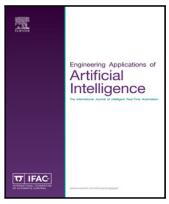
Engineering Applications of Artificial Intelligence 160 (2025) 111911

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Research paper

Secure Federated Dataset Distillation

Marco Arazzi [a], Mert Cihangiroglu [a], Serena Nicolazzo [b], Antonino Nocera [a,*]

[a] Department of Electrical, Computer and Biomedical Engineering, University of Pavia, A. Ferrata, 5, Pavia, 27100, PV, Italy
[b] Department of Science, Technology and Innovation, University of Eastern Piedmont, Viale Teresa Michel, 11, Alessandria, 15121, AL, Italy

ARTICLE INFO

Keywords:
Federated learning
Dataset Distillation
Federated distillation
Inference attack
Backdoor attack

ABSTRACT

Dataset Distillation (DD) is a powerful technique for reducing large datasets into compact, representative synthetic datasets, accelerating Machine Learning training. However, traditional DD methods operate in a centralized manner, which poses significant privacy threats and reduces its applicability. To mitigate these risks, we propose a Secure Federated Data Distillation (SFDD) framework to decentralize the distillation process while preserving privacy. Unlike existing Federated Distillation techniques that focus on training global models with distilled knowledge, our approach aims to produce a distilled dataset without exposing local contributions. We leverage the gradient-matching-based distillation method, adapting it for a distributed setting where clients contribute to the distillation process without sharing raw data. The central aggregator iteratively refines a synthetic dataset by integrating client-side updates while ensuring data confidentiality. To make our approach resilient to inference attacks perpetrated by the server that could exploit gradient updates to reconstruct private data, we create an optimized Local Differential Privacy approach, called LDPO-RLD (Label Differential Privacy Obfuscation via Randomized Linear Dispersion). Furthermore, we assess the framework's resilience against malicious clients executing backdoor attacks (such as Doorping) and demonstrate robustness under the assumption of a sufficient number of participating clients. Our experimental results demonstrate the effectiveness of SFDD and that the proposed defense concretely mitigates the identified vulnerabilities, with minimal impact on the performance of the distilled dataset. By addressing the interplay between privacy and federation in dataset distillation, this work advances the field of privacy-preserving Machine Learning making our SFDD framework a viable solution for sensitive data-sharing applications.

1. Introduction

Dataset Distillation (DD, hereafter) is defined as a set of approaches designed to generate compact yet highly informative data summaries to capture the essential knowledge of a given dataset. These distilled representations are optimized to function as efficient substitutes for the original dataset, enabling accurate and resource-efficient applications such as model training, inference, and architecture search (Wang et al., 2018). DD has attracted much attention from the deep learning community because it addresses the problem of handling unlimited data growth with limited computing power.

Typically, DD methods operate within a centralized and static framework, where the entire dataset is accessible at a single location (Wang et al., 2018; Cazenavette et al., 2022; Zhao et al., 2021). In particular, a straightforward approach to constructing a synthetic dataset implies that each data owner shares its data with a central server so DD can happen. Despite the numerous benefits, concentrating information in a single point of aggregation may lead to privacy leakages. Indeed, the central server may be honest but curious (or even malicious) and might take advantage of all the shared sensitive information. One specific application domain, in which this aspect may represent a critical issue is the sharing of medical datasets to establish the cross-hospital flow of medical information and improving the quality of medical services (for instance to construct high-accuracy computer-aided diagnosis systems) (Kumar et al., 2021; Weitzman et al., 2010). In this context, the different hospitals should share the medical information of all their patients with an external entity that is responsible for distilling the data before starting the specific analysis. In most cases, patients are reluctant to share such highly sensitive information, making DD approaches impractical (Aouedi et al., 2022).

Borrowing some ideas from the new paradigm of Federated Learning (FD, hereafter), which distributes the learning process across multiple entities to enhance privacy, our work proposes a novel framework for Secure and Federated Data Distillation (SFDD). This approach seeks to enable efficient DD while preserving data privacy by decentralizing the

* Corresponding author.
E-mail addresses: marco.arazzi01@universitadipavia.it (M. Arazzi), mert.cihangiroglu01@universitadipavia.it (M. Cihangiroglu), serena.nicolazzo@uniupo.it (S. Nicolazzo), antonino.nocera@unipv.it (A. Nocera).

https://doi.org/10.1016/j.engappai.2025.111911
Received 17 February 2025; Received in revised form 26 May 2025; Accepted 28 July 2025
Available online 6 August 2025

distillation process across multiple participants. This ensures that raw data remains local while only distilled knowledge is shared. Observe that our proposal takes a different perspective from the existing Federated Distillation (FD) schemes (Zhou et al., 2020; Cazenavette et al., 2022; Song et al., 2023). With the objective of improving FL performance and privacy guarantees, FD distills knowledge from multiple local models independently and transfers only compact, distilled information to the server that trains a global model using this distilled data. By contrast, the aim of our approach is to collaboratively distill data to produce a common distilled dataset without sharing local information. To do so, we start considering the method proposed by Zhao et al. (2021) for learning a synthetic set such that a deep network trained on it would preserve similar performance to that obtained when trained on the original large dataset. The synthetic data can later be used to train a network from scratch in a small fraction of the original computational load. We adopt this method and we make it distributed among different data owners (or clients). Firstly, a central server (or aggregator), that is in charge of aggregating the different contributions, produces and shares a random synthetic set of data. Then, the clients create local distillation contributions and at each step, they optimize the synthetic set of data and return to the server an enhanced version for aggregation till convergence is obtained.

However, in the pure FL context, some work (Zhu et al., 2019) has demonstrated that it is possible to obtain private training data from publicly shared gradients. Hence an honest but curious server may take advantage of the obtained updates sent by the clients to reconstruct the original batch of data and leak the privacy of data owners. For this reason, we first assess the vulnerability of our approach to this threat. We then enhance its security by implementing an improved Local Differential Privacy (LocalDP) strategy, called LDPO-RLD (LabelDP Obfuscation via Randomized Linear Dispersion). Adding this defense to our SFDD framework allows the clients to obfuscate the point-to-point correlation between distilled images and real ones. Experiments demonstrate that including LDPO-RLD in SFDD is not only an effective defense against deep leakage attacks but also outperforms the standard LocalDP in distillation performance.

Additionally, we test our solution also in the presence of malicious clients. Indeed, a recent work introduced a new backdoor attack method, called Doorping, which attacks during the dataset distillation process rather than after the model training (Liu et al., 2023). Our experimental campaign demonstrates that under the assumption of a sufficient number of clients, our framework is robust also to this new type of attack.

The results obtained during our experimental campaign demonstrate that our Federated Data Distillation approach is secure against known threats.

In summary, the key contributions of our framework are the following.

- We propose a strategy that allows diverse data owners to participate in a global and fully distributed Data Distillation process without sharing local data. DD is computed by merging all the contributions of the different clients, thus no raw data or detailed model parameters are exchanged, and privacy is preserved.
- Our framework is also secure against inference attacks. In fact, honest-but-curious servers cannot infer sensitive information about clients data by exploiting the gradient and update dynamics exchanged during the distillation process thanks to the presence of our LabelDP Obfuscation via Randomized Linear Dispersion (LDPO-RLD) defense strategy.
- We also demonstrate that, under the assumption of a sufficient number of clients, our framework is robust to client-side attacks, such as the Doorping attack.

To the best of our knowledge, this is the first proposal that enables multiple data owners to collaboratively participate in a global Dataset Distillation process without disclosing their raw data. The novelty of our SFDD framework also lies in its built-in resilience against honest-but-curious servers attempting to infer sensitive information from shared updates, as well as its robustness against client-side threats such as the Doorping attack.

The outline of this paper is as follows. In Section 2 we present a few practical application scenarios that could benefit from the proposed approach. Section 3, we examine the related papers present in the state-of-the-art. Section 4 is devoted to describing some basic concepts related to Federated Learning and Dataset Distillation useful for comprehending our proposal. Section 5 gives a general overview of our reference model and details the proposed framework. In Section 6, we present the experiments carried out to test our approach and show its performance. Finally, Section 7 examines intriguing leads as future work and draws our conclusions.

2. Application scenarios

In this section, we describe a possible use case scenario in healthcare that can benefit from our SFDD approach. In this context, the aim is to share datasets efficiently with other hospitals to train models effectively. Sharing medical datasets is challenging because of the privacy protection problem and the massive cost of transmitting and storing many high-resolution medical images. One possible solution is relying on DD to avoid transferring the entire dataset while still achieving similar model performance. Several studies demonstrate that DD can be a feasible method for efficient and secure medical data sharing, potentially facilitating enhanced collaborative research and clinical applications (Li et al., 2024, 2022b). However, distilled data might inadvertently memorize or reflect identifiable patient characteristics, leading to privacy concerns and compliance issues (Li et al., 2024). Our approach allows diverse hospitals to participate in a global and fully distributed DD process without sharing local data.

Another interesting application scenario is represented by emerging computational methodologies for novel material discovery and device simulation. In this context researchers often rely on distributed, high-dimensional datasets collected from different research laboratories and industrial partners (Zhang et al., 2025; Cao et al., 2025; Pan et al., 2025). In these contexts, preserving data confidentiality and intellectual property is a critical challenge. Our SFDD framework could serve as a secure foundation for collaborative training of AI models across multiple stakeholders without requiring raw data sharing, enabling: *(i)* cross-institutional collaboration on material simulations or experimental datasets; *(ii)* protection of proprietary datasets in device manufacturing; and *(iii)* use of synthetic datasets distilled securely from diverse real-world sources for training generative models or predictive systems in material science.

3. Related work

Dataset Distillation (DD, hereafter) (Wang et al., 2018) has recently emerged as a novel paradigm to synthesize a significantly smaller dataset from a large dataset, aiming to maintain the same training accuracy performance as if it was trained on the original large dataset. In this section, we describe existing proposals that combine Federated Learning (FL) with Distillation.

The work presented in Li and Wang (2019), Zhu et al. (2021), Jeong et al. (2018), Lin et al. (2020), Afonin and Karimireddy (2021) and Lu et al. (2024) leverages Knowledge Distillation (KD, for short) to transfer knowledge from local client models to a centralized FL server model to improve FL performance. Nowadays, KD-based FL is widely used to achieve collaborative learning among resource-limited devices (i.e., IoT) that are heterogeneous in data distribution, model architectures, or quantity of resources (Pang et al., 2024). In particular, Li and Wang (2019) introduces FedMD, a framework that allows participants to maintain private models while still benefiting from collaboration.

