FEDAUX: Leveraging Unlabeled Auxiliary Data in Federated Learning

Felix Sattler, Tim Korjakow, Roman Rischke, and Wojciech Samek, *Member, IEEE*

***Abstract*— Federated distillation (FD) is a popular novel algorithmic paradigm for Federated learning (FL), which achieves training performance competitive to prior parameter averaging-based methods, while additionally allowing the clients to train different model architectures, by distilling the client predictions on an unlabeled auxiliary set of data into a student model. In this work, we propose FEDAUX, an extension to FD, which, under the same set of assumptions, drastically improves the performance by deriving maximum utility from the unlabeled auxiliary data. FEDAUX modifies the FD training procedure in two ways: First, unsupervised pre-training on the auxiliary data is performed to find a suitable model initialization for the distributed training. Second, (ε, δ)-differentially private certainty scoring is used to weight the ensemble predictions on the auxiliary data according to the certainty of each client model. Experiments on large-scale convolutional neural networks (CNNs) and transformer models demonstrate that our proposed method achieves remarkable performance improvements over state-of-the-art FL methods, without adding appreciable computation, communication, or privacy cost. For instance, when training ResNet8 on non-independent identically distributed (i.i.d.) subsets of CIFAR10, FEDAUX raises the maximum achieved validation accuracy from 30.4% to 78.1%, further closing the gap to centralized training performance. Code is available at https://github.com/fedl-repo/fedaux.**

***Index Terms*— Certainty-weighted aggregation, differential privacy (DP), federated distillation (FD), federated learning (FL), unsupervised pre-training.**

I. INTRODUCTION

FEDERATED learning (FL) allows distributed entities ("clients") to jointly train (deep) machine learning models on their combined local data, without having to transfer this data to a centralized location [1]. The Federated training process is conducted over multiple communication rounds, where, in each round, a central server aggregates the training state of the participating learners, for instance, via a parameter averaging operation. Since local training data never leaves the participating devices, FL can drastically improve privacy [2]–[4], ownership rights [5], and security [6] for the participants. As the number of mobile and IoT devices and their capacities to collect and process large amounts of high-quality and privacy-sensitive data steadily grows, Federated training procedures become increasingly relevant.

While the client data in FL is typically assumed to be private, in many real-world applications, the server additionally has access to unlabeled auxiliary data, which roughly matches the distribution of the client data. For instance, for many Federated computer vision and natural language processing problems, such auxiliary data can be given in the form of public databases such as ImageNet [7] or WikiText [8]. These databases contain millions to billions of data samples but are typically lacking the necessary label information to be useful for training task-specific models.

Recently, Federated distillation (FD), a novel algorithmic paradigm for FL problems where such auxiliary data is available, was proposed. In contrast to classic parameter averaging-based FL algorithms [1], [9]–[12], which require all client's models to have the same size and structure, FD allows the clients to train heterogeneous model architectures, by distilling the client predictions on the auxiliary set of data into a student model. This can be particularly beneficial in situations where clients are running on heterogeneous hardware and recent studies show that FD-based training also has favorable communication properties [13], [14] and can outperform parameter averaging-based FL algorithms [15].

However, just like for their parameter-averaging-based counterparts, the performance of FD-based learning algorithms falls short of centralized training and deteriorates quickly if the training data is distributed in a heterogeneous ["non-independent identically distributed (i.i.d.")] way among the clients.

In this work, we aim to further close this performance gap, by exploring the core assumption of FD-based training and deriving maximum utility from the available unlabeled auxiliary data. Our main contributions are as follows.

1) We show that a wide range of (out-of-distribution) auxiliary datasets are suitable for self-supervised pre-training and can drastically improve FL performance across all levels of data heterogeneity.
2) We propose a novel certainty-weighted FD technique, which improves the performance of FD on non-i.i.d. data substantially, by exploiting the available auxiliary data, addressing a long-standing problem in FL research.
3) We derive an (ε, δ)-differentially private mechanism to constrain the privacy loss associated with transmitting certainty scores.

Manuscript received May 28, 2021; revised September 20, 2021; accepted November 16, 2021. This work was supported in part by the German Federal Ministry of Education and Research (BMBF) through the Berlin Institute for the Foundations of Learning and Data (BIFOLD) under Grant 01IS18025A and Grant 01IS18037I and in part by the EU's Horizon 2020 Project COPA EUROPE under Grant 957059. *(Corresponding author: Wojciech Samek.)*

The authors are with the Department of Artificial Intelligence, Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany (e-mail: wojciech.samek@hhi.fraunhofer.de).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TNNLS.2021.3129371.

