PROTECTING SENSITIVE DATA THROUGH FEDERATED CO-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

In many critical applications, sensitive data is inherently distributed. Federated learning trains a model collaboratively by aggregating the parameters of locally trained models. This avoids exposing sensitive local data. It is possible, though, to infer upon the sensitive data from the shared model parameters. At the same time, many types of machine learning models do not lend themselves to parameter aggregation, such as decision trees, or rule ensembles. It has been observed that in many applications, in particular healthcare, large unlabeled datasets are publicly available. They can be used to exchange information between clients by distributed distillation, i.e., co-regularizing local training via the discrepancy between the soft predictions of each local client on the unlabeled dataset. This, however, still discloses private information and restricts the types of models to those trainable via gradient-based methods. We propose to go one step further and use a form of federated co-training, where local hard labels on the public unlabeled datasets are shared and aggregated into a consensus label. This consensus label can be used for local training by any supervised machine learning model. We show that this federated co-training approach achieves a model quality comparable to both federated learning and distributed distillation on a set of benchmark datasets and real-world medical datasets. It improves privacy over both approaches, protecting against common membership inference attacks to the highest degree. Furthermore, we show that federated co-training can collaboratively train interpretable models, such as decision trees and rule ensembles, achieving a model quality comparable to centralized training.

1 INTRODUCTION

Can we collaboratively train models from distributed sensitive datasets while maintaining data privacy at a level required in critical applications, such as healthcare? Federated learning (FL) (McMahan et al., 2017) allows distributed sites, e.g., hospitals or clinics, to collaboratively train a joint model without directly disclosing their sensitive data by instead periodically sharing and aggregating parameters of locally trained models. However, it is possible for an attacker or curious observer to make non-trivial inferences about local data from model parameters (Ma et al., 2020) and model updates (Zhu & Han, 2020). Differential privacy provides a rigorous and measurable privacy guarantee (Dwork et al., 2014) that can be achieved by perturbing model parameters appropriately (Wei et al., 2020). But, this perturbation can reduce model quality, resulting in a trade-off between privacy and quality that is typically poor: differentially private distributed SGD with descent utility has slim to no actual privacy (i.e., $\epsilon = 145, \delta = 10^{-5}$) (Xiao et al., 2022). Moreover, federated learning requires models that can be aggregated, i.e., models with a parameterization in a vector space such that geometric operations like averaging can be applied. This excludes many interpretable models, such as XGBoost, decision trees, Random forest, and rule ensembles.

Distributed distillation (DD) (Bistritz et al., 2020) uses a distributed knowledge distillation approach. It shares soft predictions on a shared unlabeled dataset and adds a local regularization term that promotes clients to agree on the unlabeled data, similar to co-regularization (Sindhwani et al., 2005; Ullrich et al., 2017). Since in deep learning, models are typically large, while the unlabeled dataset can be moderate in size, this approach can substantially reduce communication. At the same time, it allows each client to use a different network architecture. However, it excludes types of models that are not trainable via gradient-based methods.

We propose to use a federated form of co-training that does not share soft predictions but goes one step further: clients iteratively share predictions on the unlabeled dataset, the server forms a consensus, and clients use this consensus as pseudo-labels for the unlabeled dataset in their local training. A straightforward consensus mechanism for classification problems is a majority vote. This federated co-training (FEDCT) allows us to locally use any supervised learning method. At the same time, sharing only hard predictions improves privacy not only over federated learning but also over distributed distillation, while retaining the communication advantage of DD.

We show theoretically that FEDCT converges for local learning methods with increasing training accuracy, and that ϵ-differential privacy can be achieved by applying a randomized mechanism suitable to binary data, such as the XOR-mechansim (Ji et al., 2021). In an extensive empirical evaluation on classification problems, we show that federated co-training achieves a model quality similar to federated learning and distributed distillation on three benchmarks and two real-world medical datasets, including non-iid data. At the same time, the empirical vulnerability to privacy attacks (Murakonda & Shokri, 2020) is substantially lower than standard FL, FL with differential privacy (Noble et al., 2022) and distributed distillation. For example, on the Pneumonia dataset, FEDCT achieved a vulnerability score of 0.51 (i.e., the success probability of a membership inference attack, so this is basically a random guess), compared to 0.76 (FEDAVG) and 0.63 (DD). Furthermore, we show that FEDCT collaboratively trains decision trees, rule ensembles, random forests, and XGBoost to a quality similar to centralized training on 5 benchmark datasets.

Our contributions are

(i) a novel federated co-training (FedCT) approach to collaboratively train models from privacy-sensitive distributed data sources via a public unlabeled dataset that achieves model quality comparable to standard federated learning and distributed distillation;

(ii) a practical and theoretical privacy analysis showing that FedCT achieves an excellent privacy-utility trade-off, i.e., a high utility for differentially private FedCT even under high privacy demands, and in practice nearly no vulnerability to known membership inference attacks.

(iii) and, the ability to seamlessly integrate any supervised learning method on clients in the federated system, including interpretable models, such as XGboost, decision trees, Random forest, and rule ensembles.

2 RELATED WORK

Semi-supervised learning: Semi-supervised learning utilizes both a labeled and unlabeled dataset, where the unlabeled set is typically large (Zhou & Li, 2005; Rasmus et al., 2015). Co-training is a semi-supervised learning approach where two classifiers are independently trained on two distinct feature sets of labeled data. It has been used to improve models using unlabeled data, typically in centralized multi-view settings (Blum & Mitchell, 1998; Ullrich et al., 2017). Semi-supervised learning has also been used in knowledge distillation.Papernot et al. (2016) proposed to collaboratively local train models from distributed sensitive datasets via using a teachers-student scheme. In a setting where teachers are trained locally on distributed sensitive datasets and then the majority vote over their predictions on unlabeled data is used to train a student network. They show that the data with teachers trained can be protected by adding Laplacian noise to the majority vote. Since the Laplacian mechanism is only applied to the majority vote, the individual predictions remain unprotected.

Distributed semi-supervised learning: Bistritz et al. (2020) propose to share soft predictions on a public unlabeled dataset instead of model parameters to reduce communication in federated deep learning. Inspired by knowledge distillation, this co-regularizes local models to have similar soft predictions. This approach performs similar to distributed SGD and—in contrast to federated learning—allows local neural networks to have different architectures. Chen & Chao (2020) presented FedBE, which employs knowledge distillation to train a student model based on predictions from a Bayesian model ensemble. Similarly, (Lin et al., 2020)'s FedDF also uses knowledge distillation in a federated context to create a global model by fusing client models. While FedDF allows for local neural models with varying sizes or structures, this method still requires a differentiable loss function.

Privacy in Federated Learning: Collaboratively training a model without sharing sensitive data is a key advantage of (horizontal) federated learning (McMahan et al., 2017) which trains local models and aggregates their parameters periodically. It has been shown, however, that communicating only

model parameters for aggregation does not entirely protect local data: An attacker or curious observer can make inferences about local data from model parameters (Shokri et al., 2017; Ma et al., 2020) and model updates (Zhu & Han, 2020). Should a malicious client obtain model updates through additional attacks, a common defense is applying appropriate clipping and noise before sending models. This guarantees ϵ, δ-differential privacy for local data (Wei et al., 2020) at the cost of a moderate loss in model quality. This technique is also proven to defend against backdoor and poisoning attacks (Sun et al., 2019). The practical utility-privacy trade-off, however, is poor: in fact, DP-Dist-SGD with descent utility achieves $(\epsilon = 145, \delta = 10^{-5})$-differential privacy) (Xiao et al., 2022). Note that the probability of an adversary learning about an individual from a dataset of size n is $\gtrsim n^{-1}e^{\epsilon}$ (Lee & Clifton, 2011). Truex et al. (2019) proposes enhancing the privacy of data exchange in traditional distributed algorithms through the use of secure multi-party communication (SMPC) and differential privacy (DP). While this enables the application of both classical distributed decision tree algorithms and federated learning methods, SMPC has scalability and efficiency limitations and DP involves a trade-off between privacy and utility. Moreover, this approach does not allow the federated training of decision trees, that is, training local models and aggregating them.

3 Federated Semi-Supervised Learning

3.1 Preliminaries

We assume learning algorithms $\mathcal{A} : \mathcal{X} \times \mathcal{Y} \to \mathcal{H}$ that produce models $h \in \mathcal{H}$ using a dataset $D \subset \mathcal{X} \times \mathcal{Y}$ from an input space $\mathcal{X}$ and output space $\mathcal{Y}$, i.e., $h_{t+1} = \mathcal{A}(D)$, or iterative learning algorithms (cf. Chp. 2.1.4 Kamp, 2019) $\mathcal{A} : \mathcal{X} \times \mathcal{Y} \times \mathcal{H} \to \mathcal{H}$ that update a model $h_{t+1} = \mathcal{A}(D, h_t)$. Given a set of $m \in \mathbb{N}$ clients with local datasets $D^1, \ldots, D^m \subset \mathcal{X} \times \mathcal{Y}$ drawn iid from a data distribution $\mathcal{D}$ and a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, the goal is to find a set of local models $h^{1*}, \ldots, h^{m*} \in \mathcal{H}$ that each minimize the risk

$$\varepsilon(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(h(x), y)] \ . \tag{1}$$

In *centralized learning*, datasets are pooled as $D = \bigcup_{i\in[m]} D^i$ and $\mathcal{A}$ is applied to D until convergence. Note that applying $\mathcal{A}$ on D can be the application to any random subset, e.g., as in mini-batch training, and convergence is measured in terms of low training loss, small gradient, or small deviation from previous iterations. In standard *federated learning* (McMahan et al., 2017), $\mathcal{A}$ is applied in parallel for $b \in \mathbb{N}$ rounds on each client locally to produce local models $h^1, \ldots, h^m$. These models are then centralized and aggregated using an aggregation operator $\mathrm{agg} : \mathcal{H}^m \to \mathcal{H}$, i.e., $\overline{h} = \mathrm{agg}(h^1, \ldots, h^m)$. The aggregated model $\overline{h}$ is then redistributed to local clients which perform another b rounds of training using $\overline{h}$ as a starting point. This is iterated until convergence of $\overline{h}$. When aggregating by averaging, this method is known as federated averaging (FedAvg).