The authors of Zhu et al. (2021) propose a data-free KD approach to mitigate the issue of non-IID data distributions across clients, extracting knowledge without relying on external data. FD (Jeong et al., 2018) uses synchronized logit statistics to reduce communication costs, while Lin et al. (2020) focus on model fusion through ensemble distillation using unlabeled data. Afonin and Karimireddy (Afonin and Karimireddy, 2021) present a model-agnostic federated scheme based on kernel methods. The proposal of Lu et al. (2024) consists of a data-free knowledge filtering and distillation approach in FL called FedKFD. In FedKFD, each client learns a prediction capability description for its locally optimized model. All these works share two characteristics: they rely on Knowledge Distillation (rather than data-level synthesis) and aim to improve the performance or efficiency of FL training.

Only a limited body of work has explored Dataset Distillation in federated settings. In particular, Zhou et al. (2020), Song et al. (2023), Xiong et al. (2023), Cazenavette et al. (2022) and Hu et al. (2022) apply DD to reduce communication in FL. In these works, clients distill their local datasets into compact synthetic datasets and send them to the server. The goal is still to train a federated model while avoiding the transmission of full gradients or model weights. Zhou et al. (2020) propose a one-shot method where each client independently performs local distillation and the server aggregates the synthetic data. The proposal in Song et al. (2023) is called FedD3 and similarly to FedDM (Xiong et al., 2023) involves clients generating synthetic datasets (or matching gradient statistics) independently before sending them to the server. FedSynth (Hu et al., 2022) proposes a gradient compression approach that compresses gradients using synthetic data to reduce communication costs. In particular, instead of transmitting the model update, each client learns and transmits a lightweight synthetic dataset. These approaches use DD as a communication-saving mechanism inside a standard FL pipeline. While promising, these methods treat DD as a client-side pre-processing step and the distillation process remains local. Moreover, they compromise data privacy principles, as they require clients to upload synthetic data directly to the server.

Instead, the authors of Jia et al. (2024) propose FedDGM, an FL dataset distillation framework, allowing clients to train smaller models to mitigate computational costs, while the server aggregates this information to train a larger model. It relies on latent code inference and uses a pre-trained generator. Moreover, exclusively transferring model parameters rather than synthetic data, allows to ensure a level of privacy preservation. Similarly, FedCache (Pan et al., 2024) maintains a shared server-side synthetic data cache that clients can update asynchronously. However, also in this approach, the central goal remains training a global model, not creating a shared dataset.

In contrast with the above-cited approaches, our proposed method changes the role of dataset distillation in FL. Rather than using DD to support FL training, we federate the distillation process itself. That is, clients do not perform local distillation or send synthetic samples. Instead, they collaboratively optimize a shared synthetic dataset held at the server by contributing privatized gradient updates. The aim is not to improve model training efficiency but to generate a single, global distilled dataset in a distributed, privacy-preserving, and attack-resilient way.

Table 1 summarizes the state of the art related to our approach. We compare each study based on *(i)* the year of publication, *(ii)* the type of considered distillation, i.e., Knowledge (KD) or Dataset Distillation (DD); *(iii)* if the work proposes an approach based on Federated Learning; *(iv)* if the distillation is performed locally (client or server-side) or if it is distributed among clients; *(v)* if the proposed approach guarantees privacy and/or robustness to inference attacks; and *(vii)* the aim of the paper.

To the best of our knowledge, no prior work explicitly federates the distillation procedure. Existing methods distribute distilled data or training signals, but do not jointly optimize a global dataset under privacy and robustness constraints. SFDD is the first framework to align dataset distillation with the core principles of FL, decentralization, privacy, and robustness while treating the dataset itself as the final product rather than an auxiliary tool for model training.

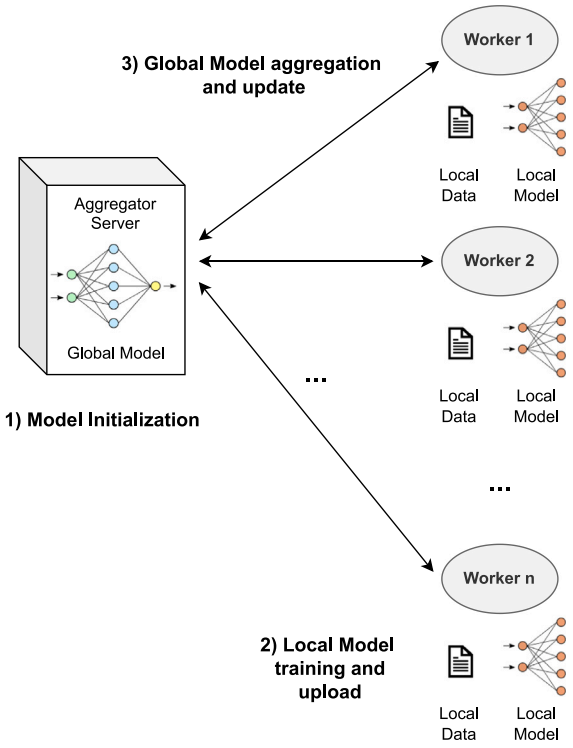


Fig. 1. FL workflow where a central server distributes a global model to multiple clients for local training. Clients return model updates, which are aggregated to refine the global model.

4. Background

In this section, we describe the main concepts that can be useful for a clear understanding of our approach. In particular, we focus on the description of the main concepts related to Federated Learning (FL) and Dataset Distillation (DD) mechanisms. Additionally, we illustrate the possible attacks in the context of FL.

Table 2 summarizes the acronyms used in this paper.

4.1. Federated learning

Federated Learning is designed to train an ML model in a decentralized manner across different devices holding local data samples. Keeping local data confidential without exchanging them with other participants or a central server allows privacy preservation; whereas sending only model updates reduces communication overhead and network traffic.

As visible in Fig. 1, the participants of this protocol are mainly of two types:

- the *worker* nodes, also called *clients*, that are C devices executing local training with their private data;
- an *aggregator* node, or *central server*, which is in charge of the coordination of the whole FL approach and aggregates the local updates.

Hence, the main goal of FL is to train a Global Model (GM), say $\mathbf{w}$, by uploading the weights of Local Models (LMs) $\{\mathbf{w}^i | i \in C\}$ to the central server. Eq. (1) shows the loss function to be optimized:

$$\min_{\mathbf{w}} l(\mathbf{w}) = \sum_{i=1}^{n} \frac{s_i}{C} L_i(\mathbf{w}^i) \tag{1}$$

where $L_i(\mathbf{w^i}) = \frac{1}{s_i} \sum_{j \in I_i} l_j(\mathbf{w}^i, x_i)$ is the loss function, s_i is the local data size of the ith worker, and I_i identifies the set of data indices with $|I_i| = s_i$, and x_j is a data point.

Table 1
Comparison with the state of the art approaches.

Ref.	Year	Type	FL	Distributed/Local distillation	Privacy preservation	Inference attack robustness	Aim
Jeong et al. (2018)	2018	KD	✓	Local, client-side	–	–	Improve FL performance
Li and Wang (2019)	2019	KD	✓	Local, client-side	–	–	Improve FL performance
Lin et al. (2020)	2020	KD	✓	Local, client-side	–	–	Improve FL performance
Zhou et al. (2020)	2020	DD	✓	Local, client-side	–	–	Reduce communication cost in FL
Afonin and Karimireddy (2021)	2021	KD	✓	Local, client-side	–	–	Improve FL performance
Zhu et al. (2021)	2021	KD	✓	Local, client-side	–	–	Improve FL performance
Cazenavette et al. (2022)	2022	DD	✓	Local, client-side	–	–	Reduce communication cost in FL
Hu et al. (2022)	2022	DD	✓	Local, client-side	–	–	Reduce communication cost in FL
Song et al. (2023)	2023	DD	✓	Local, client-side	–	–	Reduce communication cost in FL
Xiong et al. (2023)	2023	DD	✓	Local, client-side	–	–	Reduce communication cost in FL
Jia et al. (2024)	2024	DD	✓	Server-side	✓	–	Efficient and privacy-enhanced FL
Lu et al. (2024)	2024	KD	✓	Local, client-side	–	–	Improve FL performance
Pan et al. (2024)	2024	DD	✓	Server-side	✓	–	Efficient and privacy-enhanced FL
Our	2025	DD	✓	distributed	✓	✓	Generate a global distilled dataset in a distributed, privacy-preserving, and attack-resilient way

Table 2
Summary of the acronyms used in the paper.

Symbol	Description
DD	Dataset Distillation
DL	Deep Learning
FD	Federated Distillation
FL	Federated Learning
GM	Global Model
IID	Independent and Identically Distributed
KD	Knowledge Distillation
LDPO-RLD	LabelDP Obfuscation via Randomized Linear Dispersion
LocalDP	Local Differential Privacy
LM	Local Model
MD	Model Distillation
ML	Machine Learning
IPC	Image Per Class
SFDD	Secure Federated Data Distillation

The basic FL workflow can be divided into three main phases (Zhang et al., 2021). During the first stage, called *Model initialization*, the server *(i)* initializes the necessary parameters for the GM $\mathbf{w}$; and *(ii)* select the workers for the FL process. The second phase consists of the *LMs training and uploading*. The clients download the current GM and perform local training on their private data during this stage. After that, each client computes the model parameter updates and sends them to the server. The regional training involves more than one iteration of back-propagation, gradient descent, or other optimization methods to improve the LM's performance. In particular, for each iteration, the different clients update the GM with their datasets: $\mathbf{w}_t^i \leftarrow \mathbf{w}_t^i - \eta \frac{\partial L(\mathbf{w}_t, b)}{\partial \mathbf{w}_t^i}$, where η specifies the learning rate and b is the local batch. Finally, the *GM aggregation and update* phase is performed. In this step, the server collects and aggregates the model parameter updates from all the workers, $\{\mathbf{w}^i | i \in C\}$. The aggregator can employ various methods like averaging, weighted averaging, or secure multi-party computation (SMC) to incorporate the received updates from each client.

As visible in Fig. 2, FL can assume the following three configurations according to the different data partition strategies considered (Yang et al., 2019):

- **Vertical Federated Learning** (VFL) in the case in which the datasets share overlapping data samples but differ in the feature space (see Fig. 2(a)). This scheme can be applied if two different organizations (i.e., an Internet service provider and an online TV streaming provider) have data about the same group of people with different features and want to collaboratively train an ML model while keeping their data private.
- **Horizontal Federated Learning** (HFL) that is used for cases in which each device contains a dataset with the same feature space but with different sample instances. For instance, think of two branches of the same insurance company that hold the same type of data about different clients (see Fig. 2(b)).
- **Federated Transfer Learning** (FTL) borrows some characteristics from both VFL and HFL and is suitable for scenarios in which there is little overlapping in both data samples and features as visible in Fig. 2(c). A good example is the case in which a bank wants to train its ML model by cooperating with an insurance company that shares part of the client and part of the features.

