees to SMPC and DP-based approaches under specific parameter configurations [20]. The implementation of subspace perturbation is straightforward: it involves sampling the initialised auxiliary variable $\mathbf{z}^{(0)} \in \mathbb{R}^{mq}$ from a high-variance noise distribution to protect local node data through statistical obfuscation. This is achieved by splitting the space into two subspaces, the convergent subspace and the non-convergent subspace. Privacy is imposed by perturbing the non-convergent subspace with noise, while perturbations in the non-convergent subspace do not affect the output accuracy. As shown in [17], perfect secrecy can asymptotically be achieved this way, by introducing finite variance in the noise in the non-convergent subspace of the auxiliary variable $\mathbf{z}^{(0)}$.

### C. Adversarial models

In this work, we consider three types of nodes that can exist within the network. The first type is the honest node, which follows the averaging process as intended and does not attempt to infer the local data of other nodes. The other two types are adversarial nodes: passive adversarial nodes (also known as honest-but-curious) and active adversarial nodes. Passive adversarial nodes follow the protocol's instructions similar to honest nodes but attempt to infer as much information as possible, in collaboration with other passive adversarial nodes, about the local data $s_i$ of the honest nodes. In contrast, active adversarial nodes seek to disrupt the network by transmitting arbitrary updates based on their malicious intentions, which could cause the network to diverge or converge to a malicious point.

Let $\mathcal{V}_h$ denote the set of honest nodes and $\mathcal{V}_c$ the set of adversarial nodes such that $\mathcal{V} = \mathcal{V}_h \cup \mathcal{V}_c$ and $\mathcal{V}_h \cap \mathcal{V}_c = \emptyset$. In addition, let $\mathcal{V}_{c,p}$ denote the set of passive adversarial nodes and $\mathcal{V}_{c,a}$ the set of active adversarial nodes so that $\mathcal{V}_c = \mathcal{V}_{c,p} \cup \mathcal{V}_{c,a}$ and $\mathcal{V}_{c,p} \cap \mathcal{V}_{c,a} = \emptyset$.

## III. Problem Definition

The local data utilised in private distributed networks for which PDMM can be applied to, should not be revealed to outsiders or adversarial nodes. To solve this issue, one must implement privacy-preserving frameworks in their network. The works in [17] [20] [15] have shown that it is possible to implement such a framework in PDMM and to achieve perfect secrecy, without sacrificing the output correctness of the network. However, our findings in this work show there is another trade-off which occurs when perfect secrecy is achieved. We found that higher levels of privacy in a network for the private local data of its nodes come at the expense of detecting adversarial nodes that corrupt their local data

with malicious intent. This trade-off highlights the difficulty in balancing between privacy and adversarial node detection.

### A. Active adversarial attack

An adversarial model can utilise multiple different attacks to reach its goal [35]. However, in this work, we will only focus on a single attack, in which the adversarial node corrupts its local data to make the network converge to a non-optimal point.

### B. Trade-off metrics

To discuss the trade-off between the effectiveness of privacy-preserving techniques and adversarial detection methods, four metrics will be compared.

*a) Information-theoretical privacy metric:* measures the mutual information between the private data and all information available to the adversary. Let $\mathcal{O}$ denote the set of information obtained by the adversary and $\mathcal{X}_i$ the private data of node $i$. The mutual information $I(X_i;\mathcal{O})$ is given by

$$I(X_i;\mathcal{O}) = H(X_i) - H(X_i|\mathcal{O}) \leq H(X_i), \tag{5}$$

where $H(\cdot)$ denotes the (Shannon) entropy. If $I(X_i;\mathcal{O}) = 0$, the adversary cannot gain any information about the private data by observing $\mathcal{O}$. If $I(X_i;\mathcal{O}) = H(X_i)$, the adversary has complete knowledge about the private data. Thus, higher mutual information indicates greater potential privacy leakage.

*b) Output correctness:* measures the distance of the output of the network to the optimal solution. This is assessed by taking the mean square error (MSE) between the $\mathbf{x}_i$-values and the optimal solution $\mathbf{x}^*$, given by $\frac{1}{n}||\mathbf{x}_i^{(t)} - \mathbf{x}^*||^2$.