In federated semi-supervised learning, a public unlabeled dataset U is available to all clients. Distributed distillation (Bistritz et al., 2020) proposes to share soft predictions of clients on U and incorporate them into the optimization problem, similar to knowledge distillation. This can also be viewed as a distributed form of co-regularization (Sindhwani et al., 2005; Ullrich et al., 2017), where clients take up the role of views. This approach allows using different network architectures at each client, but requires gradient-based methods for local training.

3.2 A Federated Co-Training Approach

We propose to produce a pseudo-labeling of U as a consensus of the labels generated by the local models of each client, resulting in a federated form of co-training (Blum & Mitchell, 1998). That is, in a communication round $t \in \mathbb{N}$ each client $i \in [m]$ shares local labels $L_t^i = h_t^i(U)$ (not soft predictions) on U with the server, which produces a consensus labeling $L \subset \mathcal{Y}$ via an appropriate consensus mechanism. The consensus labels are used to augment local datasets. We call this approach federated co-training (FedCT). Sharing hard labels not only improves privacy over both federated averaging and distributed distillation, but also allows us to use any supervised learning method for local training. We describe federated co-training in Algorithm 1: at each client i, the local model is updated using the local dataset D^i combined with the current pseudo-labeled public dataset P (line 4). In a communication round (line 5), the updated model is used to produce improved pseudo-labels L^i for the unlabeled data U (line 6), which are sent to a server (line 7). At the server, as soon as all

Algorithm 1: Federated Co-Training (FEDCT)

Input: communication period b, m clients with local datasets $D^1, \ldots, D^m$ and local learning algorithms algorithms $\mathcal{A}^1, \ldots \mathcal{A}^m$, unlabeled shared dataset U, total number of rounds T
Output: final models $h_T^1, \ldots, h_T^m$

1 initialize local models $h_0^1, \ldots, h_0^m$, $P \leftarrow \emptyset$
2 **Locally** *at client i at time t* **do**
3 $h_t^i \leftarrow \mathcal{A}^i(D^i \cup P, h_{t-1}^i)$
4 **if** $t \,\%\, b = b - 1$ **then**
5 $L_t^i \leftarrow h_t^i(U)$
6 send L_t^i to server and receive L_t
7 $P \leftarrow (U, L_t)$
8 **end**
9 **At server** *at time t* **do**
10 receive local pseudo-labels $L_t^1, \ldots, L_t^m$
11 $L_t \leftarrow \text{consensus}(L_t^1, \ldots, L_t^m)$
12 send L_t to all clients

local prediction $L^1, \ldots, L^m$ are received (line 12), a consensus L is formed (line 13) and broadcasted back to the clients (14). At the client, upon receiving the consensus labels (line 8), the pseudo-labeled dataset is updated (line 9), and another iteration of local training is performed. For classification problems where $\mathcal{Y} \subset \mathbb{N}$, the majority vote is a reliable consensus mechanism (Papernot et al., 2016).

Convergence Analysis: The convergence of federated co-training depends of course on the convergence of the local learning algorithms $\left(\mathcal{A}^i\right)_{i\in[m]}$. Under the natural assumption that these algorithms converge on a fixed training set, it remains to show that there is a time from which the training set does not change anymore. That is, there exists a round $t_0 \in \mathbb{N}$ such that for all $t > t_0$ it holds that $L_t = L_{t-1}$. For classification problems, this naturally depends on the local training accuracy. If local training accuracy $a_t = 1.0$, then the approach trivially converges, since local models will reproduce L_t in every subsequent round. This assumption is usually fulfilled for over-parameterized models. In the following, we show that the approach also converges with high probability, if the training accuracy is ≤ 1, but linearly increasing with t.

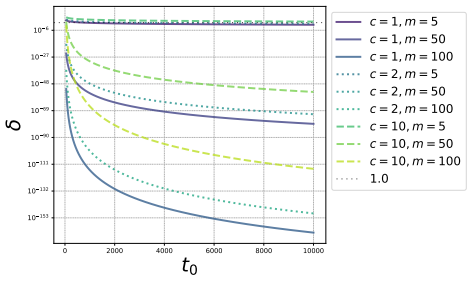

Figure 1: Numerical evaluation of the upper bound on δ for $|U| = 10000$.

Proposition 1. *For $m \geq 3$ clients with local datasets $D^1, \ldots, D^m$ and unlabeled dataset U drawn iid from $\mathcal{D}$, let $\mathcal{A}^i$ for $i \in [m]$ be a set of learning algorithms that all achieve a linearly increasing training accuracy a_t for all labelings of U, i.e., there exists $c \in \mathbb{R}_+$ such that $a_t \geq 1 - c/t$, then there exists $t_0 \in \mathbb{N}$ such that $a_t \geq 1/2$ and* FEDCT *with majority vote converges with probability $1 - \delta$, where*

$$\delta \leq |U|(4c)^{\frac{m}{2}} \zeta\left(\frac{m}{2}, t_0 + 1\right)$$

and $\zeta(x, q)$ is the Hurwitz zeta function.

Proof. **Sketch:**We show that if local models are of sufficient quality, then in round $t \geq t_0$, the probability that the consensus labels change, δ_t, is bounded. Indeed, the probability can be determined via the CDF of the binomial distribution, which can be bounded via the Chernoff bound, yielding

$$\delta_t \leq |U| 4^{\frac{m}{2}} a_t^{\frac{m}{2}} (1 - a_t)^{\frac{m}{2}} \quad .$$

We then show that the probability that the consensus labels remain constant for the remainder, i.e., the sum of δ_t from t_0 to ∞, is bounded as well. Using the assumption that a_t grows linearly, we can

express this infinite series as

$$\sum_{t=t_0}^{\infty} \delta_t \lesssim \sum_{t=0}^{\infty} \frac{1}{t}^{\frac{m}{2}} - \sum_{t=0}^{t_0} \frac{1}{t}^{\frac{m}{2}} \quad ,$$

that is, the difference of the Riemann zeta function and the t_0-th generalized harmonic number, $\sum_{t=t_0}^{\infty} \delta_t \lesssim \zeta(m/2) - H_{t_0}^{m/2}$.This difference can be expressed via the Hurwitz zeta function $\zeta(m/2, t_0 + 1)$. □

The full proof is provided in Appendix A. Note that $\delta \to 0$ for $t_0 \to \infty$, and δ is monotonically decreasing with m. We plotted δ wrt. t_0 in Figure 1. For $t_0 = 1000$, FEDCT converges with probability ≈ 1.0 for $m = 50$ and $m = 100$ with $c \in \{1, 2, 10\}$. It converges with $1 - \delta = 0.9993$ for $c = 1, m = 5$, $1 - \delta = 0.9962$ for $c = 2, m = 5$, and $1 - \delta = 0.7868$ for $c = 10, m = 5$.

Communication Complexity: The communication complexity of FEDCT is in the same order as standard federated learning, i.e., treating the message size as a constant, the communication complexity is in $\mathcal{O}(T/b)$, where b is the communication period. However, the number of bits transferred in each round depends on the size of U. Since the predictions can be encoded as binary vectors, for a classification problem with $C \in \mathbb{N}$ classes the communication complexity is in $\mathcal{O}(TC|U|/b)$. As Bistritz et al. (2020) observed, transferring predictions on U can reduce communication substantially over transferring the weights of large neural networks. For example, with $|U| = 10^4$, FEDCT achieved an ACC of 0.80 on a neural network with $669\,706$ parameters, reducing communication over FEDAVG by a factor of ≈ 67.

4 DIFFERENTIALLY PRIVACY FOR FEDERATED CO-TRAINING

We assume the following attack model: clients are honest and the server may be semi-honest (follow the protocol execution correctly, but it may try to infer sensitive information about the clients). The main goal of a semi-honest server is to infer sensitive information about the local training data of the clients. This is a stronger attacker assumption compared to a semi-honest client since the server receives the most amount of information from the clients during the protocol, and a potential semi-honest client can only obtain indirect information about the other clients. We also assume that parties do not collude. Details are referred to Appendix E.1. Sharing predictions on an unlabeled dataset (pseudo-labeling) empirically improves the privacy of sensitive local data substantially, in particular, since FEDCT only shares predictions on an unlabeled dataset, as we show in Section 5. Note that this differs from label leakage (Li & Zhang, 2021), where predictions on the private data are shared. An empirical improvement in privacy is, however, no guarantee. Differential privacy instead provides a fundamental guarantee of privacy which is achieved through randomization of shared information.

Definition 1 ((Dwork et al., 2014)). *A randomized mechansim $\mathcal{M}$ with domain $\mathcal{X}$ and range $\mathcal{Y}$ is ϵ-differential private if for any two neighboring inputs $D, D^{'} \subset \mathcal{X}$ and for a substet of outputs $S \in \mathcal{Y}$ it holds that*

$$P\left(\mathcal{M}(D) \in S\right) \leq \exp(\epsilon) P\left(\mathcal{M}(D^{'}) \in S\right) \quad .$$

To obtain differential privacy (DP), the randomization has to be suitable to the information that is published. In FEDCT local clients share the predictions on an unlabeled dataset. For classification, this means sharing binary vectors. Standard DP mechanisms, like the Gaussian (Dwork et al., 2014) or Laplacian mechanissm (Dwork et al., 2006) are not suitable for binary data. Therefore, we use a DP mechanism for binary data based on computing the XOR operation of the original data and a random binary matrix (Ji et al., 2021).