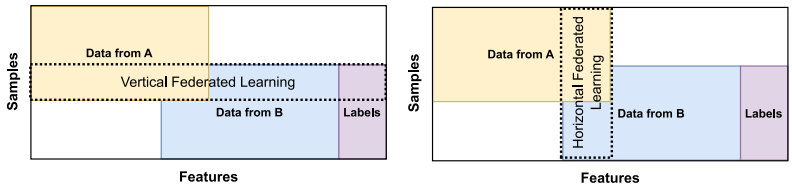
Even if FL has been designed to achieve data confidentiality it has been demonstrated that it is still prone to possible attacks targeting data privacy that any participants of the scheme can perpetrate (Lyu et al., 2022). The most common attacks in this context are the following:

- **Inference attacks** that aim at inferring the sensitive information about individual data points (attribute inference) or participants (membership inference) in the training dataset by analyzing the behavior or outputs of the federated model (Nasr et al., 2019; Arazzi et al., 2025b, 2023a).
- **Poisoning attacks** can be divided into data or model poisoning attacks. The first category involves adversaries that try to poison the training data in a certain number of devices participating in the learning process to compromise the GM accuracy. The adversary can inject poisoned data *(i)* directly into the targeted device or *(ii)* through other devices (Sun et al., 2021; Arazzi et al., 2023b). In a model poisoning attack the adversary tries to poison the LMs instead of the local data to introduce errors in the GM.
- **Backdoor attacks** through which an adversary can mislabel certain tasks without affecting the accuracy of the GM. This kind of attack manipulates a subset of training data by injecting adversarial triggers such that the models trained on the tampered dataset will make arbitrarily (targeted) incorrect predictions on the test set with the same trigger embedded (Gu et al., 2019; Arazzi et al., 2024).

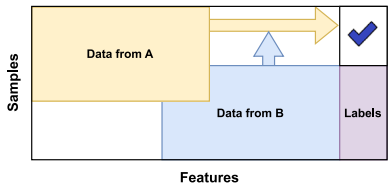
4.2. Dataset distillation

In general, in the context of ML, Distillation (known as Model Distillation) is a methodology to transfer knowledge from a larger, more complex model (called "teacher") to a smaller, simpler model (known as "student") to improve model performance or deploy the model on resource-constrained devices, such as Internet of Things (IoT) devices.

An alternative concept proposed by Wang et al. (2018) is called Dataset Distillation (DD) and consists of the summarization of real data in a few highly informative and synthetic data points in such a way that models trained on the last dataset achieve comparable



(a) Vertical Federated Learning (VFL) (b) Horizontal Federated Learning (HFL)



(c) Federated Transfer Learning (FTL)

Fig. 2. The three categories of Federated Learning based on feature and sample spaces.

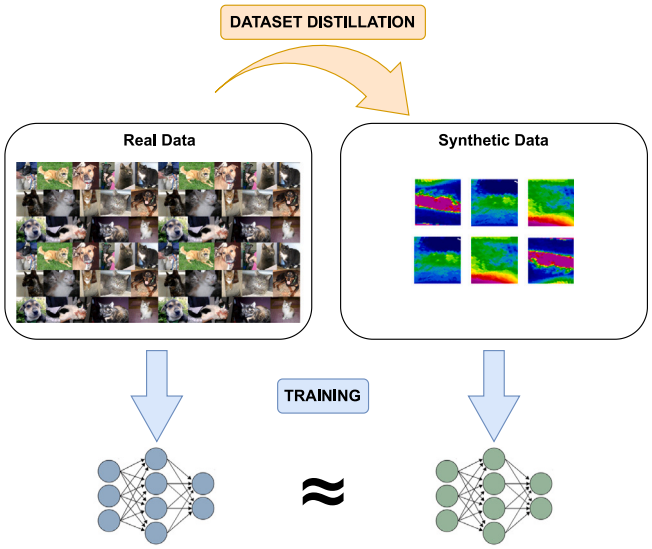


Fig. 3. The general workflow of Dataset Distillation (DD), where a synthetic dataset is iteratively optimized to capture the knowledge of a larger real dataset.

generalization performance to those trained on the real data. has been created to address this problem of large data volume (Lei and Tao, 2023). Fig. 3 illustrates the Dataset Distillation scheme, showing that the models trained on the large original dataset and small synthetic dataset demonstrate comparable performance on the test set.

To formally define DD, we start with some preliminary definitions, namely a target dataset:

$$\mathcal{T} = \{(x_i, y_i)\}_{i=1}^{m}$$

where $x_i \in \mathbb{R}^d$, d is the dimension of the input data, y_i is the ith label, and (x_i, y_i), with $1 \leq i \leq m$ are independent and identically distributed (i.i.d.) random variables drawn from the data generating distribution $\mathcal{D}$. The goal of DD is to extract the knowledge of $\mathcal{T}$ into a small synthetic dataset called:

$$\mathcal{S} = \{(s_j, y_j)\}_{j=1}^{n}$$

where $n \ll m$ and the model trained on the small distilled dataset $\mathcal{S}$ can achieve a generalization performance that can be approximated to the one of the original dataset $\mathcal{T}$:

$$\mathbb{E}_{\substack{(x,y)\sim D \\ \Theta^{(0)}\sim P}}[\ell(f_{alg(\mathcal{T})}(x), y)] \simeq \mathbb{E}_{\substack{(x,y)\sim D \\ \Theta^{(0)}\sim P}}[\ell(f_{alg(\mathcal{S})}(x), y)]$$

Table 3
Summary of the symbols used in our approach.

Symbol	Description
C	Central unit
$\mathcal{S}^g$	Global synthetic set of data
$\mathcal{W}$	Set of *Workers*
w_k	kth worker
$\mathcal{M}^{w_k}$	private model of w_k
$\mathcal{M}^{\mathcal{W}}$	Set of all the $\mathcal{M}^{w_k}$
$\mathcal{T}^{w_k}$	Private dataset of w_k
$\mathcal{T}^{\mathcal{W}}$	Set of all $\mathcal{T}^{w_k}$
$CE(\cdot)$	Cross-entropy loss
$\nabla(\cdot)$	Gradient
ML	Matching loss

where

- $\Theta^{(0)}$ is the initialized network parameter;
- $f_{alg(\mathcal{T})}$ is the model f trained on the original dataset $\mathcal{T}$;
- $f_{alg(\mathcal{S})}$ is the model f trained on the synthetic dataset $\mathcal{S}$;
- $f_{alg(\bullet)}(x)$ is the prediction or output of $f_{alg(\bullet)}$ at x;
- $\ell(f_{alg(\bullet)}(x), y)$ is the loss between the prediction $f_{alg(\bullet)}(x)$ and ground truth y;
- $\mathbb{E}_{\substack{(x,y)\sim D \\ \Theta^{(0)}\sim P}}[\ell(f_{alg(\bullet)}(x), y)]$ is the empirical risk and refers to the average loss or error of a model on a training dataset.

5. Description of our approach

This section outlines our proposed method in detail. Section 5.1 introduces the overall approach and its foundational model, including formal definitions of *(i)* the participating entities and *(ii)* the Secure Federated Data Distillation (SFDD) process. In Section 5.2, we examine a potential vulnerability of our framework to inference attacks and present the enhancements implemented to strengthen its resilience against such threats. In this section, we provide a detailed description of our proposal. Table 3 illustrates the symbols used in the description of our approach.

5.1. General overview

This section describes our SFDD architecture for decentralized and secure Data Distillation. We start by defining the fundamental components of our strategies, in particular, as visible in Fig. 4, the involved parties are:

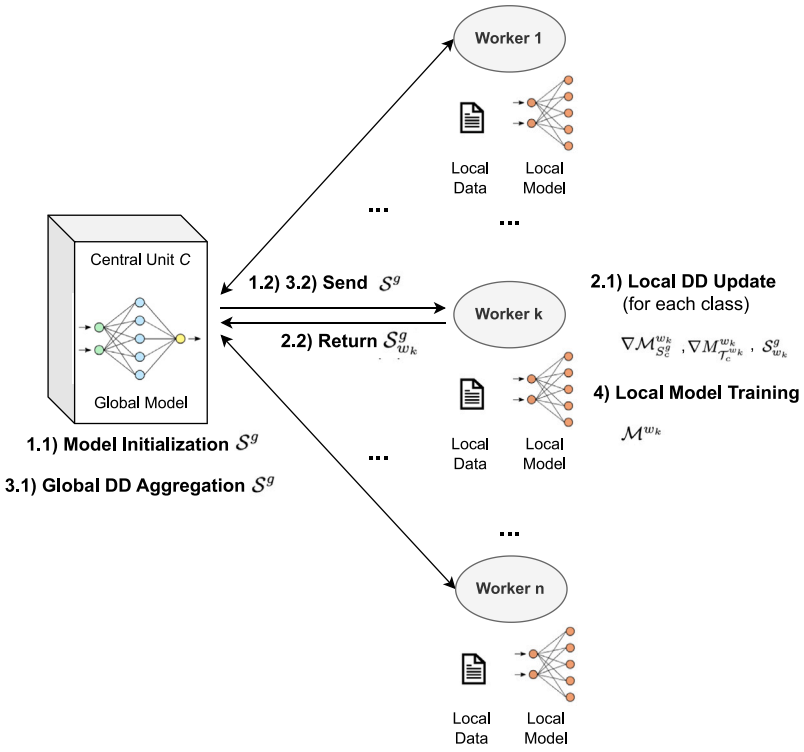


Fig. 4. Secure Federated Data Distillation (SFDD) architecture and main actors of the *starting phase*.

- a *Central Unit C* that acts as an aggregator node and is in charge of initializing the global synthetic set of data $\mathcal{S}^g$, randomly sampled from a Gaussian distribution $\mathcal{N} = (0, 1)$;
- a set of *Workers* or clients $\mathcal{W} = \{w_1, \ldots, w_z\}$ of size z, which execute the Federated Distillation algorithm.