*c) False alarm rate (FAR):* measures whether nodes implementing the detection algorithm are misclassifying honest nodes as active adversarial nodes. Let $D(i,j)$ denote the number of times a node has been identified of being malicious within a time frame of $L$ samples. The FAR is defined as $\mathit{FAR}(k) = \frac{1}{|\mathcal{E}_H|}\sum_{(i,j)\in\mathcal{E}_H}\mathbb{I}(D(i,j)(kL) > \frac{L}{2})$, where $\mathcal{E}_H \subseteq (\mathcal{V}\setminus V_{c,a}) \times (\mathcal{V}\setminus V_{c,a})$, $k \in \mathbb{N}$, and $\mathbb{I}$ is the indicator function.

*d) Rate of misdetection (MDR):* measures whether nodes implementing the detection algorithm are misclassifying active adversarial nodes as honest nodes. This is quantified as follows: $\mathit{MDR}(k) = \frac{1}{|\mathcal{E}_A|}\sum_{(i,j)\in\mathcal{E}_A}\mathbb{I}(D(i,j)(kL) < \frac{L}{2})$, where $\mathcal{E}_A \subseteq (\mathcal{V}\setminus V_{c,a}) \times V_{c,a}$.

## IV. Method of Detection

The method of detection of active adversarial nodes in this paper is based on the work of [27]. Here, the method of determining whether a node is an active adversary or not is defined with the transmitted variable $\mathbf{y}_{\cdot|i}$ node $i$ receives from its neighbours. The following assumptions are made to implement the detection algorithm.

*Assumption 1:* The amount of active adversarial neighbours a node has is less than half of the total amount of its neighbours. Hence, $\frac{d_i}{2} > |\{j \in \mathcal{N}_i \cap \mathcal{V}_{c,a} \mid (i,j) \in \mathcal{E}\}|$.

*Assumption 2:* The graph $G$ remains connected even when the node-set $\mathcal{V}_{c,a}$ is removed.

**Algorithm 1** Detection and Mitigation

**Input:** Threshold scaling $\alpha$, Segment length $L$.
Set $D(i,j) = 0$ for each $i, j \in V$.
**for** $t = 1, 2, \ldots$ **do**
  **for** all $i \in \mathcal{V}$ **do**
    **for** each agent $j \in N_i$ **do**
      Compute $\Delta Y_{i,j}(t), \delta_i$ according to (6), (7)
    **end for**
    **if** $\Delta Y_{i,j}(t) > \delta_i$ **then**
      Increase $D(i,j)$.
    **end if**
    **if** $t \equiv 0 \pmod{L}$ **then**
      **if** $D(i,j) > \frac{L}{2}$ **then**
        Node $i$ ignores[1] the updates of $j$ for the next L iterations, and stops sending updates to $j$.
      **else**
        Node $i$ continues using the updates of node $j$.
      **end if**
      Set $D(i,j)$ = 0.
    **end if**
  **end for**
**end for**

If an adversarial node were to steer away from the objective of the honest nodes by corrupting its local data $\mathbf{s}_i$, its transmitted variable $\mathbf{y}_{i|\cdot}$ would have a larger distance to the transmitted variables of other honest nodes. In the case of PDMM, because of the minus-sign difference between the $\mathbf{y}$ variables, the absolute value of $\mathbf{y}_{i|j}$ is taken. Leveraging Assumption 1, the median of the data of neighbouring nodes will be the data value of one of the honest neighbours.

Let $\mathbf{m}_i$ denote the median of the neighbouring data of node $i$, given by

$$\mathbf{m}_i = \text{med}\{|\mathbf{y}_{i|j}| : j \in \mathcal{N}_i\},$$

and let $\Delta Y_{i,j}$ be defined as

$$\Delta Y_{i,j} = \||\mathbf{y}_{i|j}| - \mathbf{m}_i\|_\infty, \quad j \in \mathcal{N}_i. \tag{6}$$

The values of $\Delta Y_{i,j}$ will then be compared to the scaled median absolute deviation (SMAD), given by

$$\text{SMAD}_i = \alpha\,\text{med}\{||\mathbf{y}_{i|j}| - \mathbf{m}_i|_\infty : j \in \mathcal{N}_i\}, \tag{7}$$

where $\alpha$ is a scaling factor. This threshold determines whether a node is corrupt or not. Again, leveraging Assumption 1, the SMAD is a powerful method of determining a threshold because it compares the distance other neighbours have from a neighbour that is guaranteed to be honest.