The XOR-Mechanism: Federated co-training shares predictions on an unlabeled dataset that for classification problems can be interpreted as binary matrices via one-hot encoding. With a given unlabeled dataset U and a classification problem with $C \in \mathbb{N}$ classes, the predictions sent by a client with local dataset $D \subset \mathcal{X}$ to the server can be interpreted as the binary matrix output of a deterministic mechanism $f(D) \in \{0, 1\}^{|U| \times C}$. Given two neighboring datasets D, D' (i.e., they differ only in a single element), the sensitivity of f is defined as $s_f = \sup_{f(D), f(D')} \|f(D) \oplus f(D')\|_F^2$, where

$\oplus$ denotes binary XOR. Now let $\mathcal{B} \in \{0,1\}^{N \times P}$ to denote a matrix-valued Bernoulli random variable, i.e., $\mathcal{B} \sim \text{Ber}_{N,P}(\mathbf{\Theta}, \mathbf{\Lambda}_{1,2}, \cdots, \mathbf{\Lambda}_{N-1,N})$ with a matrix-valued Bernoulli distribution with quadratic exponential dependence structure. Here, $\mathbf{\Theta}$ is the $P \times P$ association parametric matrix including log-linear parameters describing the association structure of the columns, and $\mathbf{\Lambda}_{i,j}$ is the $P \times P$ association parametric matrix of rows i and j. The XOR-mechanism applies this random matrix to the output of the deterministic mechanism via the XOR operator $\oplus$ and yields a randomized mechanism $\mathcal{M}(D) = f(D) \oplus \mathcal{B}$. Applying this XOR-mechanism to federated co-training means representing local predictions L_t^i as binary matrices and producing randomized predictions $\widehat{L}_t^i = L_t^i \oplus \mathcal{B}$ that are then send to the server, resulting in differentially private distributed co-training (DP-FEDCT): Defining the sensitivity of DP-FEDCT as $s_* = \max\{s_{f^1}, \ldots, s_{f^m}\}$, it follows directly from Theorem 1 in Ji et al. (2021) that DP-FEDCT achieves ϵ-differential privacy.

Corollary 1. *Applying XOR mechanism to* FEDCT *with sensitivity s_* achieves ϵ-DP if $\mathbf{\Theta}$ and $\mathbf{\Lambda}_{i,j}$ satisfy*

$$s_*(||\boldsymbol{\lambda}(\mathbf{\Theta})||_2 + \textstyle\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} ||\boldsymbol{\lambda}(\mathbf{\Lambda}_{i,j})||_2\Big) \leq \epsilon, \tag{2}$$

where $||\boldsymbol{\lambda}(\mathbf{\Theta})||_2$ and $||\boldsymbol{\lambda}(\mathbf{\Lambda}_{i,j})||_2$ are the l_2 norms of the eigenvalues of $\mathbf{\Theta}$ and $\mathbf{\Lambda}_{i,j}$.

It remains to bound the sensitivity of FEDCT. The sensitivity of FEDCT measures how much the predictions of a client on the unlabeled dataset can change if one element of its local training set is removed. For learning algorithms that are on-average-one-stable, the sensitivity can be bounded.

Definition 2 ((Shalev-Shwartz & Ben-David, 2014))**.** *(On-Average-Replace-One-Stable) Let $\epsilon : \mathbb{N} \to \mathbb{R}$ be a monotonically decreasing function, and ℓ a loss function. We say that a learning algorithm $\mathcal{A}$ is on-average-replace-one-stable with rate $\epsilon(m)$ if for every distribution $\mathcal{D}$*

$$\mathop{\mathbb{E}}_{(S,z') \sim \mathcal{D}^{m+1}, i \sim U(m)} \left[\ell\left(\mathcal{A}\left(S^{(i)}, z_i\right)\right) - \ell\left(\mathcal{A}(S), z_i\right)\right] \leq \epsilon(m).$$

Using this definition, we obtain the following bound for the sensitivity.

Proposition 2. *For classification models $h : \mathcal{X} \to \mathcal{Y}$, let ℓ be a loss function that upper bounds the $0-1$-loss and $\mathcal{A}$ a learning algorithm that is on-average-leave-one-out stable with stability rate $\epsilon(m)$ for ℓ. Let $D \cup U$ be a local training set with $|U| = n$, and $\delta \in (0,1)$. Then with probability $1-\delta$, the sensitivity s_* of $\mathcal{A}$ on U is bounded by*

$$s_* \leq \left\lceil n\epsilon(n) + P\sqrt{n\epsilon(n)(1-\epsilon(n))} + \frac{P^2}{3} \right\rceil \quad ,$$

where $P = \Phi^{-1}(1-\delta)$ with $= \Phi^{-1}$ being the probit function.

The proof is provided in Appendix B. On-average-replace-one-stability holds for many supervised learning methods. For example, every regularized risk minimizer for a convex, Lipschitz loss using a strongly convex regularizer, like Thikonov-regularization, is on-average-replace-one-stable (cf. Chp. 13.3 in Shalev-Shwartz & Ben-David, 2014). We empirically evaluate the privacy-utility trade-off of FEDCT with differential privacy in Sec. 5.

5 EMPIRICAL EVALUATION

We empirically show that federated co-training presents a more favorable privacy-utility trade-off compared to federated learning by showing that it achieves similar test accuracy with substantially improved privacy. We compare FEDCT to standard federated averaging (McMahan et al., 2017) (FEDAVG), differentially private federated averaging (DP-FEDAVG) achieved through applying the Gaussian mechanism to FEDAVG (Geyer et al., 2017), and distributed distillation (Bistritz et al., 2020) (DD)[1] on 5 benchmark datasets and 2 medical image classification datasets.

Experimental Setup We evaluate FEDCT on three benchmark image classification datasets, CIFAR10 (Krizhevsky et al., 2010), FashionMNIST (Xiao et al., 2017), and SVHN (Netzer et al., 2011), as well as two real medical image classification datasets, MRI scans for brain tumors[2], and

[1]The code is available at https://anonymous.4open.science/r/federatedcotraining-B03C

[2]https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection

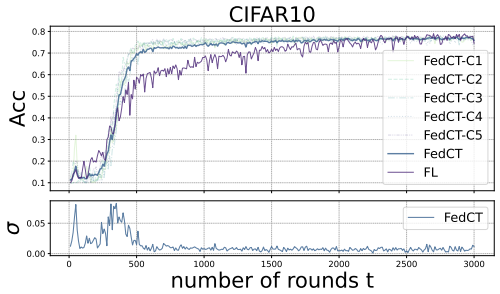

Figure 2: **Top**: Test accuracy (ACC) over time on CIFAR10 with ACC of FL, and ACC of local models and their average for FEDCT. **Bottom**: Standard deviation of test accuracy of local models in FEDCT.

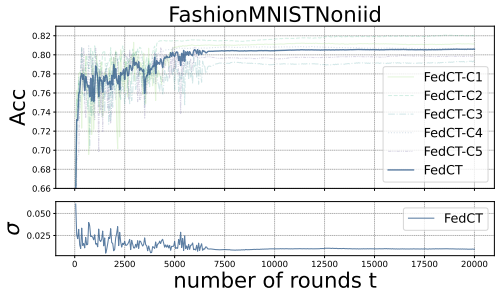

Figure 3: **Top**: Test accuracy (ACC) over time for $m = 5$ local models of FEDCT on heterogeneous distribution for the FashionMNIST dataset. **Bottom**: Standard deviation of test accuracy of local models in FEDCT.

chest X-rays for pneumonia (Kermany et al., 2018). For interpretable models, we use five benchmark datasets, WineQuality (Cortez et al., 2009), Breastcancer (Street et al., 1993), AdultsIncome (Becker & Kohavi, 1996), Mushroom (Bache & Lichman, 1987), and Covertype (Blackard, 1998). We first divide each dataset into a test and training set and further divide the training set into an unlabeled dataset U and a set of m local training sets (sampling iid. for all experiments, except for the experiments on heterogeneous data distributions). The architectures of the neural networks are provided in Appendix E. The parameters of the optimizer, as well as the communication period, are optimized individually for all methods on a subset of the training set via cross-validation. We select the number of rounds to be the maximum rounds required so that all methods converge, i.e., $T = 2 * 10^4$.

We measure empirical privacy vulnerability by performing a large number of membership inference attacks and compute the probability of inferring upon sensitive data, using the ML Privacy Meter tool (Murakonda & Shokri, 2020). The **vulnerability (VUL)** of a method is the ROC AUC of membership attacks over K runs over the entire training set. A vulnerability of 1.0 means that membership can be inferred with certainty, whereas 0.5 means that deciding on membership is a random guess.

Privacy-Utility-Trade-Off: We first evaluate the performance of FEDCT and the baselines for deep learning on a homogeneous data distribution for 5 image classification datasets. We use an unlabeled dataset of size $|U| = 10^4$ for CIFAR10, $|U| = 5 \cdot 10^4$ for FashionMNIST, $|U| = 170$ for MRI, $|U| = 900$ for Pneumonia, and $|U| = 35 \cdot 10^4$ for SVHM. Note that only FEDCT, differentially private FEDCT (DP-FEDCT), and distributed distillation (DD) use the unlabeled dataset. The remaining training data is distributed among the $m = 5$ clients. We repeat all experiments 3 times and report average test accuracy and standard deviation. Further details are deferred to Appendix E.