Digital Object Identifier 10.1109/TNNLS.2021.3129371

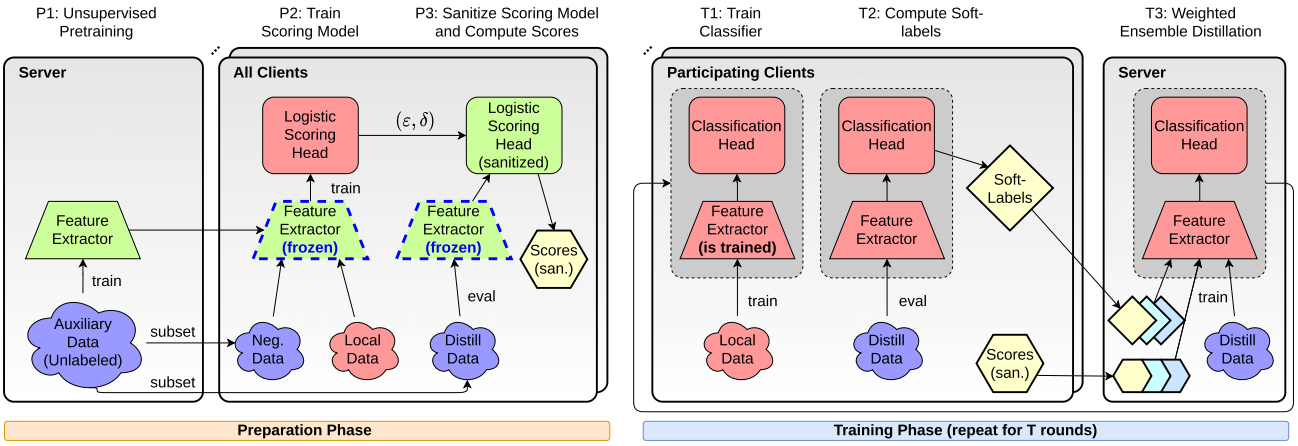


Fig. 1. FL procedure of FEDAUX is organized in a preparation and a training phase: Preparation phase: P1) The unlabeled auxiliary data is used to pre-train a feature extractor (e.g., using contrastive representation learning). P2) The feature-extractor is sent to the clients, where it is used to initialize the client models. Based on the extracted features, a logistic scoring head is trained to distinguish local client data from a subset of the auxiliary data. P3) The trained scoring head is sanitized using a (ε, δ)-differentially private mechanism and then used to compute (differentially private) certainty scores on the distillation data. Training phase: T1) In each communication round, a subset of the client population is selected for training. Each selected client downloads a model initialization from the server and then updates the full model f_i (feature extractor and scoring head) using their private local data. T2) The locally trained classifier and scoring models f_i and s_i are sent to the server, where they are combined into a weighted ensemble. T3) Using the unlabeled auxiliary data and the weighted ensemble as a teacher, the server distills a student model which is used as the initialization point for the next round of Federated training. Note that, in practice, it is more practical to perform computation of soft labels and scores at the server to save client resources.

4) We extensively evaluate our new method on a wide variety of Federated image and text classification problems, using large-scale convolutional neural networks (CNNs) and transformer models.

Notably, as we will see, the observed significant performance improvements achieved by FEDAUX are possible: 1) under the same assumptions made in the FD literature; 2) with only negligible additional computational overhead for the resource-constrained clients; and 3) with small quantifiable excess privacy loss.

The remainder of this manuscript is organized as follows: In Section II, we give an introduction to FD and clearly state our assumptions on the FL setting. In Section III, we describe the components of our proposed FEDAUX algorithm, namely unsupervised pre-training and weighted ensemble distillation and derive an (ε, δ)-differentially private mechanism to obfuscate the ensemble weights. In Section IV, we provide the detailed algorithm for the general FL setting where clients may locally train different model architectures. In Section V, we give an overview over the current state of research in FD as well as FL in the presence of unlabeled auxiliary data, in general. In Section VI, we perform extensive numerical studies evaluating the performance, privacy properties, and sensitivity to auxiliary data of FEDAUX against several important baseline methods in a variety of different FL scenarios, including resource constrained settings. In Section VII, we complement these quantitative results with a qualitative analysis of our method, before concluding in Section VIII.