Utilising Assumption 1 and Assumption 2, the nodes can flag and isolate an adversarial neighbour using Algorithm 1 while also being able to achieve the optimal solution in the network for the objective function with its constraints. The method of detection would be implemented in PDMM between

[1]The utilisation of the updates of node $j$ stops, but node $i$ keeps receiving the updates for the next test, but does not acknowledge it as its neighbour.

(3) and (4), to determine whether or not it should use the update or acknowledge the neighbour.

The argument of the trade-off between privacy preservation and adversarial detection can be made with any detection method. This is a fundamental trade-off, as stronger privacy measures inherently reduce the ability to gather information for adversarial detection. As explained previously, if the mutual information equals zero, then independent of any detection method, no information about the private data of the adversarial nodes can be gained.

## V. Simulations

In this section, we will present the simulation results to show the trade-off between achieving higher levels of privacy and the ability to detect active adversarial nodes in the network. Here we simulated a distributed network by generating a random geometric graph (RGG) with $n = 50$ nodes and a communication radius of $r = \sqrt{\frac{2\log(n)}{n}}$, as this ensures that the graph is connected [36]. The data will be scalar-valued $(q = 1)$. We will introduce a single corrupt node $b$ in the graph, for which its local data will be $s_b = 10^4$, this will make the average converge to a non-optimal point if not removed. The other nodes will have local data which is generated with a Gaussian distribution around a mean of 25 with a variance of 30 $(s_i \sim \mathcal{N}(25, 30)) : \forall i \in (\mathcal{V} \setminus V_{c,a})$. First, for $\theta = \frac{1}{2}$ (ADMM), we set $\rho = 1, \alpha = 13, L = 5$, as for the case of $\theta = 1$ (PDMM), we set $\rho = 1, \alpha = 10, L = 2$. Figure 1 and 2 show the FAR, MDR and MSE output correctness, with the implementation of the privacy-preserving framework and the detection algorithm for ADMM and PDMM, respectively.

### A. False alarm rate

The top subplot of the figures corresponds to the FAR. It is observed that even as the noise variance in $z^{(0)}$ increases, the FAR remains low, indicating how infrequently honest nodes are incorrectly flagged as adversarial nodes. This is the case for both ADMM and PDMM.

### B. Rate of misdetection

The middle subplot shows the MDR, which decreases slower with higher noise variance. The high MDR shows that the detection algorithm often misses the adversarial node when the noise variance in $z^{(0)}$ is large. For $\sigma^2 = 10^4$ it is seen that for ADMM the MDR converges to 0, but for PDMM the MDR does not converge.

### C. Output correctness

The bottom subplot shows the MSE, demonstrating that when the MDR is nonzero, $MDR \neq 0$, the MSE remains high because the adversarial node $b$ can still skew the network's output for both ADMM and PDMM. Thus, while increased noise in $z^{(0)}$ preserves privacy, it also hinders the detection algorithm's ability to detect the adversarial node, revealing the trade-off and validating our theoretical claims.

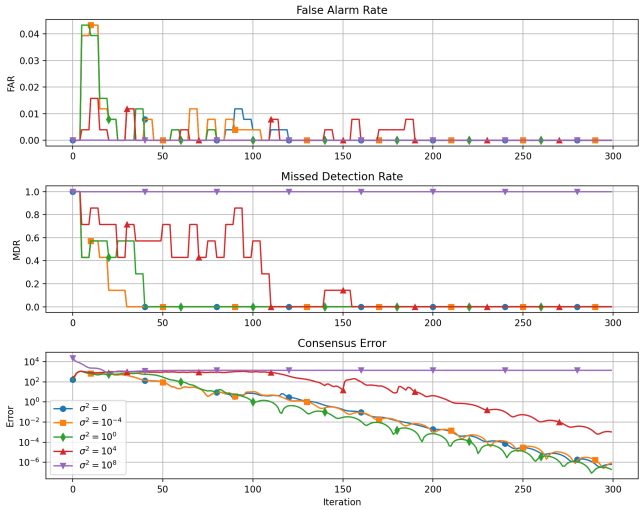


Fig. 1. ADMM simulation with different variance levels for the noise in $z^{(0)}$.