In our configuration, each worker w_i in $\mathcal{W}$ holds a private data set $\mathcal{T}^{w_i} = \{(x_j^z, y_j^z)\}_{j=1}^m$ to be distilled. All the private datasets of the network are independent and identically distributed (IID) and do not overlap. The set composed of all the $\mathcal{T}^{w_i}$ for each worker w_i can be formally defined as follows:

$$\mathcal{T}^{\mathcal{W}} = \{\mathcal{T}^{w_1}, \ldots, \mathcal{T}^{w_z}\};\ \mathcal{T}^{w_i} = \{(x_j^z, y_j^z)\}_{j=1}^m$$

where m is the number of data points for a set.

Our adopted technique relies on gradient matching and follows the main step described in Zhao et al. (2021). According to this strategy, each worker $w_k \in \mathcal{W}$ holds a private model $\mathcal{M}^{w_k}$, randomly initialized, used to transfer knowledge from its private set of data $\mathcal{T}^{w_k}$ to the global synthetic set $\mathcal{S}^g$. As demonstrated in the initial centralized model in Zhao et al. (2021), the distillation method is adaptable to various architectures, ensuring the proposed solution's independence from the architecture selection. The set of private models can be referred to as:

$$\mathcal{M}^{\mathcal{W}} = \{\mathcal{M}^{w_1}, \ldots, \mathcal{M}^{w_z}\}$$

In the following, we detail all the phases of our Secure Federated Data Distillation approach. In particular, we describe all the steps related to the *starting phase* and the *operative phase*.

As for the *starting phase* the steps are:

1. Data Initialization;
2. Local Data Distillation Update;
3. Global Data Distillation Aggregation;
4. Local Model Training.

Fig. 4 shows the steps performed by our framework during the *starting phase* and the involved actors.

Data Initialization. The first step of the *starting phase* of our framework, called *Data Initialization* (see step 1.1 in Fig. 4), is performed by the central unit C that starts the process by randomly initializing the synthetic set of data $\mathcal{S}^g$. Specifically, it initializes a given number ipc of data (representing the number of images per class) for each class of the original dataset as follows:

$$\mathcal{S}^g = \{\mathcal{S}_0^g, \ldots, \mathcal{S}_c^g, \ldots, \mathcal{S}_{n_c}^g\};\ \mathcal{S}_c^g = \{(s_{i_c}, y_c)\}_{i_c=1}^{ipc}$$

where y_c is the label assigned to class c and n_c is the total number of classes (see step 1.2 in Fig. 4). The synthetic dataset $\mathcal{S}^g$ is distributed to all the workers in $\mathcal{W}$.

Local Data Distillation Update. Once each worker receives $\mathcal{S}^g$, the second step of *Local Data Distillation Update* takes place and the local distillation phase of the process carried out by the workers starts in a parallel and independent way. In particular, each worker w_k distills the data contained in its private set $\mathcal{T}^{w_k}$ using its private model $\mathcal{M}^{w_k}$ (see step 2.1 in Fig. 4).

The distillation process is conducted separately for each class of the dataset and it is performed generating two contributions, namely: *(i)* $\nabla\mathcal{M}^{w_k}_{S_c^g}$, the gradients of the local model $\mathcal{M}^{w_k}$ associated to the synthetic data S_c^g (see Eq. (2)), and *(ii)* $\nabla\mathcal{M}^{w_k}_{\mathcal{T}_c^{w_k}}$, the gradients of the local model $\mathcal{M}^{w_k}$ associated to a batch of real data (see Eq. (3)).

$$\nabla\mathcal{M}^{w_k}_{S_c^g} \leftarrow CE(\mathcal{M}^{w_k}(S_c^g), y_c); \tag{2}$$

$$\nabla\mathcal{M}^{w_k}_{\mathcal{T}_c^{w_k}} \leftarrow CE(\mathcal{M}^{w_k}(\mathcal{T}_c^{w_k}), y_c) \tag{3}$$

In the above equations, $CE(\cdot)$ is the cross-entropy loss used to compute the gradients on $\mathcal{M}^{w_k}$, and $\mathcal{T}_c^{w_k}$ is a subset of $\mathcal{T}^{w_k}$ in which $y_i^z = y_c$.

The *Local Data Distillation Update* outcome, for each worker w_k is a total loss $\mathcal{S}^g_{w_k}$ that is back-propagated onto the global synthetic dataset $\mathcal{S}^g$. The idea is to update the synthetic images to make them generate partial gradients similar to the ones generated by a batch of real data. To do so, we compute a matching loss ML between the two partial gradients (namely, $\nabla\mathcal{M}^{w_k}_{S_c^g}$ and $\nabla\mathcal{M}^{w_k}_{\mathcal{T}_c^{w_k}}$) to measure their distance for

each iteration of the process. In particular, we measure this distance through the L_2 norm $\|\cdot\|_2$, so $ML = \|\cdot\|_2$

More formally, this step can be formulated as follows:

$$S^g_{w_k} = S^g \leftarrow \sum_{c=1}^{n_c} ML(\nabla\mathcal{M}^{w_k}_{S^g_c}, \nabla\mathcal{M}^{w_k}_{\mathcal{T}_c^{w_k}}). \tag{4}$$

where $S^g_{w_k}$ is the locally updated version of S^g obtained by the worker w_k. This contribution is then sent to the Central unit for the aggregation (see step 2.2 in Fig. 4)

Global Data Distillation Aggregation. The next step of the framework is the *Global Data Distillation Aggregation*, in which the obtained updates on $S^g_{w_k}$ are sent back to the central unit C to be aggregated (see step 3.1 in Fig. 4). The employed method is FedAvg strategy (McMahan et al., 2017) in which the Central Unit performs a weighted average of the clients' updates to produce a new global model. Finally, the Central Unit sends back to the workers the new global S^g (see step 3.2 in Fig. 4).

It is worth observing that, as is typically done in the related literature, we have adopted a standard aggregation strategy based on FedAVG. This method has been shown to be effective in disparate application domains, even in the presence of a low number of clients. In fact, in our experiments in Section 6.2, we demonstrate the high quality of the aggregation even with a limited number of clients (that is, 5 clients). In addition, the recent literature has also analyzed FedAVG performance in a non-IID setting. In the last case, although FedAVG was not originally designed for situations in which clients have different data distributions, it has proven to be still partially effective and competitive with respect to specifically tailored approaches (Li et al., 2022a; Arazzi et al., 2025a), such as FedProx (Li et al., 2020), FedNova (Wang et al., 2020), and SCAFFOLD (Karimireddy et al., 2020). Without loss of generality, in such scenarios, to increase the obtained performance, more robust aggregation schemes, like the ones mentioned above, can still be adopted.

Local Model Training. The last step of this starting phase includes that all the $S^g_{w_k}$ data are then used in the *Local Models* $\mathcal{M}^{w_k}$ *Update* process before the next iteration of the Federated Distillation (see step 4 in Fig. 4).

Fig. 5 shows the details of the *starting phase* of our framework.

As for the *operative phase*, the steps are the following:

1. Local Data Distillation Update;
2. Global Data Distillation Aggregation;
3. Local Model Training.

These steps are repeated till the model's accuracy on the distilled dataset converges. Fig. 6 shows the steps performed by our framework during the *starting phase* and the involved actors.

The workers perform the first step of *Local DD Update* to compute the $S^g_{w_k}$ contribution (see step 1.1 of Fig. 6) and then send it back to the Central Unit C (step 1.2 of Fig. 6). At this point, C performs the DD aggregation and returns the result S^g to the different workers (steps 2.1 and 2.2 of Fig. 6). Finally, the workers execute the local model training (see step 3 of Fig. 6).

5.2. Defence against data leakage attack

In this section, we analyze a potential vulnerability of our scheme to inference attacks and present the modifications introduced to enhance its robustness against this specific threat. Like traditional Federated Learning strategies, as tested experimentally in Section 6, our approach is vulnerable to data leak attacks perpetrated by an honest but curious server (Zhu et al., 2019). This attack is carried out to obtain the private training data from the publicly shared gradients. To perform it, a server takes advantage of the updates returned by the different workers to reverse the distillation process and obtain the original batch of data $\mathcal{T}_c^{w_k}$ used in the current epoch by the worker w_k.

In particular, in this context, the server can randomly initialize a tensor $\mathcal{LB}$ of the same shape as the one used by the workers to distill the knowledge ($batch - size\ X\ n_c$). The tensor is defined as follows:

$$\mathcal{LB}^{w_k} = \{\mathcal{LB}_0^{w_k}, \ldots, \mathcal{LB}_c^{w_k}, \ldots, \mathcal{LB}_{n_c}^{w_k}\}$$

where $\mathcal{LB}_c$ is the leaked batch used by w_k to distill the synthetic data S^g_c for the corresponding class c. To reverse the process, the server tries to replicate the distillation strategy by freezing the S^g at the previous step and calculating the updates $\nabla S^{g'}_{w_k}$ emulating the workers but using $\mathcal{LB}_c^{w_k}$ instead of $\mathcal{T}_c^{w_k}$ for all the classes. In this way, the server tries to replicate and match the updates $\nabla S^g_{w_k}$ on S^g returned by the worker w_k. The process can be formalized as follows:

$$\nabla\mathcal{SM}^{w_k}_{S^g_c} \leftarrow CE(\mathcal{SM}^{w_k}(S^g_c), y_c); \nabla\mathcal{M}^{w_k}_{\mathcal{LB}_c^{w_k}} \leftarrow \tag{5}$$

$$CE(\mathcal{SM}^{w_k}(\mathcal{LB}_c^{w_k}), y_c) \tag{6}$$

$$l = \sum_{c=1}^{n_c} ML(\nabla\mathcal{M}^{w_k}_{S^g_c}, \nabla\mathcal{M}^{w_k}_{\mathcal{LB}_c^{w_k}});\ \nabla S^{g'}_{w_k} = \frac{\partial l}{\partial S^g} \tag{7}$$

$$\mathcal{LB}^{w_k} \leftarrow \|\nabla S^g_{w_k} - \nabla S^{g'}_{w_k}\|_2 \tag{8}$$

where the matching between $\nabla S^g_{w_k}$ and $\nabla S^{g'}_{w_k}$ is performed using the L_2 norm $\|\cdot\|_2$.

To guarantee the preservation of the privacy of the workers from an honest but curious server adopting the method above, we include in our framework an enhanced Local Differential Privacy (LocalDP) strategy. Traditional LocalDP can fit the considered scenario by adding *Gaussian* or *Laplacian* noise and clipping to the obtained updates of S^g. For our approach to be effective and, at the same time, to preserve the performance of the distilled data we have to estimate the amount of both *(i)* the noise and the *(ii)* clipping level. To achieve this, a worker must perform a grid search over various noise and clipping parameters while simultaneously evaluating performance retention and testing the attack effectiveness under these conditions. However, the computational overhead introduced by this exhaustive search could significantly slow down the distillation process.