II. FEDERATED DISTILLATION

We assume the conventional FL setting, where a population of n clients is holding potentially non-i.i.d. subsets of private labeled data $D_1, \ldots, D_n$, from a training data distribution

$$\left(\bigcup_{i \leq n} D_i\right) \sim \varphi(\mathcal{X}, \mathcal{Y}). \tag{1}$$

The goal in FL is to train a single model f on the combined private data of all local clients. This is generally achieved by performing multiple communication rounds, where each round consists of the following steps.

1) A subset $\mathcal{S}_t \subseteq \{1, \ldots, n\}$ of the client population is selected for training and downloads a model initialization from the server.
2) Starting from this model initialization, each client then proceeds to train a model f_i on its local private data D_i by taking multiple steps of stochastic gradient descent over the model parameters θ_i.
3) Finally, the updated models f_i, $i \in \mathcal{S}_t$, are sent back to the server, where they are aggregated to form a new server model f, which is used as the initialization point for the next round of FL.

The goal of FL is to obtain a server model f, which optimally generalizes to new samples from the training data distribution φ, within a minimum number of communication rounds $t \leq T$.

FD offers a new way of performing the last step of the FL protocol, namely the aggregation of the contributions of FL clients into a single-server model [13], [15]–[17]. Instead of aggregating the client model parameters θ_i directly (for instance, via an averaging operation), the server leverages distillation [18] to train a model on the combined predictions of the client models f_i on some public auxiliary set of unlabeled data

$$D_{\text{aux}} \sim \psi(\mathcal{X}). \tag{2}$$

The distribution of the unlabeled auxiliary data $\psi(\mathcal{X})$ hereby is generally assumed to deviate from the unknown private data distribution $\varphi(\mathcal{X})$.

Let $x \in D_{\text{aux}}$ be a batch of data from the auxiliary distillation dataset. Then one iteration of distillation over the parameters of the server model θ^t in communication round t

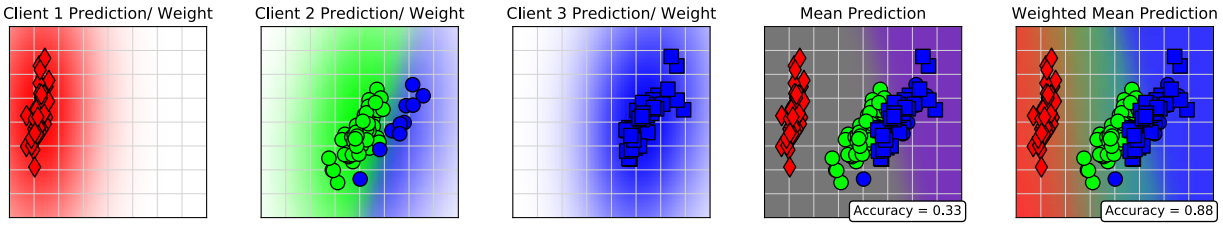


Fig. 2. Weighted ensemble distillation illustrated in a toy example on the Iris dataset (data points are projected to their two principal components). Three FL clients hold disjoint non-i.i.d. subsets of the training data. Panels 1–3: Predictions made by linear classifiers trained on the data of each client. Labels and predictions are color-coded, client certainty (measured via Gaussian KDE) is visualized via the alpha-channel. The mean of client predictions (panel 4) only poorly captures the distribution of training data. In contrast, the certainty-weighted mean of client predictions (panel 5) achieves much higher accuracy.

is performed as

$$\theta^t \leftarrow \theta^t - \eta \frac{\partial D_{\mathrm{KL}}\left(\mathcal{A}(\{f_i(x)|i \in \mathcal{S}_t\}), \sigma\left(f\left(x, \theta^t\right)\right)\right)}{\partial \theta^t}. \quad (3)$$

Hereby, D_{KL} denotes the Kullback–Leibler divergence, $\eta > 0$ is the learning rate, σ is the softmax function, and $\mathcal{A}$ is a mechanism to aggregate the soft labels. Existing work [15] aggregates the client predictions by taking the mean according to

$$\mathcal{A}_{\mathrm{mean}}(\{f_i(x)|i \in \mathcal{S}_t\}) = \sigma\left(\frac{\sum_{i \in \mathcal{S}_t} f_i(x)}{|\mathcal{S}_t|}\right). \quad (4)$$

FD is shown to yield better model fusion than parameter averaging-based techniques, like FEDAVG, resulting in better generalization performance within fewer communication rounds [15]. However, like for all other FL methods, performance of models trained via FD still lacks behind centralized training and convergence speed suffers considerably if training data is distributed in a non-i.i.d. way among the clients.