The results presented in Table 1 show that FEDCT achieves a test accuracy comparable to both FEDAVG and DD, while preserving privacy to the highest level. That is, FEDCT performs best on CIFAR10, has a similar performance to both on FashionMNIST, Pneumonia, and SVHN, and is slightly worse on MRI. The vulnerability is around 0.5, so that membership inference attacks are akin to random guessing. FEDAVG instead consistently has a vulnerability over 0.7. DP-FEDAVG improves privacy, but also reduces the test accuracy substantially. Our experiments show that DD substantially improves privacy over both FEDAVG and DP-FEDAVG, yet it is still vulnerable ($VUL \approx 0.6$). We show the convergence behavior of individual client models in FEDCT in terms of test accuracy on CIFAR10 and compare it to FEDAVG in Figure 2. FEDCT converges faster than FEDAVG, though the latter increases its test accuracy slightly further, eventually. Plotting the standard deviation of test accuracies of local models in Figure 2, we see that they converge to a consensus after around 700 rounds with only slight deviations afterward.

Privacy-Utility Trade-Off With Differential Privacy: Differential privacy guarantees typically come at a cost in terms of utility, which in our case means a loss in model quality. Analyzing this privacy-utility trade-off requires estimating the sensitivity. Since stability-bounds for neural

Dataset	FEDCT		DP-FEDCT		FEDAVG		DP-FEDAVG		DD	
	ACC	VUL	ACC	VUL	ACC	VUL	ACC	VUL	ACC	VUL
CIFAR10	$\mathbf{0.77} \pm 0.003$	0.52	0.76 ± 0.002	**0.51**	$\mathbf{0.77} \pm 0.020$	0.73	0.68 ± 0.002	0.70	0.67 ± 0.012	0.61
FashionMNIST	0.82 ± 0.004	**0.51**	0.80 ± 0.001	0.52	$\mathbf{0.83} \pm 0.024$	0.72	0.69 ± 0.002	0.71	0.82 ± 0.016	0.60
Pneumonia	0.76 ± 0.008	**0.51**	0.75 ± 0.004	**0.51**	0.74 ± 0.013	0.76	0.61 ± 0.004	0.69	$\mathbf{0.77} \pm 0.003$	0.63
MRI	0.63 ± 0.004	0.52	0.62 ± 0.002	**0.51**	0.66 ± 0.015	0.73	0.56 ± 0.003	0.62	$\mathbf{0.68} \pm 0.008$	0.60
SVHN	0.88 ± 0.002	**0.53**	0.86 ± 0.001	**0.53**	$\mathbf{0.91} \pm 0.026$	0.71	0.71 ± 0.005	0.70	0.73 ± 0.014	0.59

Table 1: Test accuracy (ACC) and privacy vulnerability (VUL, smaller is better) for $m = 5$ clients and homogeneous local data distributions.

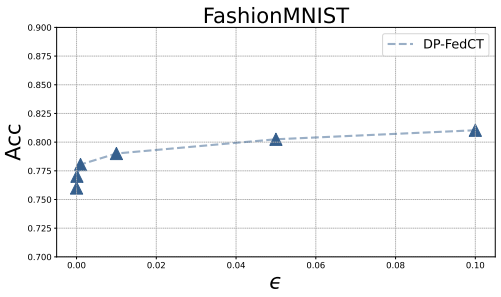

Figure 4: Accuracy (ACC) of DP-FEDCT on the FashionMNIST dataset under different levels of privacy ϵ.

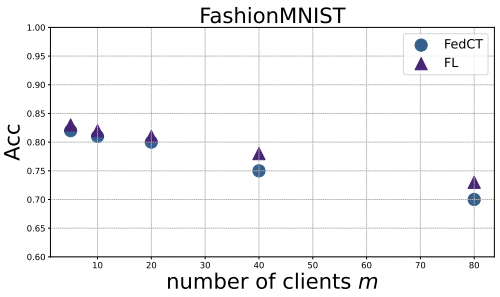

Figure 5: Test accuracy (ACC) of FEDCT and FEDAVG (FL) FashionMNIST with $|U| = 5 \cdot 10^5$ for various numbers of clients m.

networks tend to underestimate the on-average-replace-one stability, leading to vacuous results for generalization (Nagarajan & Kolter, 2019; Petzka et al., 2021), using them to bound sensitivity would underestimate utility. Using an empirical approximation provides a more accurate estimate for the privacy-utility trade-off (Rubinstein & Aldà, 2017). To get this approximation, we apply FEDCT with $m = 5$ clients on the FashionMNIST dataset (Xiao et al., 2017) for various privacy levels ϵ. We estimate the sensitivity of DP-FEDCT by sampling $n = 100$ datasets $D'_1, \ldots, D'_n$ neighboring a local training set D to approximate

$$s_* \approx \max_{i \in [n]} \|f(D) \oplus f(D'_i)\|_F^2 \ ,$$

which yields $s_* \approx 3000$.Using this estimate, Figure 4 shows that DP-FEDCT achieves a high utility in terms of test accuracy even for moderate-to-high privacy levels ϵ with an accuracy of 0.8 for $\epsilon = 0.1$ (without any noise, FEDCT achieves an accuracy of 0.82 in this setup). This hints at a substantially improved privacy-utility trade-off over DP-SGD, which achieves a privacy level of $\epsilon = 145$ with high utility (Xiao et al., 2022). A reason for the good trade-off could lie in the consensus mechanism: for a single unlabeled example $\mu > m/2$ clients predict the majority class; the noise of the XOR-mechanism has to change the predictions of at least $\mu - m/2$ many clients to change the consensus. Note that using the trivial upper bound of $s_*^W = |U| = 5 \cdot 10^4$ instead of the estimate results in a slightly higher epsilon: for a noise level that achieves $\epsilon = 0.1$ with the empirical estimate of s_*, the worst-case bound results in $\epsilon = 0.1 \cdot s_*^W / s_* = 5/3$, instead.

Heterogeneous Data Distributions: In most realistic applications, local datasets are not iid distributed. While this is not the main focus of this work, we show that FEDCT performs similar to FEDAVG for non-pathological non-iid data distributions. We compare FEDCT and FEDAVG on local datasets where half is sampled from a Dirichlet distribution over labels with $\alpha = 2$ (mild heterogeneity) and half from with $\alpha = 100$. The accuracy remains high for both methods with 0.81 for FEDCT and 0.82 for FEDAVG and vulnerability is similar to the iid case with 0.53 for FEDCT and 0.71 for FEDAVG. We observe, however, that the test accuracies of individual clients have greater variance, as shown in Figure 3.

Scalability: We compare the scalability in terms of the number of clients of FEDCT compared to FEDAVG on FashionMNISt, using the same setup as before. We increase the number of clients $m \in \{5, 10, 20, 40, 80\}$ and keep the overall training set size constant, so for larger numbers of clients the local training set size decreases. The results in Figure 5 show that higher levels of distribution

Dataset	DT		RuleFit		XGBoost		Random Forest	
	FEDCT	CENTRALIZED	FEDCT	CENTRALIZED	FEDCT	CENTRALIZED	FEDCT	CENTRALIZED
WineQuality	0.95 ± 0.01	0.92	0.93 ± 0.01	0.95	0.94 ± 0.01	0.94	0.96 ± 0.01	0.98
BreastCancer	0.89 ± 0.01	0.89	0.92 ± 0.01	0.93	0.93 ± 0.01	0.94	0.90 ± 0.02	0.93
AdultsIncome	0.81 ± 0.01	0.82	0.84 ± 0.02	0.85	0.85 ± 0.02	0.87	0.85 ± 0.01	0.86
Mushroom	0.98 ± 0.01	1	0.98 ± 0.02	1	0.98 ± 0.01	1	0.99 ± 0.01	1
Covertype	0.88 ± 0.02	0.94	0.73 ± 0.02	0.76	0.84 ± 0.02	0.87	0.90 ± 0.01	0.95

Table 2: ACC of Interpretable Models.

reduce the accuracy slightly, but both FEDCT and FEDAVG show only a moderate decline, with FEDAVG performing slightly better than FEDCT.

Interpretable Models: A major advantage of FEDCT over FEDAVG and DD is that it allows training interpretable models that do not lend themselves to aggregation. Examples of such models are decision trees, XGBoost, Random Forest, and rule ensembles. For these approaches, no method for aggregating local models exists, so they cannot be trained in a federated setup. To test this, we run FEDCT on the WineQuality (Cortez et al., 2009), Breastcancer (Street et al., 1993), AdultsIncome (Becker & Kohavi, 1996), Mushroom (Bache & Lichman, 1987), and Covertype (Blackard, 1998) datasets with $m = 5$ clients and compare the performance of distributed co-training to pooling all the data and training a model centrally (Centralized). For WineQuality we use $U = 4100$, $U = 370$ for Breastcancer, $U = 10^4$ for AdultsIncome, $U = 4000$ for Mushroom, and $U = 5 \cdot 10^4$ for Covertype. As models, we use classical decision trees, rule ensembles trained via the popular RuleFit (Friedman & Popescu, 2008) algorithm, XGBoost as well as Random Forest. The results in Table 2 show that FEDCT can train interpretable models in a federated learning setup, achieving a model quality comparable to centralized training.

6 DISCUSSION AND CONCLUSION

We propose a semi-supervised, federated co-training approach that collaboratively trains models via sharing predictions. It uses an unlabeled dataset U, producing pseudo-labels L for it by synthesis from the predictions of all local models. Unlabeled data and pseudo-labels form an additional public, shared dataset P that is combined with local data for training. While such an unlabeled dataset is not always available, in many applications, such as healthcare, they are available or can be synthetically generated (El Emam et al., 2020).