Algorithm 1 LabelDP Obfuscation via Randomized Linear Dispersion

Require:
y_c: label of the current distilled class
k: number of obfuscation classes
ϵ: smoothing parameter
n: number of classes
$RandomIndexes = Rand(k, n)$
$SL \leftarrow zeros(n)$
for i in $RandomIndexes$ **do**
  if $HL[i] == y_c$ **then**
    $SL[i] \leftarrow 1 - \epsilon$
  else
    $SL[i] \leftarrow \epsilon/(k-1)$
  end if
end for

To overcome this limitation, we propose an optimized strategy based on a community-oriented Label Differential Privacy (LabelDP) method, inspired by Arazzi et al. (2025b). Our approach, in Algorithm 1, called LabelDP Obfuscation via Randomized Linear Dispersion (LDPO-RLD, hereafter), enables workers to obscure the correlation between distilled and real images. Instead of using one-hot encoded labels to compute gradient matching, workers redistribute an ϵ fraction of the primary label's probability across a set of k randomly selected labels using a linear function, $Lin(\cdot)$. To generate noisy labels, the authors of Arazzi et al. (2025b) propose employing Knowledge Distillation through a pre-trained teacher network. However, the use of a teacher network would add unnecessary overhead, particularly in resource-constrained environments. In our scenario, such an additional complexity is not required, as we use the noisy-label approach to just

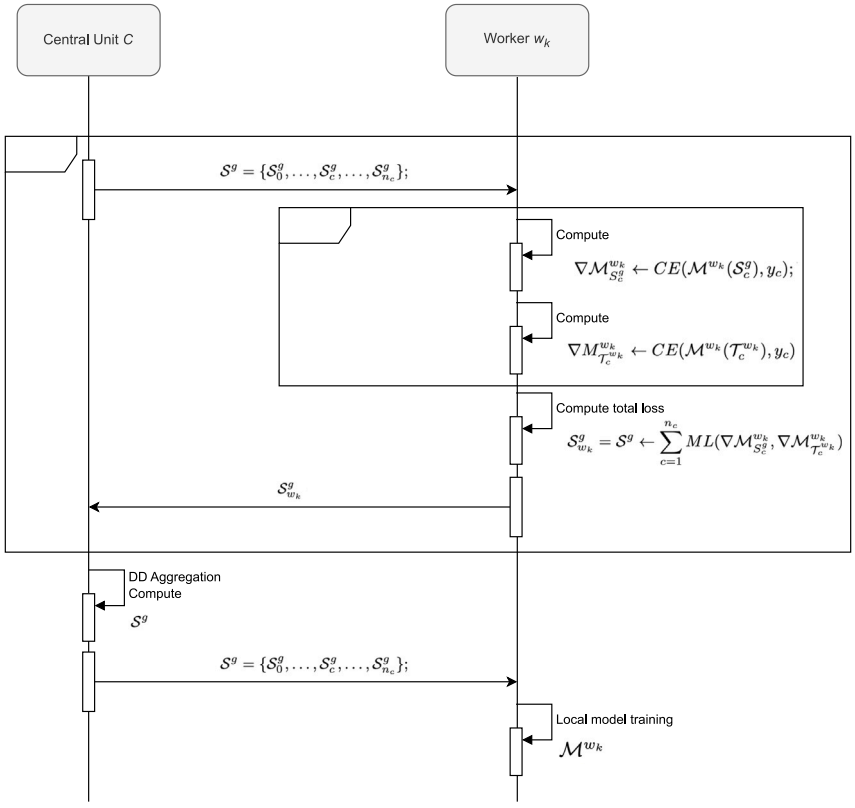


Fig. 5. Secure Federated Data Distillation (SFDD) sequence diagram.

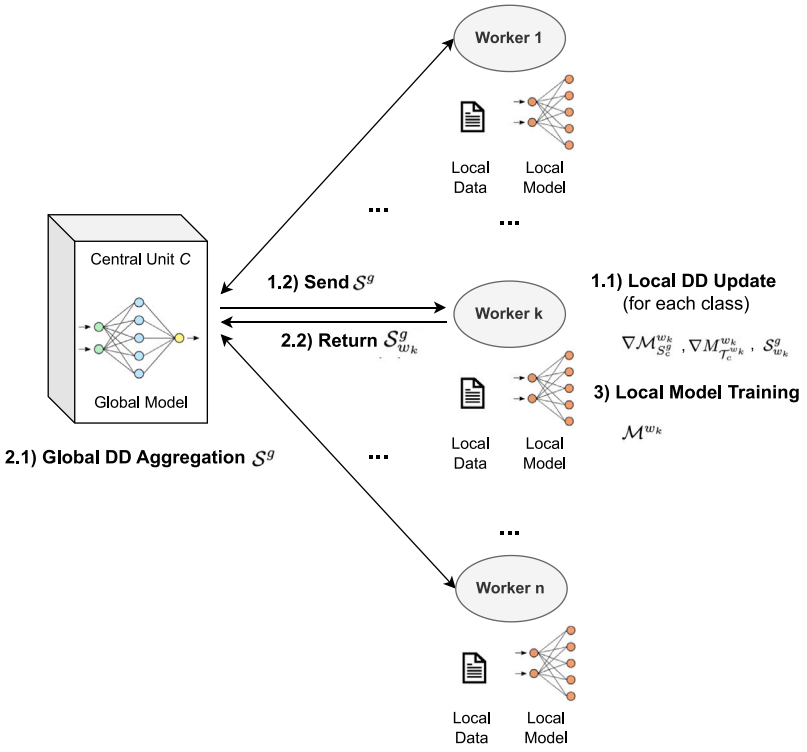


Fig. 6. Secure Federated Data Distillation (SFDD) architecture and main actors of the *operative phase*.

obfuscate the correlation between real and distilled images. Therefore, we can select the k labels at random. This modification reduces the complexity while maintaining the intended obfuscation level. The resulting changes in the loss and gradient calculations are as follows:

$$\nabla \mathcal{SM}^{w_k}_{S^g_c} \leftarrow CE(\mathcal{SM}^{w_k}(S^g_c), Lin(y_c, k, \epsilon)); \tag{9}$$

$$\nabla \mathcal{M}^{w_k}_{\mathcal{LB}^{w_k}_c} \leftarrow CE(\mathcal{SM}^{w_k}(\mathcal{LB}^{w_k}_c), Lin(y_c, k, \epsilon)) \tag{10}$$

With this new formulation, an attacker attempting to reverse the distillation process to recover the original batch of images cannot accurately reconstruct the worker's scenario, as the specific label distribution used remains unknown. This strategy is not only effective but also specifically designed for the federated context, minimizing additional overhead. Instead of creating obfuscated images via Knowledge Distillation with a pre-trained teacher network as in Arazzi et al. (2025b), this modification significantly cuts down the computational resources needed to generate obfuscated labels by merely selecting k random additional labels aside from the primary one making this applicable in a context that requires the minimal latency like Industry 5.0 scenario.

In the following section, we present the experimental results, assessing the effectiveness of the modified loss function against the reference Data Leakage attack, as well as the final performance of the distilled dataset on the test set.

6. Experiments and results

In this section, we discuss the experiments carried out to assess the performance of our framework. Specifically, in Section 6.1, we describe the datasets and the metrics used for our experimental campaign. Section 6.2 is dedicated to evaluating the performance of our solution, and finally, in Section 6.3, we show the performance of our defense strategy (LDPO-RLD) against both server and client-side attacks.

6.1. Experimental setup

In our experiments, we utilized five state-of-the-art datasets, selected for their diversity in image content and classification complexity, allowing for a comprehensive evaluation of our SFDD approach. Below, we provide details on each of the five datasets used:

- **MNIST** (Deng, 2012) consists of 70,000 grayscale images of handwritten digits (0–9), with 60,000 images designated for training and 10,000 for testing. Each image is 28 × 28 pixels in size.
- **CIFAR-10** (Krizhevsky et al., 2009) contains 60,000 32 × 32 color images categorized into 10 classes, with 50,000 images for training and 10,000 images for testing.
- **SVHN (Street View House Numbers)** (Netzer et al., 2011) is a real-world image dataset designed for digit recognition in natural scene images. The dataset is provided by Stanford University and is divided into 73,257 training and 26,032 testing images. Additionally, there are 531,131 extra images, which are considered somewhat less challenging by the dataset provider; however, these additional images were not used in our experiments.
- **GTSRB (German Traffic Sign Recognition Benchmark)** (Stallkamp et al., 2012) contains 39,270 images of traffic signs categorized into 43 classes. The dataset is divided into 26,640 training images and 12,630 testing images.
- **CIFAR-100** (Krizhevsky et al., 2009) is similar to CIFAR-10, CIFAR-100 contains 60,000 32 × 32 color images, but with 100 classes instead of 10. Each class has 600 images, divided into 500 training images and 100 testing images per class.

All the datasets have been employed using their original size without any transformation. The following metrics are employed to quantify system performance:

- **Accuracy** is defined as the proportion of correctly predicted instances out of the total number of instances:

  $$Accuracy = \frac{TP}{TP + FP}$$

  where TP (True Positives) refers to the number of instances where the model correctly predicts the positive class, whereas FP (False Positives) refers to the number of times that the model incorrectly predicts a positive class.
- **Mean Squared Error (MSE)** measures the average squared difference between the estimated values and the true value:

  $$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

  where n is the number of data points, y_i is the actual (true) value, $\hat{y}_i$ is the predicted (estimated) value, and the squared difference $(y_i - \hat{y}_i)^2$ measures the error for each prediction.
- **Attack Success Rate (ASR)** refers to the percentage of times an attack achieves its intended goal or outcome. In the context of backdoor attacks like the Doorping attack, the success rate would be the proportion of instances in which the attack successfully manipulates the model or dataset as intended by the attacker.

6.2. Performance analysis

In this section, we analyze the results of the experiments designed to evaluate the performance of our approach. As outlined in Section 4.2, the distillation process requires two models, one for distillation and the second for the performance evaluation of the distilled images. In our performance analysis, we employ the same CONVNET architecture described in Zhao et al. (2021) for both models. This architecture features multiple convolutional layers, each followed by normalization, activation, and pooling layers. The design of the CONVNET allows for extensive customization, including the number of filters (default `net_width=128`), network depth (default `net_depth=3`), activation functions (default `net_act=ReLu`), normalization techniques (default `net_norm=instancenorm`), and pooling strategies (default `net_pooling=avgpooling`). The optimization of both networks and synthetic images is performed using the Stochastic Gradient Descent algorithm with a learning rate of 0.1 for the images and 0.01 for the networks, respecting the parameters used by the original centralized approach. As can be seen, the proposed algorithm can be performed even using basic network architectures, executable even by basic devices, fitting perfectly with the scenario of distributed learning. The input images are standardized to a size of 32 × 32 pixels. We maintained consistent network settings for both the distillation and evaluation models across the centralized version and the federated clients.