To address these issues, in this work, we will present two improvements to FD-based training, which, as we will demonstrate, drastically improve training performance in FL scenarios with both homogeneous and heterogeneous client data, leading to greater model performance within fewer communication rounds T.

III. Improving FD via the FedAUX Framework

In this section, we describe how FD-based training can be improved by deriving maximum utility from the available unlabeled auxiliary data. An illustration of our proposed FEDAUX training framework is given in Fig. 1. We first describe FEDAUX for the homogeneous setting where all clients locally train the same model architecture. This setting can readily be generalized to heterogeneous client model architectures as we will describe in Section IV, where also the detailed training procedure is given. An exhaustive qualitative comparison between FEDAUX and baseline methods is given in Section VII.

A. Self-Supervised Pre-Training

As the first component of the FEDAUX training procedure, we will exploit the fact that all FD methods require access to unlabeled auxiliary data D_{aux}. Self-supervised representation learning can leverage such large records of unlabeled data to create models which extract meaningful features. For the two types of data considered in this study—image and sequence data—strong self-supervised training algorithms are known in the form of contrastive representation learning [19], [20] and next-token prediction [21], [22].

Let

$$f_i = g_i \circ h_i \quad (5)$$

denote a decomposition of the local client models $f_i, i = 1, \ldots, n$ into a feature extractor h_i and a classification head g_i. Such a decomposition can trivially be given, for instance, for CNNs and transformer models, where the feature extractor g contains all but the final layer of the network, while the classification head is just a single fully connected layer, followed by a sigmoid activation. As part of the FEDAUX preparation phase (cf. Fig. 1, P1) we propose to pre-train the feature extractor models h_i at the server using self-supervised training on the auxiliary data D_{aux}. We emphasize that this step is only performed once at the beginning of training and makes no assumptions on the similarity between the local training data and the auxiliary data. The pre-training operation results in a parameterization for the feature extractor h_0. Since the training is performed at the server, using only publicly available data, this step inflicts neither computational overhead nor privacy loss on the resource-constrained clients.

B. Weighted Ensemble Distillation

Different studies have shown that the training speed, stability, and maximum achievable accuracy in existing FL algorithms deteriorate if the training data is distributed in a heterogeneous "non-i.i.d." way among the clients [12], [23], [24]. Federated Ensemble Distillation (FedDF) makes no exception to this rule [15].

The underlying problem of combining hypotheses derived from different source domains has been explored in multiple-source domain adaptation theory [25], [26], which shows that standard convex combinations of the hypotheses of the clients as done in [15] may perform poorly on the target domain. Instead, a distribution-weighted combination of the local hypotheses f_i, obtained on data distributions D_i, according to

$$\bar{f}(x) = \sum_i \frac{D_i(x)}{\sum_j D_j(x)} f_i(x) \quad (6)$$

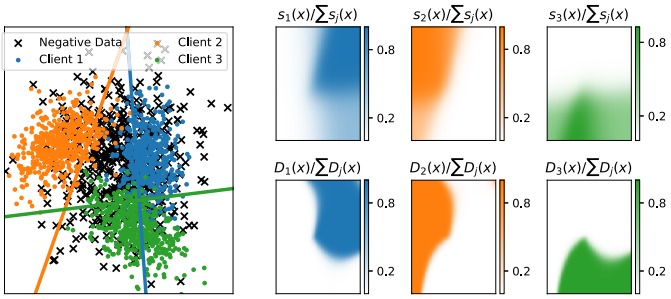


Fig. 3. Left: Toy example with three clients holding data sampled from multivariate Gaussian distributions D_1, D_2, and D_3. All clients solve optimization problem J by contrasting their local data with the public negative data, to obtain scoring models s_1, s_2, s_3, respectively. As can be seen in the plots to the right, our proposed scoring method approximates the robust weights proposed in [25] as it holds $s_i(x)/\sum_j s_j(x) \approx D_i(x)/\sum_j D_j(x)$ on the support of the data distributions.