FEDCT allows clients to use different models, so that model type or neural network architecture can be tailored to each site's specific needs and characteristics. Exploring such heterogeneous local models, as well as different consensus mechanisms, e.g., staple (Warfield et al., 2004) or averages for regression, makes for excellent future work. Furthermore, investigating client subsampling in FEDCT and its impact on the consensus mechanism, as well as other communication-efficient strategies (e.g., Kamp et al., 2016; 2019) is interesting. To ensure a meaningful consensus, the labels produced by local models need to be of sufficient quality, so local datasets should not be too small. Thus, another intriguing question is how strategies to mitigate small datasets in federated learning (Kamp et al., 2023) can be applied.

We showed both theoretically and empirically that FEDCT achieves a model quality comparable to FEDAVG and DD, while improving privacy over both FEDAVG and DD, as well as DP-FEDAVG. Moreover, FEDCT allows us to train interpretable models, such as decision trees, rule of ensembles, XGBoost, and random forest in a federated learning setup.

REFERENCES

Kevin Bache and Moshe Lichman. Mushroom. UCI Machine Learning Repository, 1987. DOI: https://doi.org/10.24432/C5959T.

Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

Ilai Bistritz, Ariana Mann, and Nicholas Bambos. Distributed distillation for on-device learning. *Advances in Neural Information Processing Systems*, 33:22593–22604, 2020.

Jock Blackard. Covertype. UCI Machine Learning Repository, 1998. DOI: https://doi.org/10.24432/C50K5N.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on computational learning theory*, pp. 92–100, 1998.

Hong-You Chen and Wei-Lun Chao. Fedbe: Making bayesian model ensemble applicable to federated learning. In *International Conference on Learning Representations*, 2020.

Huancheng Chen, Haris Vikalo, et al. The best of both worlds: Accurate global and personalized models through federated learning with data-free hyper-knowledge distillation. *arXiv preprint arXiv:2301.08968*, 2023.

Yae Jee Cho, Andre Manoel, Gauri Joshi, Robert Sim, and Dimitrios Dimitriadis. Heterogeneous ensemble knowledge transfer for training large models in federated learning. *arXiv preprint arXiv:2204.12703*, 2022.

Yae Jee Cho, Jianyu Wang, Tarun Chirvolu, and Gauri Joshi. Communication-efficient and model-heterogeneous personalized federated learning via clustered knowledge transfer. *IEEE Journal of Selected Topics in Signal Processing*, 17(1):234–247, 2023.

Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision support systems*, 47(4): 547–553, 2009.

Enmao Diao, Jie Ding, and Vahid Tarokh. Semifl: Semi-supervised federated learning for unlabeled clients with alternate training. *Advances in Neural Information Processing Systems*, 35:17871–17884, 2022.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pp. 265–284. Springer, 2006.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Khaled El Emam, Lucy Mosquera, and Richard Hoptroff. *Practical synthetic data generation: balancing privacy and the broad availability of data*. O'Reilly Media, 2020.

Jerome H Friedman and Bogdan E Popescu. Predictive learning via rule ensembles. *The annals of applied statistics*, pp. 916–954, 2008.

Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

Sohei Itahara, Takayuki Nishio, Yusuke Koda, Masahiro Morikura, and Koji Yamamoto. Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. *IEEE Transactions on Mobile Computing*, 22(1):191–205, 2021.

Tianxi Ji, Pan Li, Emre Yilmaz, Erman Ayday, Yanfang Ye, and Jinyuan Sun. Differentially private binary-and matrix-valued data query: an xor mechanism. *Proceedings of the VLDB Endowment*, 14(5):849–862, 2021.

Michael Kamp. *Black-Box Parallelization for Machine Learning*. PhD thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, Universitäts-und Landesbibliothek Bonn, 2019.

Michael Kamp, Sebastian Bothe, Mario Boley, and Michael Mock. Communication-efficient distributed online learning with kernels. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part II 16*, pp. 805–819. Springer, 2016.

Michael Kamp, Linara Adilova, Joachim Sicking, Fabian Hüger, Peter Schlicht, Tim Wirtz, and Stefan Wrobel. Efficient decentralized deep learning by dynamic model averaging. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*, pp. 393–409. Springer, 2019.

Michael Kamp, Jonas Fischer, and Jilles Vreeken. Federated learning from small datasets. In *The Eleventh International Conference on Learning Representations*, 2023.

Daniel S Kermany, Michael Goldbaum, Wenjia Cai, Carolina CS Valentim, Huiying Liang, Sally L Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *cell*, 172(5):1122–1131, 2018.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research), 2010.

Jaewoo Lee and Chris Clifton. How much is enough? choosing ε for differential privacy. In *Information Security*, pp. 325–340. Springer, 2011.

Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10713–10722, 2021.

Zheng Li and Yang Zhang. Membership leakage in label-only exposures. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 880–895, 2021.

Haowen Lin, Jian Lou, Li Xiong, and Cyrus Shahabi. Semifed: Semi-supervised federated learning with consistency and pseudo-labeling. *arXiv preprint arXiv:2108.09412*, 2021.

Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363, 2020.

Chuan Ma, Jun Li, Ming Ding, Howard H Yang, Feng Shu, Tony QS Quek, and H Vincent Poor. On safeguarding privacy and security in the framework of federated learning. *IEEE network*, 34(4): 242–248, 2020.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Sasi Kumar Murakonda and Reza Shokri. Ml privacy meter: Aiding regulatory compliance by quantifying the privacy risks of machine learning. *arXiv preprint arXiv:2007.09339*, 2020.

Vaishnavh Nagarajan and J Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning, 2011.

Maxence Noble, Aurélien Bellet, and Aymeric Dieuleveut. Differentially private federated learning on heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pp. 10110–10145. PMLR, 2022.

Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

Henning Petzka, Michael Kamp, Linara Adilova, Cristian Sminchisescu, and Mario Boley. Relative flatness and generalization. *Advances in neural information processing systems*, 34:18420–18432, 2021.

Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. *Advances in neural information processing systems*, 28, 2015.

Benjamin IP Rubinstein and Francesco Aldà. Pain-free random differential privacy with sensitivity sampling. In *International Conference on Machine Learning*, pp. 2950–2959. PMLR, 2017.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Michael Short. On binomial quantile and proportion bounds: With applications in engineering and informatics. *Communications in Statistics-Theory and Methods*, 52(12):4183–4199, 2023.

Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. A co-regularization approach to semi-supervised learning with multiple views. In *Proceedings of ICML workshop on learning with multiple views*, volume 2005, pp. 74–79. Citeseer, 2005.

W Nick Street, William H Wolberg, and Olvi L Mangasarian. Nuclear feature extraction for breast tumor diagnosis. In *Biomedical image processing and biomedical visualization*, volume 1905, pp. 861–870. SPIE, 1993.

Cathie Sudlow, John Gallacher, Naomi Allen, Valerie Beral, Paul Burton, John Danesh, Paul Downey, Paul Elliott, Jane Green, Martin Landray, et al. Uk biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS medicine*, 12(3): e1001779, 2015.

Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*, 2019.

Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM workshop on artificial intelligence and security*, pp. 1–11, 2019.

Katrin Ullrich, Michael Kamp, Thomas Gärtner, Martin Vogt, and Stefan Wrobel. Co-regularised support vector regression. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part II 10*, pp. 338–354. Springer, 2017.

Simon K Warfield, Kelly H Zou, and William M Wells. Simultaneous truth and performance level estimation (staple): an algorithm for the validation of image segmentation. *IEEE transactions on medical imaging*, 23(7):903–921, 2004.

Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Hanshen Xiao, Jun Wan, and Srinivas Devadas. Differentially private deep learning with modelmix. *arXiv preprint arXiv:2210.03843*, 2022.

Zhi-Hua Zhou and Ming Li. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on knowledge and Data Engineering*, 17(11):1529–1541, 2005.

Ligeng Zhu and Song Han. Deep leakage from gradients. In *Federated learning*, pp. 17–31. Springer, 2020.

Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International conference on machine learning*, pp. 12878–12889. PMLR, 2021.

A PROOF OF PROPOSITION 1

For convenience, we restate the proposition.

Proposition 1. *For $m \geq 3$ clients with local datasets $D^1, \ldots, D^m$ and unlabeled dataset U drawn iid from $\mathcal{D}$, let $\mathcal{A}^i$ for $i \in [m]$ be a set of learning algorithms that all achieve a linearly increasing training accuracy a_t for all labelings of U, i.e., there exists $c \in \mathbb{R}_+$ such that $a_t \geq 1 - c/t$, then there exists $t_0 \in \mathbb{N}$ such that $a_t \geq 1/2$ and* FEDCT *with majority vote converges with probability $1 - \delta$, where*

$$\delta \leq |U|(4c)^{\frac{m}{2}} \zeta\left(\frac{m}{2}, t_0 + 1\right)$$

and $\zeta(x, q)$ is the Hurwitz zeta function.