In the first experiment, we compared the performance in terms of the accuracy result of the standard DD solution against our SFDD approach with 5 clients as a baseline. To fulfill the assessment, we evaluated both solutions by changing the setting for the produced Images Per Class (IPC). In particular, for each considered dataset, we collected the results producing 1, 10, and 50 images for each class, as it has been done in the original paper that presented the centralized version. It is important to stress that this paper does not introduce a new distillation technique, but rather designs a novel federated approach to adapt existing solutions in a distributed context. In our case, we adopt the approach proposed in Zhao et al. (2021). Then, our proposal focuses on the security threats introduced by the distributed nature of our solution and proposes the inclusion of a suitable defense mechanism.

As shown in Table 4, our SFDD approach consistently achieved comparable performance across the five different datasets. For the MNIST dataset, our approach slightly surpasses the centralized approach with 1 and 10 IPC and remains highly competitive with 50 IPC. In the CIFAR10 dataset, SFDD performs slightly better with 1 IPC and maintains very

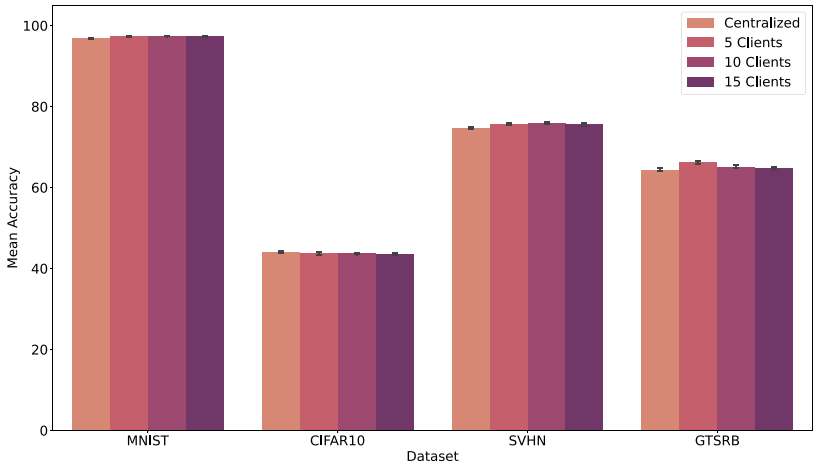


Fig. 7. SFDD performance with different numbers of clients.

Table 4
Comparison of the performance of the Centralized DD approach and SFDD.

Dataset	IPC	Centralized (Zhao et al., 2021)	SFDD
MNIST	1	91.92%	92.02%
	10	97.49%	97.58%
	50	98.30%	98.72%
CIFAR10	1	27.82%	28.10%
	10	44.29%	43.65%
	50	53.14%	53.40%
SVHN	1	30.58%	29.10%
	10	75.19%	75.89%
	50	81.70%	81.58%
GTSRB	1	31.94%	32.13%
	10	66.55%	65.38%
	50	75.64%	71.08%
CIFAR100	1	12.41%	12.57%
	10	24.82%	24.68%
	50	29.73%	29.43%

close performance with 10 and 50 IPC. The SVHN dataset results show that SFDD is nearly as effective as the centralized method with 1 IPC and slightly outperforms it with 10 IPC. For the GTSRB dataset, our method exhibits superior performance with 1 IPC and shows strong results with 50 IPC. In the CIFAR100 dataset, the SFDD performs admirably with 1 IPC and is nearly identical in performance with 10 IPC. Overall, our SFDD approach proves to be a robust and effective method, consistently delivering performance on par with the state-of-the-art centralized method highlighting the minor variations introduced by the distributed scenario, which adds more dynamism to the parameters initialized differently by each client.

The second experiment aims to assess whether the number of clients participating in SFDD impacts the quality of the generated images. To evaluate this, we conducted a performance analysis using 10 and 15 clients, alongside the default setting of 5 clients. We measured the final mean accuracy and standard error on test sets across multiple randomly initialized networks trained with the distilled images. For consistency, we set the number of images per class to 10 as the baseline for this experiment.

Fig. 7 presents the results for this second experiment. On the basis of this diagram, we can conclude that increasing the number of clients does not significantly impact distillation performance. In MNIST and SHVN, an increase in the number of clients results in slightly better distillation performance. In CIFAR-10 and GTSRB, it results in a slight decrease. Overall, the number of clients does not affect the results, as the differences are negligible, also in this case the detected oscillation might result from the randomicity introduced by the different initialization of the local models in each client, which produce insignificant minor variations. This finding allows us to guarantee that multiple clients can participate in our framework without affecting the quality of the final results.

6.3. Security analysis

As for the experiments dealing with the security analysis, we employed the same CONVNET architecture from Zhao et al. (2021) for both distillation and evaluation models, but with certain modifications to simplify the architecture and put the attacker in the best possible scenario for them. Specifically, we discarded the pooling layers and substituted the ReLU activation function with Sigmoid. These modifications are consistent with the changes made in the original paper presenting the Deep Leakage attack (Zhu et al., 2019). The rationale behind replacing ReLU with Sigmoid is that Sigmoid facilitates better gradient flow, which is advantageous for an attacker. Demonstrating robustness under these conditions suggests that our approach should also be effective against attacks on more complex models.

In our experimental setup, we assume that the attacker (i.e., an honest but curious server) has full knowledge of the initial model parameters used by clients. Additionally, to maintain consistency with the performance analysis, the number of clients in the SFDD approach is set to 5.

The first set of experiments is meant to assess the robustness of the SFDD framework against the Deep Leakage attack (Zhu et al., 2019). Hence, we conducted preliminary experiments aimed at simulating the most advantageous scenario for an attacker. This involved implementing specific architectural changes in the model that distills the dataset on the client side. As illustrated in Tables 5 and 6, our approach empowered with the LDPO-RLD method effectively prevents the attacker from reconstructing the original image used by the client. Indeed, the results clearly show that the Mean Squared Errors (MSEs) increase significantly when using this countermeasure.

From these experiments, we can derive two important findings, namely *(i)* LDPO-RLD is effective as a countermeasure against deep leakage attacks; and *(ii)* LDPO-RLD has a negligible impact on the distillation performance.

The second experiment aims to compare the performance of the standard Local Differential Privacy (LDP) and our LDPO-RLD.

LDP requires each client to perform a grid search to determine the optimal hyperparameter configuration that prevents the attacker from reconstructing the original image. However, this process introduces significant computational overhead, particularly when clients seek a configuration that minimally impacts distillation performance. Finding

Table 5
MSE values between the normalized ground truth batch (from the real dataset) and the normalized reconstructed batch computed during the attack, with and without our defense.

Dataset	SFDD w/o LDPO-RLD	SFDD with LDPO-RLD
MNIST	0.245	1.575
CIFAR10	0.80	2.42
SVHN	0.71	2.02
GTSRB	0.725	1.84
CIFAR100	0.57	–

Table 6
Accuracy variation of a reference deep learning model trained on the dataset distilled by our approach with the LDPO-RLD defense compared to the same model trained on a dataset distilled using the original centralized distillation scheme.

Dataset	Accuracy variation (%)
MNIST	−0.48%
CIFAR10	+0.28%
SVHN	+0.10%
GTSRB	−1.11%
CIFAR100	+0.27%

Fig. 8. Visual comparison of gradient-leakage attack results on CIFAR-10 samples. **Left:** Images reconstructed by the attacker when LDPO-RLD defense is applied. **Middle:** Reconstructions without any defense mechanism. **Right:** Original private images from the client. The comparison illustrates how LDPO-RLD significantly limits the visual fidelity of leaked reconstructions.

the best hyperparameters may also require manual intervention, as estimating an appropriate MSE threshold for attack simulations during grid search is challenging (it heavily depends on the dataset's image characteristics). Additionally, to accurately determine an MSE threshold that ensures the desired level of privacy, multiple attack simulations with the same hyperparameters must be conducted to filter out non-significant variations in MSE. In contrast, our LDPO-RLD approach eliminates these overheads entirely. It operates seamlessly without requiring additional settings or manual inspection of grid search results, making it a more efficient, user-friendly, and reproducible solution.

Table 7 shows the average performance in terms of the variation in the accuracy results between LDP and LDPO-RLD across various datasets. From this table, we can observe that for the MNIST dataset,

Table 7
Average performance differences between LDPO-RLD and classical LDP.

Dataset	Accuracy variation (%)
MNIST	−0.29%
CIFAR10	0.60%
SVHN	2.56%
GTSRB	8.94%

there is a negligible decrease in performance with LDPO-RLD (−0.29%), which suggests that the two methods perform similarly for simpler datasets. With the CIFAR10 dataset, instead, there is a slight improvement of 0.60%, demonstrating that LDPO-RLD marginally outperforms LDP. SVHN experiment has a more significant improvement (2.56%), indicating that our method handles moderately complex datasets better than LDP. Finally, GTSRB has the highest improvement (8.94%), showing that LDPO-RLD significantly outperforms distillation performance for more complex datasets.

The results show that LDPO-RLD is not only an effective defense against deep leakage attacks but also outperforms LDP in distillation performance. As dataset complexity increases, the performance gap grows, underscoring LDPO-RLD's enhanced robustness and efficiency in real-world applications. These findings demonstrate that our LDPO-RLD countermeasure is both secure and advantageous, improving the overall SFDD framework's performance without the need for additional configurations or increased time complexity.

Moreover, we run a comprehensive grid search over the key hyperparameters of LDPO-RLD to better understand their effect on the privacy-utility trade-off. Specifically, we vary the number of obfuscation classes k and the smoothing parameter ϵ to observe their individual and combined impacts. As shown in Fig. 9, increasing k introduces a predictable decline in accuracy, as higher obfuscation adds ambiguity to the gradient signal. However, moderate settings such as $k = 3$ maintain strong performance and offer a favorable trade-off. Fig. 10 further shows that with no defense, the attacker obtains the lowest reconstruction error across all experiments, confirming the vulnerability of unprotected setups. In contrast, all defended configurations—regardless of ϵ or k result in markedly higher attack difficulty. Notably, even the minimal setting of $k = 2$ significantly increases both the attack loss and the MSE, demonstrating the immediate privacy benefit introduced by LDPO-RLD. These quantitative results are also visually supported in Fig. 8, where the attacker is able to recover recognizable image content in the absence of any defense, but fails to do so when LDPO-RLD is applied, producing reconstructions that are structurally and semantically uninformative.