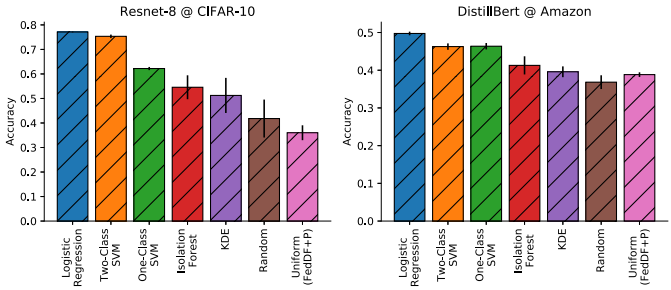


Fig. 4. Comparison of validation performance for FD of ResNet-8 on the CIFAR-10 dataset (left) and DistillBert on the Amazon dataset (right) when different scoring techniques are used to obtain the certainty weights $s_i(x)$ used during ensemble distillation. Certainty scores obtained via two-class logistic regression achieve the best performance and can readily be augmented with a differentially private mechanism.

is shown to be robust [25], [26] (in slight abuse of notation $D_i(x)$ hereby refers to the probability density of the local data D_i). A simple toy example, displayed in Fig. 2, further illustrates this point: Displayed as scatter points are elements of the Iris dataset, projected to their two main PCA components. The training data is distributed among three clients in a non-i.i.d. fashion, with the label of each data point being indicated by the marker color in the plot. Overlayed in the background are the predictions of linear classifier models that were trained on the local data of each client. As we can see, the models which were trained on the data of clients 1 and 3, uniformly predict that all inputs belong to the "red" and "blue" class, respectively. The predictive power of these models and consequently their value as teachers for model distillation is thus very limited. This is also visualized in panel 4, where the mean prediction of the teacher models is displayed. We can, however, improve the teacher ensemble quite significantly, if we weight each teacher's predictions at every location x by its certainty $s(x)$ (approximated via Gaussian KDE), illustrated via the alpha channel in panels 1–3. As we can see in panel 5, weighing the ensemble predictions raises the accuracy from 33% to 88% in this particular toy example.

Based on these insights, we propose to modify the aggregation rule of FD (4) to a certainty-weighted average

$$\mathcal{A}_s(\{(f_i(x), s_i(x)) | i \in \mathcal{S}_t\}) = \sigma\left(\frac{\sum_{i\in\mathcal{S}_t} s_i(x) f_i(x)}{\sum_{j\in\mathcal{S}_t} s_j(x)}\right). \quad (7)$$

The question remains, how to calculate the certainty scores $s_i(x)$ in a privacy preserving way and for arbitrary high-dimensional data, where simple methods, such as Gaussian KDE used in our toy example, fall victim to the curse of dimensionality. To this end, we propose the following methodology.

We split the available auxiliary data randomly into two disjoint subsets

$$D^- \cup D_{\text{distill}} = D_{\text{aux}} \quad (8)$$

the "negative" data and the "distillation" data. Using the pre-trained model h_0 ($\rightarrow$ Section III-A) as a feature extractor, on each client, we then train a logistic regression classifier to separate the local data D_i from the negatives D^-, by optimizing the following regularized empirical risk minimization (ERM) problem:

$$w_i^* = \arg\min_w J\left(w, h_0, D_i, D^-\right) \quad (9)$$

with

$$J\left(w, h_0, D_i, D^-\right) = a \sum_{x\in D_i\cup D^-} l\left(t_x \langle w, \tilde{h}_0(x)\rangle\right) + \lambda R(w). \quad (10)$$

Hereby, $t_x = 2(\mathbb{1}_{x\in D_i}) - 1 \in [-1, 1]$ defines the binary labels of the separation task, $a = (|D_i| + |D^-|)^{-1}$ is a normalizing factor and $\tilde{h}_0(x) = h_0(x)(\max_{x\in D_i\cup D^-} \|h_0(x)\|)^{-1}$ are the normalized features. We choose $l(z) = \log(1+\exp(z))$ to be the logistic loss and $R(w) = (1/2)\|w\|_2^2$ to be the ℓ_2-regularizer. Since J is λ-strongly convex in w, problem (9) is uniquely solvable. This step is performed only once on every client, during the preparation phase (cf. Fig. 1, P2) and the computational overhead for the clients of solving (9) is negligible in comparison to the cost of multiple rounds of training the (deep) model f_i.