Proof. Let P_t denote the consensus label at time $t \in \mathbb{N}$. We first show that the probability δ_t of $P_t \neq P_{t-1}$ is bounded. Since the learning algorithm $\mathcal{A}$ at time $t \geq t_0$ achieves a training accuracy $a_t \geq 0.5$, the probability can be determined via the CDF of the binomial distribution, i.e.,

$$\begin{aligned}
\delta_t =& \mathbb{P}\left(\exists u \in U : \sum_{i=1}^{m} \mathbb{1}_{h_t^i(u)=v} < \left\lfloor \frac{m}{2} \right\rfloor\right) \\
=& F\left(\left\lfloor \frac{m}{2} \right\rfloor - 1, m, a_t\right) = \sum_{i=1}^{\left\lfloor \frac{m}{2} \right\rfloor - 1} \binom{m}{i} a_t^i (1 - a_t)^{m-i} \quad .
\end{aligned}$$

Applying the Chernoff bound and denoting by $D(\cdot\|\cdot)$ the Kullback-Leibler divergence yields

$$\begin{aligned}
\delta_t \leq& \exp\left(-mD\left(\frac{\left\lfloor \frac{m}{2} \right\rfloor - 1}{m} \middle\| a_t\right)^2\right) \\
=& \exp\left(-m\left(\frac{\left\lfloor \frac{m}{2} \right\rfloor - 1}{m} \log \frac{\frac{\left\lfloor \frac{m}{2} \right\rfloor - 1}{m}}{a_t} + \left(1 - \frac{\left\lfloor \frac{m}{2} \right\rfloor - 1}{m}\right) \log \frac{1 - \frac{\left\lfloor \frac{m}{2} \right\rfloor - 1}{m}}{1 - a_t}\right)\right) \\
\leq& \exp\left(-m\left(\frac{\frac{m}{2}}{m} \log \frac{\frac{\frac{m}{2}}{m}}{a_t} + \left(1 - \frac{\frac{m}{2}}{m}\right) \log \frac{1 - \frac{\frac{m}{2}}{m}}{1 - a_t}\right)\right) \\
=& \exp\left(-m\left(\frac{1}{2} \log \frac{\frac{1}{2}}{a_t} + \frac{1}{2} \log \frac{\frac{1}{2}}{1 - a_t}\right)\right) = \exp\left(-\frac{m}{2} \log \frac{1}{2a_t} - \frac{m}{2} \log \frac{1}{2(1 - a_t)}\right) \\
=& \exp\left(\frac{m}{2}\left(\log 2a_t + \log 2(1 - a_t)\right)\right) = (2a_t)^{\frac{m}{2}} \left(2(1 - a_t)\right)^{\frac{m}{2}} = 4^{\frac{m}{2}} a_t^{\frac{m}{2}} (1 - a_t)^{\frac{m}{2}} \quad .
\end{aligned}$$

The union bound over all $u \in U$ yields

$$\delta_t \leq |U| 4^{\frac{m}{2}} a_t^{\frac{m}{2}} (1 - a_t)^{\frac{m}{2}} \quad .$$

To show convergence, we need to show that for $t_0 \in \mathbb{N}$ it holds that

$$\sum_{t=t_0}^{\infty} \delta_t \leq \delta$$

for $0 \leq \delta < 1$. Since we assume that a_t grows linearly, we can write wlog. $a_t = 1 - c/t$ for some $c \in \mathbb{R}_+$ and $t \geq 2c$. With this, the sum can be written as

$$\begin{aligned}
\sum_{t=t_0}^{\infty} \delta_t \leq& |U| \sum_{t=t_0}^{\infty} 4^{\frac{m}{2}} \left(1 - \frac{c}{t}\right)^{\frac{m}{2}} \left(\frac{c}{t}\right)^{\frac{m}{2}} = |U| 4^{\frac{m}{2}} \sum_{t=t_0}^{\infty} \left(\frac{\frac{t}{c} - 1}{\frac{t^2}{c^2}}\right)^{\frac{m}{2}} \\
\leq& |U| 4^{\frac{m}{2}} \sum_{t=t_0}^{\infty} \left(\frac{\frac{t}{c}}{\frac{t^2}{c^2}}\right)^{\frac{m}{2}} = (4c)^{\frac{m}{2}} \sum_{t=t_0}^{\infty} \left(\frac{1}{t}\right)^{\frac{m}{2}} = |U|(4c)^{\frac{m}{2}} \zeta\left(\frac{m}{2}\right) - H_{t_0}^{\left(\frac{m}{2}\right)} \quad , \quad ,
\end{aligned}$$

where $\zeta(x)$ is the Riemann zeta function and $H_n^{(x)}$ is the generalized harmonic number. Note that $H_n^{(x)} = \zeta(x) - \zeta(x, n+1)$, where $\zeta(x, q)$ is the Hurwitz zeta function, so that this expression can be simplified to

$$\sum_{t=t_0}^{\infty} \delta_t \leq |U|(4c)^{\frac{m}{2}} \zeta\left(\frac{m}{2}\right) - \zeta\left(\frac{m}{2}\right) + \zeta\left(\frac{m}{2}, t_0 + 1\right) = |U|(4c)^{\frac{m}{2}} \zeta\left(\frac{m}{2}, t_0 + 1\right) \quad .$$

□

B Proof of Proposition 2

For convenience, we restate the proposition.

Proposition 2. *For classification models $h : \mathcal{X} \to \mathcal{Y}$, let ℓ be a loss function that upper bounds the $0-1$-loss and $\mathcal{A}$ a learning algorithm that is on-average-leave-one-out stable with stability rate $\epsilon(m)$ for ℓ. Let $D \cup U$ be a local training set with $|U| = n$, and $\delta \in (0, 1)$. Then with probability $1 - \delta$, the sensitivity s_* of $\mathcal{A}$ on U is bounded by*

$$s_* \leq \left\lceil n\epsilon(n) + P\sqrt{n\epsilon(n)(1-\epsilon(n))} + \frac{P^2}{3} \right\rceil \quad ,$$

where $P = \Phi^{-1}(1-\delta)$ with $= \Phi^{-1}$ being the probit function.

Proof. The sensitivity s_* is defined as the supremum of the Frobenius norm of the symmetric difference between the predictions on the unlabeled dataset U for two models h_s and h'_s trained on datasets s and s' that differ by one instance.

$$s_* = \sup_{S,S'} \|h_S(U) \Delta h_{S'}(U)\|_F$$

Since $\mathcal{A}$ is on-average-replace-one stable with rate ϵ for ℓ and ℓ upper bounds the $0-1$-loss, $\mathcal{A}$ is on-average-replace-one stable with rate at most ϵ for the $0-1$-loss. Thus, the expected change in loss on a single element of the training set is bounded by $\epsilon(|D \cup U|)$. Since the $0-1$-loss is either 0 or 1, this can be interpreted as a success probability in a Bernoulli process. The expected number of differences on the unlabeled dataset then is the expected value of the corresponding binomial distribution, i.e., $|U|\epsilon(|D \cup U|) \leq |U|\epsilon(|U|)$. We are interested in the maximum number of successes such that the cumulative distribution function of the binomial distribution is smaller than $1 - \delta$. This threshold k can be found using the quantile function (inverse CDF) for which, however, no closed form exists. Short (2023) has shown that the quantile function $Q(n, p, R)$ can be bounded by

$$Q(n, p, R) \leq \left\lceil np + \Phi^{-1}(R)\sqrt{np(1-p)} + \frac{\Phi^{-1}(R)^2}{3} \right\rceil \quad ,$$

where Φ^{-1} is the probit function (inverse of standard normal's cdf). With $n = |u|$, $p = \epsilon(|U|)$, and $R = 1 - \delta$, the number of differences in predictions on the unlabeled dataset, i.e., the sensitivity s_*, is upper bounded by

$$s_* \leq \left\lceil |U|\epsilon(|U|) + \Phi^{-1}(1-\delta)\sqrt{|U|\epsilon(|U|)(1-\epsilon(|U|))} + \frac{\Phi^{-1}(1-\delta)^2}{3} \right\rceil$$

with probability $1 - \delta$.

□

C Additional Empirical Evaluation

C.1 Mixed Model Types

Sharing hard labels allows us to train any supervised learning method on each client. That allows us to even use different models for different clients in FedCT. To demonstrate this, we compare using the best performing interpretable model on the BreatCancer dataset (XGBoost) on every client to two heterogeneous ensembles using decision trees (DT), random forests (RF), rule ensembles (RuleFit), gradient-boosted decision trees (XGBoost), and neural networks (MLP). The results in Table 3 show that using a diverse ensemble of models can further improve accuracy.

Dataset	C1	C2	C3	C4	C5	ACC
BreastCancer	DT	RF	RuleFit	XGBoost	RF	0.95
BreastCancer	DT	MLP	RuleFit	XGBoost	RF	0.93
BreastCancer	XGBoost	XGBoost	XGBoost	XGBoost	XGBoost	0.94

Table 3: Mixed model experiment

C.2 Comparison to PATE

PATE (Papernot et al., 2016) is a distillation algorithm with the goal of producing a single student model from an ensemble of teachers. To protect the private dataset the teachers have been trained upon, a Laplace mechanism is applied to the consensus - more precisely, the prediction counts. Thereby, a curious student cannot infer upon the private training data. In both PATE and FedCT, hard labels are used to form a consensus. PATE, however, is not a collaborative training method and is not concerned with protecting the output of teachers against the entity that produces the consensus (in our case, this would be an honest-but-curious server). Since both FedCT and PATE share hard labels, it is interesting to evaluate the benefits of collaborative training over distillation - despite their difference in privacy protection. For that, we instantiate PATE with each client being a teacher where a model is trained to convergence. The predictions of the teachers on the unlabeled dataset are then used to form consensus labels with which a single student model is trained to convergence. In Table 4 we report the results with $m = 5$ and $m = 100$ clients/teachers. Indeed, collaborative training (FedCT) achieves substantially higher test accuracy than distillation (PATE), which is unsurprising since in collaborative training, the consensus is used to iteratively improve the client models, whereas in distillation the teachers are only trained once.

m = 5		
Dataset	**FedCT**	**PATE**
FashionMNIST	0.7658	0.6451
CIFAR10	0.7608	0.7039
Pneumonia	0.7478	0.7208
MRI	0.6274	0.6038
SVHN	0.8805	0.8721
m = 100		
FashionMNIST	0.7154	0.6318
Pneumonia	0.7269	0.6903

Table 4: PATE Vs FedCT on m=5 clients and on m=100 clients

C.3 Additional Results on Scalability

In addition to our results on the scalability of FedCT wrt. the number of clients on the Fashion-MNISTdataset in paragraph 5, we here report the test accuracy wrt. the number of clients on the Pneumonia dataset. The results in Figure 6 show that also on this dataset FedCT scales as well as FedAvg with the number of clients.