The final experiment focuses on evaluating the effectiveness of our SFDD strategy against backdoor attacks. Specifically, we tested our solution using the Doorping attack (Liu et al., 2023), which serves as a benchmark for state-of-the-art backdoor attacks targeting dataset distillation. Our goal being the development of a framework that compiles synthetic data rather than local models, as typical in horizontal federated learning, led us to concentrate on cutting-edge research concerning backdoors within the distillation context, which do not align with poisoning attacks aimed at federated learning. To assess the maximum resilience of our approach, we considered the worst-case scenario where the attacker controls $\frac{z-1}{2}$ of the workers involved. The experiment was conducted across multiple scenarios, varying the number of workers and datasets, to observe how SFDD performs against the Doorping attack under different levels of complexity introduced by the distributed nature of our approach.

In Fig. 11, we report the results obtained in a configuration with 3, 5, and 10 clients. In particular, differently from the previous experiment, we report the results obtained with 3 clients to show that with a very small number of clients, the attack remains effective achieving an attack success rate close to 100%, the performance remains largely unchanged, indicating that in this setup, a distributed framework with a

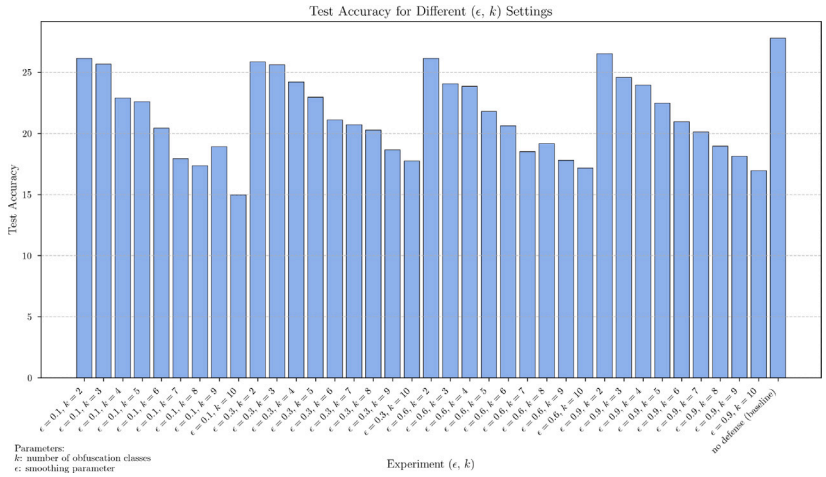


Fig. 9. Test accuracy on CIFAR-10 across different configurations of the LDPO-RLD hyperparameters: the smoothing parameter (ϵ) and the number of obfuscation classes (k). The results illustrate the trade-off between utility and privacy.

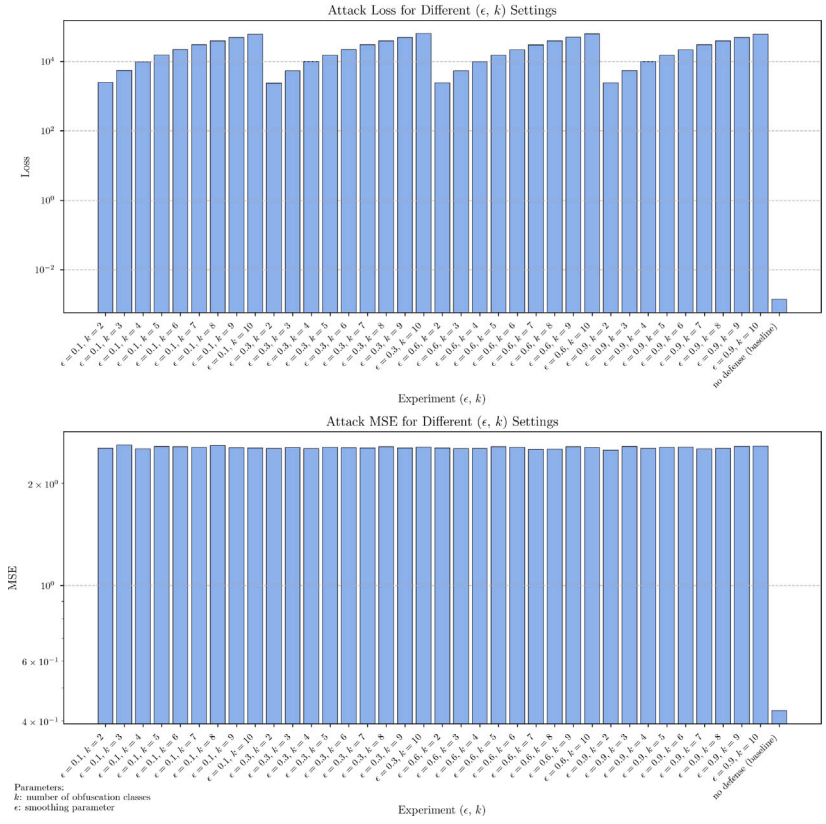


Fig. 10. Attack loss and mean squared error (MSE) for different settings of the smoothing parameter (ϵ) and the number of obfuscation classes (k). Each bar represents the result for a specific (ϵ, k) configuration. The top subplot shows the final attack loss, while the bottom subplot shows the corresponding MSE.

limited number of clients is insufficient to prevent the attack. However, such a setting is rare in practical Federated Learning deployments, which typically involve tens or hundreds of clients. As the number of clients increases, we observe a consistent and notable decline in the attack's success rate, especially for more complex datasets such as CIFAR-10 and GTSRB. This behavior aligns with an insight as the distillation process becomes more federated, the influence of any single (or small group of) malicious participant(s) is naturally diluted. The shared synthetic dataset is optimized from a broader, more diverse pool of client updates, which makes it significantly harder for adversarial signals to dominate the global gradient trajectory. In contrast to centralized dataset distillation, where a single compromised optimization pathway can inject targeted behavior into the distilled data, SFDD inherently imposes structural noise and diversity that counteracts this influence. While this robustness is a natural consequence of federating the distillation process, we acknowledge that stronger defenses can be integrated. In the Federated Learning literature, several robust aggregation techniques have been proposed to defend against adversarial or anomalous client updates. Notable examples include TrimmedMean (Yin et al., 2018), Bulyan (Guerraoui et al., 2018) and Krum (Blanchard et al., 2017). These methods aim to mitigate the influence of malicious participants by statistically filtering out outlier updates during aggregation. Translating these methods to our setting,

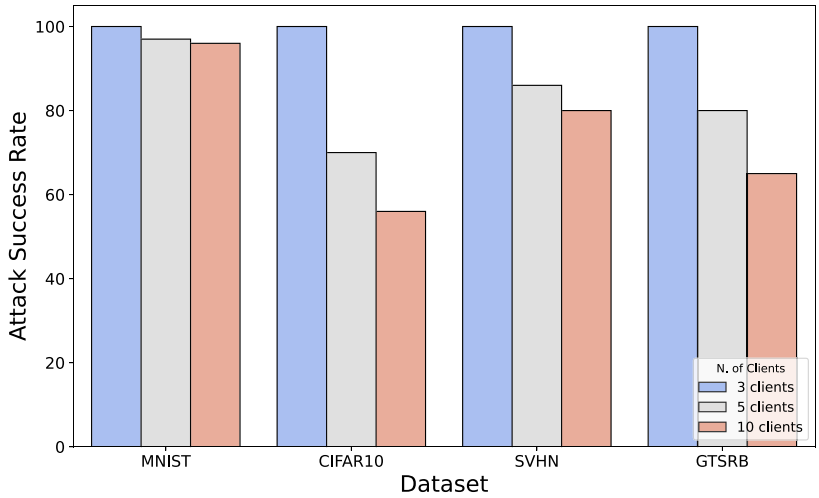


Fig. 11. Effect of SFDD on Doorping attack for different datasets.

however, presents new challenges: SFDD aggregates synthetic dataset updates rather than model weights, which means that traditional statistical anomaly detection must be adapted to image space. In future work, we plan to develop and evaluate principled filtering mechanisms to detect malicious synthetic update contributions, inspired by robust aggregation techniques from FL.

7. Conclusion

Dataset Distillation (DD) compresses the knowledge of an entire training data into a few synthetic training images. Current architectures for DD require a centralized entity that collects the data to be distilled in a single point of aggregation, thus leading to critical privacy concerns. To address these risks, we introduce a Secure and Federated Data Distillation (SFDD) framework, inspired by Federated Learning (FL), which decentralizes the distillation process while maintaining privacy. We apply a gradient-matching-based distillation method, modified for a distributed environment where clients participate in the distillation process without sharing raw data. The central aggregator iteratively refines a synthetic dataset by incorporating updates from clients while ensuring data confidentiality. To safeguard against potential inference attacks by the server, which could use gradient updates to reconstruct private data, we integrate an enhanced Local Differential Privacy (LocalDP) approach called LDPO-RLD (LabelDP Obfuscation via Randomized Linear Dispersion). Experiment results demonstrate that the SFDD approach consistently achieved performance comparable to the classic DD framework across different datasets. Moreover, we prove that the LDPO-RLD is an effective countermeasure against deep leakage attacks and does not affect the distillation performance. Additionally, we evaluate the framework's resilience to malicious clients carrying out backdoor attacks, such as Doorping, and show our framework's robustness in scenarios with a large number of participating clients.

Our proposed solution represents a significant step forward in secure dataset distillation, enabling multiple data owners to collaboratively generate a synthetic dataset without exposing their private data. As part of our future work, we plan to explore advanced cryptographic techniques, such as Secure Multi-party Computation (SMC) and homomorphic encryption, to further strengthen the privacy guarantees of our approach and lower the data leakage risk during the exchange of gradients. Additionally, we aim to develop more sophisticated defenses against emerging threats, including Doorping attacks, particularly in edge settings with limited client participation. Optimizing aggregation strategies and adaptive mechanisms to ensure robustness in non-IID and adversarial environments also represents a promising research direction. Beyond healthcare, we envision extending the SFDD framework to other domains requiring confidentiality-preserving collaboration. Additionally, an interesting direction consists of leveraging Reinforcement Learning to control the distillation schedule and personalize client objectives (Farhadi et al., 2024).