Given the solution of the regularized ERM problem w_i^*, the certainty scores on the distillation data D_{distill} can be obtained via the logistic scoring head

$$s_i(x) = \left(1 + \exp\left(-\langle w_i^*, \tilde{h}_0(x)\rangle\right)\right)^{-1} + \xi. \quad (11)$$

A small additive $\xi > 0$ ensures numerical stability when taking the weighted mean in (7). We always set $\xi = 1e-8$.

While the scores $s_i(x)$ can be estimated using a number of different techniques like density estimation, uncertainty quantification [27], or outlier detection [28], [29], we will now present three distinct motivations for using the logistic regression-based approach described above.

First of all, as illustrated using the toy example given in Fig. 3, the scores obtained via our proposed logistic regression-based approach (11) give a good approximation to the distribution weights suggested by domain adaptation theory [25]. As we can see in the panels to the right, it approximately holds

$$\frac{s_i(x)}{\sum_j s_j(x)} \approx \frac{D_i(x)}{\sum_j D_j(x)} \quad \forall\, x \in \mathcal{X},\ i = 1, \ldots, n \quad (12)$$

on the support of the data distributions $p_i \sim D_i$.

Second, scores obtained via logistic regression yield strong empirical performance on highly complex image data. Fig. 4 shows the maximum accuracy achieved after ten communication rounds, by different weighted FedDF methods in an FL scenario with ten clients and highly heterogeneous data ($\alpha = 0.01$, further details on the data splitting strategy are given in Section VI). As we can see, the contrastive logistic scoring approach described above distinctively outperforms the uniform scoring approach used in [15] and also yields better results than other generative and discriminative scoring methods, like Gaussian KDE, Isolation Forests, or One- and Two-Class SVMs. Details on the implementation of these scoring methods are given in Supplementary Materials C.

Finally, as we will see in Section III-C, the logistic scoring mechanism can readily be augmented with differential privacy (DP) and provides high utility even under strong formal privacy constraints.

C. Differentially Private Weighted Ensemble Distillation

Sharing the certainty scores $\{s_i(x)|x \in D_{\text{distill}}\}$ with the central server intuitively causes privacy loss for the clients. After all, a high score $s_i(x)$ indicates that the public data point $x \in D_{\text{distill}}$ is similar to the private data D_i of client i (in the sense of (9)). To protect the privacy of the clients as well as quantify and limit the privacy loss, we propose to use data-level DP (cf. Fig. 1, P3). Following the classic definition of [30], a randomized mechanism is called differentially private, if its output on any input database d is indistinguishable from output on any neighboring database d' which differs from d in one element.

Definition 1: A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ satisfies (ε, δ)-DP if for any two adjacent inputs d and d' that differ in only one element and for any subset of outputs $S \subseteq \mathcal{R}$, it holds that

$$P[\mathcal{M}(d) \in S] \leq \exp(\varepsilon) P\left[\mathcal{M}\left(d'\right) \in S\right] + \delta. \tag{13}$$

DP of a mechanism $\mathcal{M}$ can be achieved, by limiting its sensitivity

$$\Delta(\mathcal{M}) = \max_{d_1, d_2 \in \mathcal{D}} \|\mathcal{M}(d_1) - \mathcal{M}(d_2)\| \tag{14}$$

and then applying a randomized noise mechanism. We adapt a theorem from [31] to establish the sensitivity of (9).

Theorem 1: If $R(\cdot)$ is differentiable and one-strongly convex and l is differentiable with $|l'(z)| \leq 1 \ \forall z$, then the ℓ^2-sensitivity $\Delta_2(\mathcal{M})$ of the mechanism

$$\mathcal{M} : D_i \mapsto \arg\min_w J\left(f, h_0, D_i, D^-\right) \tag{15}$$

is at most $2(\lambda(|D_i| + |D^-|))^{-1}$.

The proof can be found in Supplementary Materials G. As we can see the sensitivity scales inversely with the size of the total data $|D_i| + |D^-|$. From Theorem 1 and application of the Gaussian mechanism [30], it follows that the randomized mechanism:

$$\mathcal{M}_{\text{san}} : D_i \mapsto \arg\min_f J\left(f, h_0, D_i, D^-\right) + N \tag{16}$$

with $N \sim \mathcal{N}(\mathbf{0}, I\sigma^2)$, and $\sigma^2 = (8\ln(1.25\delta^{-1}))/(\varepsilon^2\lambda^2(|D_i| + |D^-|)^2)$ is (ε, δ)-differentially private.