C.4 Additional Results on Data heterogeneity

In our Previous experiment on heterogeneous data distributions using the FashionMINST dataset, we sampled 10% of the local training data rather homogeneously wrt. the labels using a Dirichlet distribution with $\alpha_1 = 100$, and the remaining 90% mildly heterogeneous with $\alpha_2 = 2$. In this section, increase the heterogeneity using $\alpha_2 = 0.01$ on the heterogeneous part. The results in Table 5 show that FedCT achieved comparable accuracy to FedAvg and DD in this scenario as well—the convergence behavior for all local clients is shown in Figure 7.

We conjecture that as long as clients achieve a minimum performance on all data, a meaningful consensus can be formed and label sharing (both hard and soft labels) works well. To test this, we investigate the performance on a pathological distribution (i.e., all data drawn with $\alpha = 0.01$ and clients only observe a small subset of labels). Here, the performance of all methods decreases, but

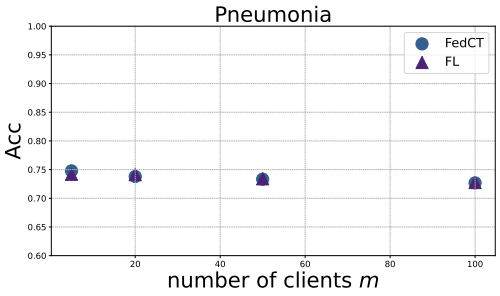

Figure 6: Test accuracy (ACC) of FEDCT and FEDAVG (FL) on Pneumonia with $|U| = 200$ for various numbers of clients m.

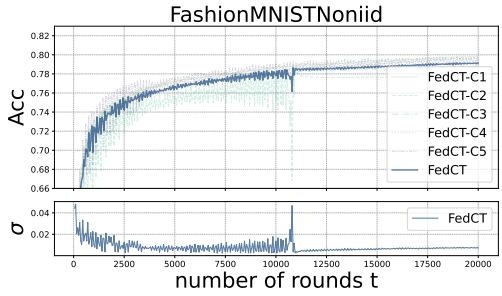

Figure 7: **Top**: Test accuracy (ACC) over time for $m = 5$ local models of FEDCT on heterogeneous distribution for the FashionMNIST dataset. **Bottom**: Standard deviation of test accuracy of local models in FEDCT.

label sharing approaches (both DD and FedCT) perform substantially worse than model parameters sharing (FedAvg), supporting our conjecture.

Dirichlet α	**FedCT**	**FedAvg**	**DD**
$\alpha_1 = 100, \alpha_2 = 0.01$	0.7981	0.7907	0.79034
$\alpha_1 = \alpha_2 = 0.01$	0.3663	0.7339	0.3585

Table 5: Average test accuracy ACC of FEDCT, FEDAVG, and DD

We evaluate the performance of FEDCT on one more data heterogeneity scenario proposed by Li & Wang (2019). In this scenario, a private labeled dataset and a public labeled dataset are used, where each instance has two labels: a fine-grained class label, and a more coarse superclass label. An example of such a dataset is CIFAR100, where the 100 classes fall into 20 superclasses. Data heterogeneity is achieved by homogeneously distributing superclasses over clients, but in such a way that each client only observes a single class per superclass. This means, while all clients observe vehicles, some only observe cars, others only bicycles. The classification task is to predict the superclass. This should ensure that a meaningful consensus can be achieved. We compare FEDCT to the method Li & Wang (2019) propose (FedMD), although Li & Wang (2019) assume a labeled public dataset (not an unlabeled one). In this scenario, FEDCT achieves an average test accuracy acc of 0.5106, slightly outperforming FedMD which achieves an average accuracy of 0.5. Note that this comparison is biased in favor of FedMD, since it uses transfer learning on the labeled public dataset which FEDCT does not.

C.5 EFFECT OF UNLABELED DATASET SIZE $|U|$

Since FEDCT utilizes a public unlabeled dataset, we evaluated the performance of FedCT under different unlabeled dataset sizes U The evaluation has been done on the Pneumonia dataset where we fixed the local training data set size to 100 examples. Our results in Figure 8 show that increasing the unlabeled data set size substantially improves FEDCT accuracy.

D DETAILED DISCUSSION OF RELATED WORK

The main goal of FEDCT is to improve the privacy of current federated learning approaches while maintaining model quality. For that, we consider a classical FL scenario where clients hold a private local dataset. We additionally assume that they have access to a public unlabeled dataset and the client's aim is to train models collaboratively without sharing either data or model parameters. To improve privacy over existing methods, FEDCT shares hard labels instead of model parameters or soft labels. Our empirical evaluation shows that sharing hard labels indeed improves privacy substantially, both over model parameters and soft label sharing. In Table 6 We summarize the main differences between sharing hard labels, soft labels, and model parameters. Sharing hard labels not

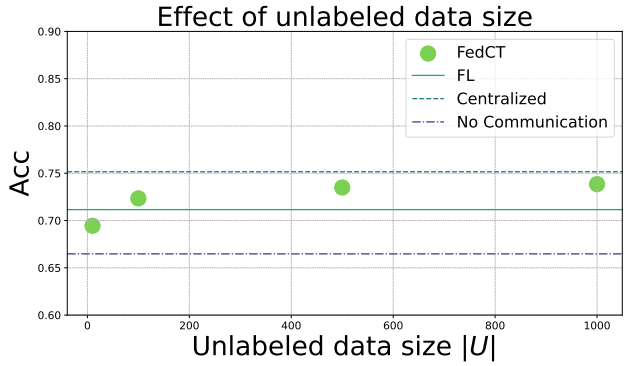

Figure 8: Test accuracy (ACC) of FEDCT under different unlabeled dataset size U.

only reveals less information about the local private data but also allows training interpretable models collaboratively.

There is a wide range of semi-supervised federated learning methods that do not fit our scenario. For example, FedMD (Li & Wang, 2019) uses a public labeled data. Fed-ET (Cho et al., 2022), Moon (Li et al., 2021), semiFed(Lin et al., 2021), SemiFL(Diao et al., 2022), FedGen(Zhu et al., 2021), FedHKD(Chen et al., 2023) require clients to share their model parameters with the server, therefore, do not improve privacy over our baseline FEDAVG (McMahan et al., 2017). Furthermore, Moon assumes that the unlabeled dataset is only accessible by the server. Cho et al. (2023) uses co-regularization for personalized federated learning, so we naturally compare it to the non-personalized variant distributed distillation. Itahara et al. (2021) shares soft labels that are similar to our baseline distributed distillation DD (Bistritz et al., 2020).

Shared Information	Model Quality IID	Model Quality Non-IID	Privacy	Interpretable Models
Model Parameters	++	++	-	-
Soft Labels	++	+	+	-
Hard Labels (FedCT)	++	+	++	++

Table 6: Comparison of parameter, soft label, and hard label sharing in federated learning.

E DETAILS ON EXPERIMENTS

E.1 DETAILS ON PRIVACY VULNERABILITY EXPERIMENTS

We measure privacy vulnerability by performing membership inference attacks against FEDCT and FEDAVG In both attacks, the attacker creates an attack model using a model it constructs from its training and test datasets. Similar to previous work Shokri et al. (2017), we assume that the training data of the attacker has a similar distribution to the training data of the client. Once the attacker has its attack model, it uses this model for membership inference. In blackbox attacks (in which the attacker does not have access to intermediate model parameters), it only uses the classification scores it receives from the target model (i.e., client's model) for membership inference. On the other hand, in whitebox attacks (in which the attacker can observe the intermediate model parameters), it can use additional information in its attack model. Since the proposed FEDCT does not reveal intermediate model parameters to any party, it is only subject to blackbox attacks. Vanilla federated learning on the other hand is subject to whitebox attacks. Each inference attack produces a membership score of a queried data point, indicating the likelihood of the data point being a member of the training set. We measure the success of membership inference as ROC AUC of these scores. The **vulnerability (VUL)** of a method is the ROC AUC of membership attacks over K runs over the entire training set (also called attack epochs) according to the attack model and scenario. A vulnerability of 1.0 means that membership can be inferred with certainty, whereas 0.5 means that deciding on membership is a random guess.

We assume the following attack model: clients are honest and the server may be semi-honest (follow the protocol execution correctly, but it may try to infer sensitive information about the clients). The main goal of a semi-honest server is to infer sensitive information about the local training data of the clients. This is a stronger attacker assumption compared to a semi-honest client since the server receives the most amount of information from the clients during the protocol, and a potential semi-honest client can only obtain indirect information about the other clients. We also assume that parties do not collude.

The attack scenario for FEDCT and DD is that the attacker can send a (forged) unlabeled dataset to the clients and observe their predictions, equivalent to one attack epoch ($K = 1$); the one for FEDAVG and DP-FEDAVG is that the attacker receives model parameters and can run an arbitrary number of attacks—we use $K = 500$ attack epochs.

E.2 DATASETS

We use 3 standard image classification datasets: CIFAR10 (Krizhevsky et al., 2010), Fashion-MNIST (Xiao et al., 2017), and SVHN (Netzer et al., 2011). We describe the datasets and our preprocessing briefly.

CIFAR10 consists of 50 000 training and 10 000 test 32×32 color images in 10 classes with equal distribution (i.e., a total of 6 000 images per class). Images are normalized to zero mean and unit variance. *FashionMNIST* consists of 60 000 training and 10 000 test 28×28 grayscale images of clothing items in 10 classes with equal distribution. Images are not normalized. *SVHN* (Street View House Numbers) consists of 630 420 32×32 color images of digits from house numbers in Google Street View, i.e., 10 classes. The datasest is partitioned into 73 257 for training, 26 032 for testing, and 531 131 additional training images. In our experiments, we use only the training and testing set. Images are not normalized.