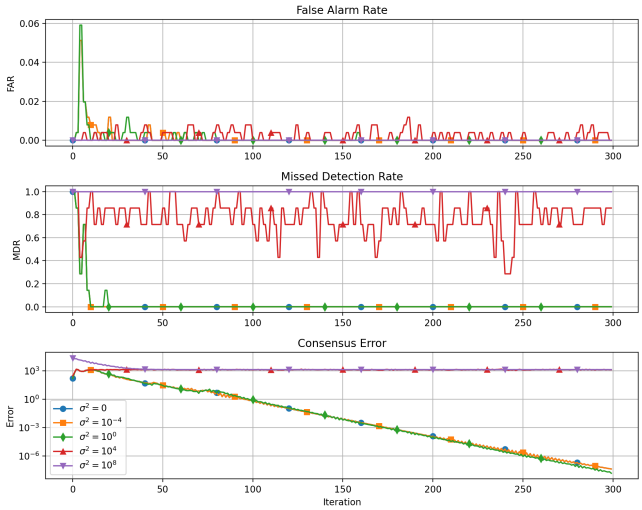


Fig. 2. PDMM simulation with different variance levels for the noise in $z^{(0)}$.

## VI. Conclusion

In this paper, we presented a hybrid approach for detecting active adversarial nodes while utilising a privacy-preserving framework for the PDMM algorithm. We have shown that when the data of nodes achieves perfect secrecy, it becomes impossible for any detection algorithm to detect active adversarial nodes in distributed average consensus algorithms. Therefore, when the mutual information converges to zero $(I(X_i; \mathcal{O}) = 0)$, no information about the private data can be inferred, making the nodes indistinguishable from one another with respect to one another's private data. Numerical results under various settings further consolidate our claims.

## References

[1] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. Tsitsiklis, "On distributed averaging algorithms and quantization effects," *Automatic Control, IEEE*

*Transactions on*, vol. 54, 2009.

[2] Z. Chen, M. Dahl, and E. G. Larsson, "Decentralized learning over wireless networks: The effect of broadcast with random access," in *2023 IEEE 24th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2023, pp. 316–320.

[3] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, vol. 54, pp. 1273–1282, 2017.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine learning, vol. 3, no. 1, pp. 1–122*, 2011.

[6] G. Zhang and R. Heusdens, "Distributed optimization using the primal-dual method of multipliers," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 173–187, 2017.

[7] T. W. Sherson, R. Heusdens, and W. B. Kleijn, "Derivation and analysis of the primal-dual method of multipliers based on monotone operator theory," *IEEE transactions on signal and information processing over networks*, vol. 5, no. 2, pp. 334–347, 2018.

[8] R. Heusdens and G. Zhang, "Distributed optimisation with linear equality and inequality constraints using pdmm," *IEEE Transactions on Signal and Information Processing over Networks*, 2024.

[9] M. Mageshwari and R. Naresh, "Decentralized data privacy protection and cloud auditing security management," in *2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 2022, pp. 103–109.

[10] C. Dwork, F. McSherry, K. Nissim, A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory of Cryptography Conf. , pp. 265-284*, 2006.

[11] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[12] R. Cramer, I. B. Damgård, and J. B. Nielsen, *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.

[13] E. Nozari, P. Tallapragada, and J. Cortés, "Differentially private distributed convex optimization via functional perturbation," *IEEE Trans. Control Netw. Syst., vol. 5, no. 1, pp 395-408*, 2018.

[14] Q. Li, J. S. Gundersen, R. Heusdens and M. G. Christensen, "Privacy-preserving distributed processing: Metrics, bounds, and algorithms," in *IEEE Trans. Inf. Forensics Secur.*, vol. 16, 2021, pp. 2090–2103.