The post-processing property of DP ensures that the release of any number of scores computed using the output of mechanism $\mathcal{M}_{\text{san}}$ is still (ε, δ)-private. Note that in this work we restrict ourselves to the privacy analysis of the scoring mechanism. The differentially private training of deep classifiers f_i is a challenge in its own right and has been addressed, for example, in [32]. Following the basic composition theorem [30], the total privacy cost of running FEDAUX is the sum of the privacy loss of the scoring mechanism $\mathcal{M}_{\text{san}}$ and the privacy loss of communicating the updated models f_i (the latter is the same for all FL algorithms).

IV. DETAILED ALGORITHM FOR THE GENERAL MODEL HETEROGENEOUS SETTING

Like many other FD methods, FEDAUX can natively be applied to FL scenarios where the clients locally train different model architectures. To perform model fusion in such heterogeneous scenarios, FEDAUX constructs several prototypical models on the server, where each prototype represents all clients with identical architecture.

Let us denote by $\mathcal{P}$ the set of all such model prototypes. Then we can define a HashMap $\mathcal{R}$ that maps each client i to its corresponding model prototype P as well as the inverse HashMap $\tilde{\mathcal{R}}$ that maps each model prototype P to the set of corresponding clients (s.t. $i \in \tilde{\mathcal{R}}[\mathcal{R}[i]] \ \forall i$).

The training procedure of FEDAUX can be divided into a preparation phase, which is given in Algorithm 1 and a training phase, which is given in Algorithm 2.

A. Preparation Phase

In the preparation phase, the server uses the unlabeled auxiliary data D_{aux}, to pre-train the feature extractor h^P for each model prototype P using self-supervised training. Suitable methods for self-supervised pre-training are contrastive representation learning [19], or self-supervised language modeling/next-token prediction [21]. The pre-trained feature extractors h_0^P are then communicated to the clients and used to initialize part of the local classifier $f = g \circ h$. The server also communicates the negative data D^- to the clients (in practice, we can instead communicate the extracted features $\{|h_0^P(x)|x \in D^-\}$ of the raw data D^- to save communication). Each client then optimizes the logistic similarity objective J (9) and sanitizes the output by adding properly scaled Gaussian noise. Finally, the sanitized scoring model w_i^* is communicated to the server, where it is used to compute certainty scores s_i on the distillation data (the certainty scores can also be computed on the clients, however this results in additional communication of distillation data and scores).

B. Training Phase

The training phase is carried out in T communication rounds. In every round $t \leq T$, the server randomly selects a subset $\mathcal{S}_t$ of the overall client population and transmits to them the latest server models $\theta^{\mathcal{R}[i]}$, which match their model

Algorithm 1 FEDAUX Preparation Phase (With Different Model Prototypes $\mathcal{P}$)

init: Split $D^- \cup D_{distill} \leftarrow D_{aux}$
init: HashMap $\mathcal{R}$ that maps client i to model prototype P
Server does:
for each model prototype $P \in \mathcal{P}$ **do**
  $h_0^P \leftarrow$ train_self_supervised(h^P, D_{aux})
end for
for each client $i \in \{1, \ldots, n\}$ **in parallel do**
  Client i does:
  $P \leftarrow \mathcal{R}[i]$
  $\sigma^2 \leftarrow \frac{8 \ln(1.25\delta^{-1})}{\varepsilon^2 \lambda^2 (|D_i|+|D^-|)^2}$
  $w_i^* \leftarrow \arg\min_w J(w, h_0^P, D_i, D^-) + \mathcal{N}(\mathbf{0}, I\sigma^2)$
  $\gamma_i \leftarrow \max_{x \in D_i \cup D^-} \|h_0^P(x)\|$
end for
Server does:
for $i = 1, \ldots, n$ **do**
  create HashMap
  $s_i \leftarrow \{x \mapsto (1 + \exp(-\langle w_i^*, \gamma_i^{-1} h_0^P(x) \rangle))^{-1} + \xi$ for $x \in D_{distill}\}$
end for

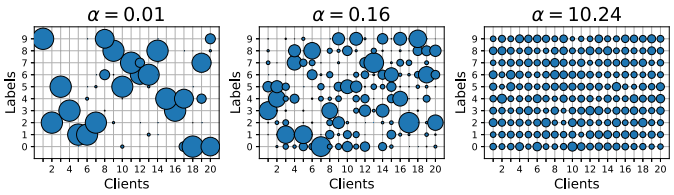


Fig. 5. Illustration of the Dirichlet data splitting strategy we use throughout the article, exemplary for an FL setting with 20 clients and ten different classes. Marker size indicates the number of samples held by one client for each particular class. Lower values of α lead to more heterogeneous distributions of client data. Figure adapted from [15].

prototype P (in round $t = 1$ only the pre-trained feature extractor h_0^P is transmitted). Each selected client updates its local model by performing multiple steps of stochastic gradient descent (or its variants) on its local training data. This results in an updated parameterization θ_i on every client, which is communicated to the server. After all clients have finished their local training, the server gathers the updated parameters θ_i.