We use five standard datasets from the UCI Machine Learning repository for our experiments on collaboratively training interpretable models: WineQuality (Cortez et al., 2009), BreastCancer (Sudlow et al., 2015), AdultsIncome (Becker & Kohavi, 1996), Mushroom (Bache & Lichman, 1987), and Covertype (Blackard, 1998). A short description of the five datasets follows. *WineQuality* is a tabular dataset of 6 497 instances of wine with 11 features describing the wine (e.g., alcohol content, acidity, pH, and sulfur dioxide levels) and the label is a wine quality score from 0 to 10. We remove duplicate rows and transform the categorial type attribute to a numerical value. We then normalize all features to zero mean and unit variance. *BreastCancer* is a medical diagnostics tabular dataset with 569 instances of breast cell samples with 30 features describing cell nuclei with 2 classes (malignant and benign). We followed the same preprocessing steps as WineQuality dataset. *AdultIncome* is a tabular dataset with $48,842$ instances of adults from various backgrounds with 14 features describing attributes such as age, work class, education, marital status, occupation, relationship, race, gender, etc. The dataset is used to predict whether an individual earns more than $50,000$\$ a year, leading to two classes: income more than $50,000$\$, and income less than or equal to $50,000$\$. *Mushroom* is a biological tabular dataset with 8124 instances of mushroom samples with 22 features describing physical characteristics such as cap shape, cap surface, cap color, bruises, odor, gill attachment, etc. The dataset is used to classify mushrooms as edible or poisonous, leading to two classes: edible and poisonous. *Covertype* is an environmental tabular dataset with $581,012$ instances of forested areas with 54 features describing geographical and cartographical variables, such as elevation, aspect, slope, horizontal distance to hydrology, vertical distance to hydrology, horizontal distance to roadways, hillshade indices, and wilderness areas and soil type binary indicators. The dataset is used to

Dataset	training size	testing size	unlabeled size $\lvert U \rvert$	communication period b	number of rounds T
CIFAR10	$40 \cdot 10^3$	$10 \cdot 10^3$	$10 \cdot 10^3$	10	$3 \cdot 10^3$
FashionMNIST	$10 \cdot 10^3$	$10 \cdot 10^3$	$50 \cdot 10^3$	50	$20 \cdot 10^3$
Pneumonia	4386	624	900	20	$20 \cdot 10^3$
MRI	30	53	170	6	$2 \cdot 10^3$
SVHN	38 257	26 032	$35 \cdot 10^3$	10	$20 \cdot 10^3$

Table 7: Dataset descriptions for image classification experiments.

Layer	Output Shape	Activation	Parameters
Conv2D	(32, 32, 32)	ReLU	896
BatchNormalization	(32, 32, 32)	-	128
Conv2D	(32, 32, 32)	ReLU	9248
BatchNormalization	(32, 32, 32)	-	128
MaxPooling2D	(16, 16, 32)	-	-
Dropout	(16, 16, 32)	-	-
Conv2D	(16, 16, 64)	ReLU	18496
BatchNormalization	(16, 16, 64)	-	256
Conv2D	(16, 16, 64)	ReLU	36928
BatchNormalization	(16, 16, 64)	-	256
MaxPooling2D	(8, 8, 64)	-	-
Dropout	(8, 8, 64)	-	-
Conv2D	(8, 8, 128)	ReLU	73856
BatchNormalization	(8, 8, 128)	-	512
Conv2D	(8, 8, 128)	ReLU	147584
BatchNormalization	(8, 8, 128)	-	512
MaxPooling2D	(4, 4, 128)	-	-
Dropout	(4, 4, 128)	-	-
Flatten	(2048,)	-	-
Dense	(128,)	ReLU	262272
BatchNormalization	(128,)	-	512
Dropout	(128,)	-	-
Dense	(10,)	Linear	1290

Table 8: CIFAR10 architecture

predict forest cover type, leading to 7 distinct classes: Spruce/Fir, Lodgepole Pine, Ponderosa Pine, Cottonwood/Willow, Aspen, Douglas-fir, and Krummholz.

Furthermore, we use 2 medical image classification datasets, Pneumonia (Kermany et al., 2018), and MRI[3]. *Pneumonia* consists of 5 286 training and 624 test chest x-rays with labels *normal*, *viral pneumonia*, and *bacterial pneumonia*. We simplify the labels to *healthy* and *pneumonia* with a class imbalance of roughly 3 pneumonia to 1 healthy. The original images in the Pneumonia dataset do not have a fixed resolution as they are sourced from various clinical settings and different acquisition devices. We resize all images to a resolution of 224×224 pixels without normalization. *MRI* consists of 253 MRI brain scans with a class imbalance of approximately 1.5 brain tumor scans to 1 healthy scan. Out of the total 253 images, we use 53 images as testing set. Similar to the pneumonia dataset, the original images have no fixed resolution and are thus resized to 150×150 without normalization.

E.3 Experimental Setup

We now describe the details of the experimental setup used in our empirical evaluation.

In our privacy-utility trade-off experiments, we use $m = 5$ clients for all datasets. We report the split into training, test, and unlabeled dataset per dataset, as well as the used communication period b and number of rounds T in Table 7. For the scalability experiments, we use the same setup, varying $m \in \{5, 10, 20, 40, 80\}$ clients. For the experiments on heterogeneous data distributions, we use the same setup as for the privacy-utility trade-off, but we sample the local dataset from a Dirichlet distribution as described in the main text.

For all experiments, we use Adam as an optimization algorithm with a learning rate 0.01 for CIFAR10, and 0.001 for the remaining datasets. A description of the DNN architecture for each dataset follows.

The neural network architectures used for each dataset are given in the following. For CIFAR10 we use a CNN with multiple convolutional layers with batch normalization and max pooling. The details of the architecture are described in Table 8. For FashionMNIST, we use a simple feed forward architecture on the flattened input. The details of the architecture are described in Table 9. For

[3] https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection

Layer	Output Shape	Activation	Parameters
Flatten	(784,)	-	-
Linear	(784, 512)	-	401,920
ReLU	(512,)	ReLU	-
Linear	(512, 512)	-	262,656
ReLU	(512,)	ReLU	-
Linear	(512, 10)	-	5,130

Table 9: FashionMNIST architecture

Pneumonia, we use a simple CNN, again with batch normalization and max pooling, with details given in Table 10. For MRI we use an architecture similar to pneumonia with details described in

Layer	Output Shape	Activation	Parameters
Conv2d	(3, 32, 32)	-	896
BatchNorm2d	(32, 32, 32)	-	64
Conv2d	(32, 32, 32)	-	18,464
BatchNorm2d	(64, 32, 32)	-	128
MaxPool2d	(64, 16, 16)	-	-
Conv2d	(64, 16, 16)	-	36,928
BatchNorm2d	(64, 16, 16)	-	128
MaxPool2d	(64, 8, 8)	-	-
Flatten	(4096,)	-	-
Linear	(2,)	-	4,194,306

Table 10: Pneumonia architecture

Table 11. For SVHN, we use again a standard CNN with batch normalization and max pooling,

Layer	Output Shape	Activation	Parameters
Conv2d	(3, 32, 32)	-	896
BatchNorm2d	(32, 32, 32)	-	64
Conv2d	(32, 32, 32)	-	18,464
BatchNorm2d	(64, 32, 32)	-	128
MaxPool2d	(64, 16, 16)	-	-
Conv2d	(64, 16, 16)	-	36,928
BatchNorm2d	(64, 16, 16)	-	128
MaxPool2d	(64, 8, 8)	-	-
Flatten	(32768,)	-	-
Linear	(2,)	-	2,636,034

Table 11: MRI architecture

detailed in Table 12.

For our experiments on interpretable models, we use $m = 5$ clients. For decision trees (DT), we split by the Gini index with at least 2 samples for splitting. For RuleFit, we use a tree size of 4 and a maximum number of rules of 200. For the WineQuality dataset, we use an unlabeled dataset size of $U = 4100$, a training set size of 136, and a test set size of 1059. For BreastCancer, we use an unlabeled dataset of size $U = 370$, a training set of size 85, and a test set of size 114. For the AdultsIncome dataset, we use an unlabeled dataset of size $U = 10^4$, a training set of size $31,073$, and a test set of size 7769. For the Mushroom dataset, we use an unlabeled dataset of size $U = 4000$, a training set of size 2499, and a test set of size 1625. For the covertype dataset, we use an unlabeled dataset of size $U = 5 \cdot 10^4$, a training set of size $414,810$, and a test set of size $116,202$.

Layer	Output Shape	Parameters
Conv2d	(3, 32, 32)	896
BatchNorm2d	(32, 32, 32)	64
Conv2d	(32, 32, 32)	9,248
MaxPool2d	(32, 16, 16)	-
Dropout2d	(32, 16, 16)	-
Conv2d	(32, 16, 16)	18,464
BatchNorm2d	(64, 16, 16)	128
Conv2d	(64, 16, 16)	36,928
MaxPool2d	(64, 8, 8)	-
Dropout2d	(64, 8, 8)	-
Conv2d	(64, 8, 8)	73,856
BatchNorm2d	(128, 8, 8)	256
Conv2d	(128, 8, 8)	147,584
MaxPool2d	(128, 4, 4)	-
Dropout2d	(128, 4, 4)	-
Flatten	(2048,)	-
Linear	(128,)	262,272
Dropout	(128,)	-
Linear	(10,)	1,290

Table 12: SVHN architecture