[15] Q. Li, R. Heusdens, and M. G. Christensen, "Convex optimisation-based privacy-preserving distributed average consensus in wireless sensor networks," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 5895–5899.

[16] Q. Li, R. Heusdens and M. G. Christensen, "Convex optimization-based privacy-preserving distributed least squares via subspace perturbation," in *Proc. Eur. Signal Process. Conf.*, 2020.

[17] Q. Li, R. Heusdens, and M. G. Christensen, "Privacy-preserving distributed optimization via subspace perturbation," *IEEE Transactions on Signal Processing*, vol. 68, pp. 5983–5996, 2020.

[18] S. O. Jordan, Q. Li, and R. Heusdens, "Privacy-preserving distributed optimisation using stochastic PDMM," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2024, pp. 8571–8575.

[19] Q. Li, R. Heusdens, and M. G. Christensen, "Communication efficient privacy-preserving distributed optimization using adaptive differential quantization," *Signal Process.*, vol. 194, p. 108456, 2022.

[20] Q. Li, J. S. Gundersen, M. Lopuhaä-Zwakenberg, and R. Heusdens, "Adaptive differentially quantized subspace perturbation (adqsp): A unified framework for privacy-preserving distributed average consensus," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 1780–1793, 2024.

[21] X. Dong, Z. Wu, Q. Ling, and Z. Tian, "Byzantine-robust distributed online learning: Taming adversarial participants in an adversarial environment," *IEEE Transactions on Signal Processing*, vol. 72, pp. 235–248, 2024.

[22] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, vol. 108, 2020, pp. 2938–2948.

[23] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2018, p. 6106–6116.

[24] Y. Li, X. Wei, Y. Li, Z. Dong, and M. Shahidehpour, "Detection of false data injection attacks in smart grid: A secure federated deep learning approach," *IEEE Transactions on Smart Grid*, vol. PP, 2022.

[25] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, 2017.

[26] G. Damaskinos, E. M. El Mhamdi, R. Guerraoui, R. Patra, and M. Taziki, "Asynchronous Byzantine machine learning (the case of SGD)," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, 2018, pp. 1145–1154.

[27] O. Shalom, A. Leshem, and A. Scaglione, "Localization of data injection attacks on distributed m-estimation," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 8, pp. 655–669, 2022.

[28] T. W. Sherson, R. Heusdens, and W. B. Kleijn, "Derivation and analysis of the primal-dual method of multipliers based on monotone operator theory," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 2, pp. 334–347, 2019.

[29] M. Kefayati, M. S. Talebi, H. R. Rabiee and B. H. Khalaj, "Secure consensus averaging in sensor networks using random offsets," in *Proc. of the IEEE Int. Conf. on Telec., and Malaysia Int. Conf. on Commun., pp. 556–560*. IEEE, 2007.

[30] Z. Huang, S. Mitra, and G. Dullerud, "Differentially private iterative synchronous consensus," in *ACM workshop Privacy electron. Soc., pp. 81–90*, 2012.

[31] E. Nozari, P. Tallapragada, and J. Cortés, "Differentially private average consensus: Obstructions, trade-offs, and optimal algorithm design," *Automatica, vol. 81, pp. 221–231*, 2017.

[32] R. C. Hendriks, Z. Erkin, and T. Gerkmann, "Privacy preserving distributed beamforming based on homomorphic encryption," in *Proc. Eur. Signal Process. Conf., pp. 1-5*, 2013.

[33] Q. Li and M. G. Christensen, "A privacy-preserving asynchronous averaging algorithm based on shamir's secret sharing," in *Proc. Eur. Signal Process. Conf., pp. 1-5*, 2019.

[34] Q. Li, I. Cascudo, and M. G. Christensen, "Privacy-preserving distributed average consensus based on additive secret sharing," in *Proc. Eur. Signal Process. Conf., pp. 1-5*, 2019.

[35] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, "Rsa: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," *AAAI*, vol. 33, no. 1, pp. 1544–1551, 2019.

[36] J. Dall and M. Christensen, "Random geometric graphs," *Phys. Rev. E*, vol. 66, p. 016121, 2002.