CRediT authorship contribution statement

Marco Arazzi: Writing – original draft, Software, Formal analysis, Conceptualization, Validation, Resources, Data curation. **Mert Cihangiroglu:** Writing – original draft, Validation, Investigation, Visualization, Software, Data curation. **Serena Nicolazzo:** Writing – review & editing, Visualization, Writing – original draft, Methodology. **Antonino Nocera:** Writing – review & editing, Methodology, Formal analysis, Supervision, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the project "GoTMaT - Governing Technology to Manage the Transition" founded by the European Community - Next Generation EU, Mission 4 Component 2 Investment 1.3 - CUP B53C22003990006 and by the project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU, as the work of Serena Nicolazzo was performed while she was with Università degli Studi di Milano. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the Italian MUR. Neither the European Union nor the Italian MUR can be held responsible for them.

The authors wish to express their gratitude to Daniele Murer (e-mail: daniele.murer01@universitadipavia.it) for his valuable contribution to the software used to test the initial version of our solution. This contribution helped to achieve the high quality and depth of our work.

Data availability

All the data used in this paper are public available datasets.

References

Afonin, A., Karimireddy, S.P., 2021. Towards model agnostic federated learning using knowledge distillation. arXiv preprint arXiv:2110.15210.

Aouedi, O., Sacco, A., Piamrat, K., Marchetto, G., 2022. Handling privacy-sensitive medical data with federated learning: challenges and future directions. IEEE J. Biomed. Heal. Inform. 27 (2), 790–803.

Arazzi, M., Cihangiroglu, M., Nocera, A., 2025a. Privacy preserving and robust aggregation for cross-silo federated learning in non-IID settings. arXiv preprint arXiv:2503.04451.

Arazzi, M., Conti, M., Koffas, S., Krcek, M., Nocera, A., Picek, S., Xu, J., 2023a. Label inference attacks against node-level vertical federated GNNs. arXiv preprint arXiv:2308.02465.

Arazzi, M., Conti, M., Nocera, A., Picek, S., 2023b. Turning privacy-preserving mechanisms against federated learning. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. pp. 1482–1495.

Arazzi, M., Koffas, S., Nocera, A., Picek, S., 2024. Let's focus: Focused backdoor attack against federated transfer learning. arXiv preprint arXiv:2404.19420.

Arazzi, M., Nicolazzo, S., Nocera, A., 2025b. A defense mechanism against label inference attacks in vertical federated learning. Neurocomputing 129476.

Blanchard, P., El Mhamdi, E.M., Guerraoui, R., Stainer, J., 2017. Machine learning with adversaries: Byzantine tolerant gradient descent. Adv. Neural Inf. Process. Syst. 30.

Cao, N., Du, H., Lu, J., Li, Z., Qiang, Q., Lu, H., 2025. Designing ionic liquid electrolytes for a rigid and li+-conductive solid electrolyte interface in high performance lithium metal batteries. Chem. Phys. Lett. 141959.

Cazenavette, G., Wang, T., Torralba, A., Efros, A.A., Zhu, J.-Y., 2022. Dataset distillation by matching training trajectories. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4750–4759.

Deng, L., 2012. The mnist database of handwritten digit images for machine learning research. IEEE Signal Process. Mag. 29 (6), 141–142.

Farhadi, A., Mirzarezaee, M., Sharifi, A., Teshnehlab, M., 2024. Domain adaptation in reinforcement learning: a comprehensive and systematic study. Front. Inf. Technol. Electron. Eng. 25 (11), 1446–1465.

Gu, T., Liu, K., Dolan-Gavitt, B., Garg, S., 2019. Badnets: Evaluating backdooring attacks on deep neural networks. IEEE Access 7, 47230–47244.

Guerraoui, R., Rouault, S., et al., 2018. The hidden vulnerability of distributed learning in byzantium. In: International Conference on Machine Learning. PMLR, pp. 3521–3530.

Hu, S., Goetz, J., Malik, K., Zhan, H., Liu, Z., Liu, Y., 2022. Fedsynth: Gradient compression via synthetic data in federated learning. arXiv preprint arXiv:2204.01273.

Jeong, E., Oh, S., Kim, H., Park, J., Bennis, M., Kim, S.-L., 2018. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. arXiv preprint arXiv:1811.11479.

Jia, Y., Vahidian, S., Sun, J., Zhang, J., Kungurtsev, V., Gong, N.Z., Chen, Y., 2024. Unlocking the potential of federated learning: The symphony of dataset distillation via deep generative latents. In: European Conference on Computer Vision. Springer, pp. 18–33.

Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S.J., Stich, S.U., Suresh, A.T., 2020. SCAFFOLD: Stochastic controlled averaging for federated learning. arXiv preprint arXiv:1910.06378. URL https://arxiv.org/abs/1910.06378.

Krizhevsky, A., Hinton, G., et al., 2009. Learning Multiple Layers of Features from Tiny Images (Master's thesis). Toronto, ON, Canada.

Kumar, R., Wang, W., Kumar, J., Yang, T., Khan, A., Ali, W., Ali, I., 2021. An integration of blockchain and AI for secure data sharing and detection of CT images for the hospitals. Comput. Med. Imaging Graph. 87, 101812.

Lei, S., Tao, D., 2023. A comprehensive survey of dataset distillation. IEEE Trans. Pattern Anal. Mach. Intell..

Li, M., Cui, C., Liu, Q., Deng, R., Yao, T., Lionts, M., Huo, Y., 2024. Dataset distillation in medical imaging: A feasibility study. arXiv preprint arXiv:2407.14429.

Li, Q., Diao, Y., Chen, Q., He, B., 2022a. Federated learning on non-iid data silos: An experimental study. In: 2022 IEEE 38th International Conference on Data Engineering. ICDE, IEEE, pp. 965–978.

Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V., 2020. Federated optimization in heterogeneous networks. In: Proceedings of Machine Learning and Systems. MLSys, URL https://arxiv.org/abs/1812.06127.

Li, G., Togo, R., Ogawa, T., Haseyama, M., 2022b. Dataset distillation for medical dataset sharing. arXiv preprint arXiv:2209.14603.

Li, D., Wang, J., 2019. Fedmd: Heterogenous federated learning via model distillation. arXiv preprint arXiv:1910.03581.

Lin, T., Kong, L., Stich, S.U., Jaggi, M., 2020. Ensemble distillation for robust model fusion in federated learning. Adv. Neural Inf. Process. Syst. 33, 2351–2363.

Liu, Y., Li, Z., Backes, M., Shen, Y., Zhang, Y., 2023. Backdoor attacks against dataset distillation. arXiv:2301.01197. URL https://arxiv.org/abs/2301.01197.

Lu, Z., Wang, J., Jiang, C., 2024. Data-free knowledge filtering and distillation in federated learning. IEEE Trans. Big Data.

Lyu, L., Yu, H., Ma, X., Chen, C., Sun, L., Zhao, J., Yang, Q., Philip, S.Y., 2022. Privacy and robustness in federated learning: Attacks and defenses. IEEE Trans. Neural Netw. Learn. Syst..

McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A., 2017. Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics. PMLR, pp. 1273–1282.

Nasr, M., Shokri, R., Houmansadr, A., 2019. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In: 2019 IEEE Symposium on Security and Privacy. SP, IEEE, pp. 739–753.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y., et al., 2011. Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning. Vol. 2011, Granada, Spain, p. 7, (5).

Pan, Q., Sun, S., Wu, Z., Wang, Y., Liu, M., Gao, B., Wang, J., 2024. FedCache 2.0: Federated edge learning with knowledge caching and dataset distillation. arXiv preprint arXiv:2405.13378.

Pan, Y., Xu, K., Wang, R., Wang, H., Chen, G., Wang, K., 2025. Lithium-ion battery condition monitoring: A frontier in acoustic sensing technology. Energies 18 (5), 1068.

Pang, X., Hu, J., Sun, P., Ren, J., Wang, Z., 2024. When federated learning meets knowledge distillation. IEEE Wirel. Commun. 31 (5), 208–214.

Song, R., Liu, D., Chen, D.Z., Festag, A., Trinitis, C., Schulz, M., Knoll, A., 2023. Federated learning via decentralized dataset distillation in resource-constrained edge environments. In: 2023 International Joint Conference on Neural Networks. IJCNN, IEEE, pp. 1–10.

Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C., 2012. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. Neural Netw. 32, 323–332.

Sun, G., Cong, Y., Dong, J., Wang, Q., Lyu, L., Liu, J., 2021. Data poisoning attacks on federated machine learning. IEEE Internet Things J. 9 (13), 11365–11375.

Wang, J., Liu, Q., Liang, H., Joshi, G., Poor, H.V., 2020. Tackling the objective inconsistency problem in heterogeneous federated optimization. arXiv preprint arXiv:2007.07481. URL https://arxiv.org/abs/2007.07481.

Wang, T., Zhu, J.-Y., Torralba, A., Efros, A.A., 2018. Dataset distillation. arXiv preprint arXiv:1811.10959.

Weitzman, E.R., Kaci, L., Mandl, K.D., 2010. Sharing medical data for health research: the early personal health record experience. J. Med. Internet Res. 12 (2), e1356.

Xiong, Y., Wang, R., Cheng, M., Yu, F., Hsieh, C.-J., 2023. Feddm: Iterative distribution matching for communication-efficient federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16323–16332.

Yang, Q., Liu, Y., Chen, T., Tong, Y., 2019. Federated machine learning: Concept and applications. ACM Trans. Intell. Syst. Technol. (TIST) 10 (2), 1–19.

Yin, D., Chen, Y., Kannan, R., Bartlett, P., 2018. Byzantine-robust distributed learning: Towards optimal statistical rates. In: International Conference on Machine Learning. Pmlr, pp. 5650–5659.

Zhang, H., Li, Z., Liu, Y., Du, X., Gao, Y., Xie, W., Zheng, X., Du, H., 2025. Oxygen vacancies-modulated C-WO3/biobr heterojunction for highly efficient benzene degradation. Vacuum 114117.

Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., Gao, Y., 2021. A survey on federated learning. Knowl.-Based Syst. 216, 106775.

Zhao, B., Mopuri, K.R., Bilen, H., 2021. Dataset condensation with gradient matching. arXiv:2006.05929. URL https://arxiv.org/abs/2006.05929.

Zhou, Y., Pu, G., Ma, X., Li, X., Wu, D., 2020. Distilled one-shot federated learning. arXiv preprint arXiv:2009.07999.

Zhu, Z., Hong, J., Zhou, J., 2021. Data-free knowledge distillation for heterogeneous federated learning. In: International Conference on Machine Learning. PMLR, pp. 12878–12889.

Zhu, L., Liu, Z., Han, S., 2019. Deep leakage from gradients. Adv. Neural Inf. Process. Syst. 32.