Following the recommendations from [15], each prototypical student model is initialized with the average of the parameters from all client models which share the same architecture, according to

$$\theta^P \leftarrow \sum_{i \in \mathcal{S}_t \cap \tilde{\mathcal{R}}[P]} \frac{|D_i|}{\sum_{l \in \mathcal{S}_t \cap \tilde{\mathcal{R}}[P]} |D_l|} \theta_i. \tag{17}$$

Using these model averages as a starting point, for each prototype, the server then distills a new model, based on the client's certainty-weighted predictions.

Algorithm 2 FEDAUX Training Phase (With Different Model Prototypes $\mathcal{P}$). Training Requires Feature Extractors h_0^P and Scores s_i From Alg. 1. The Same $D^- \cup D_{\text{distill}} \leftarrow D_{\text{aux}}$ as in Alg. 1 Is Used. Choose Learning Rate η and Set $\xi = 10^{-8}$

init: HashMap $\mathcal{R}$ that maps client i to model prototype P
init: Inverse HashMap $\tilde{\mathcal{R}}$ that maps model prototype P to set of clients (s.t. $i \in \tilde{\mathcal{R}}[\mathcal{R}[i]]\ \forall i$)
init: Initialize model prototype weights θ^P with feature extractor weights h^P from Alg. 1
for communication round $t = 1, \ldots, T$ **do**
  select subset of clients $\mathcal{S}_t \subseteq \{1, \ldots, n\}$
  for selected clients $i \in \mathcal{S}_t$ **in parallel do**
    Client i does:
    $\theta_i \leftarrow \text{train}(\theta_0 \leftarrow \theta^{\mathcal{R}[i]}, D_i)$ # Local Training
  end for
  Server does:
  for each model prototype $P \in \mathcal{P}$ **do**
    $\theta^P \leftarrow \sum_{i \in \mathcal{S}_t \cap \tilde{\mathcal{R}}[P]} \frac{|D_i|}{\sum_{l \in \mathcal{S}_t \cap \tilde{\mathcal{R}}[P]} |D_l|} \theta_i$ # Parameter Averaging
    for mini-batch $x \in D_{distill}$ **do**
      $\tilde{y} \leftarrow \sigma\left(\frac{\sum_{i \in \mathcal{S}_t} s_i[x] f_i(x, \theta_i)}{\sum_{i \in \mathcal{S}_t} s_i[x]}\right)$ # Can be arbitrary
      $\theta^P \leftarrow \theta^P - \eta \frac{\partial D_{KL}(\tilde{y}, \sigma(f(x, \theta^P)))}{\partial \theta^P}$ # Optimizer
    end for
  end for
end for

V. Related Work

A. Ensemble Distillation in FL

FD is a new area of research, which has attracted tremendous attention in the past couple of years. FD techniques have at least three distinct advantages over prior, parameter averaging-based methods and related work can be organized according to which of these aspects it primarily focuses on.

First, FD enables aggregation of client knowledge independent of the model architecture and thus allows clients to train models of different architecture, which gives additional flexibility, especially in hardware-constrained settings. FEDMD [33], Cronus [34], and FEDH2L [35] are methods which focus on this aspect. While the main focus of FEDAUX is to improve performance, our proposed approach is still flexible enough to handle heterogeneous client models as shown in Section IV.

A second line of FD research explores the advantageous communication properties of the framework. As models are aggregated by means of distillation instead of parameter averaging, it is no longer necessary to communicate the raw parameters. Instead, it is sufficient for the clients to only send their soft-label predictions on the distillation data. Consequently, the communication in FD scales with the size of the distillation dataset and not with the size of the jointly trained model as in the classical parameter averaging-based FL. This leads to communication savings, especially if the local models are large and the distillation dataset is small. Jeong *et al.* and subsequent work [13], [14], [16], [36] focus on this aspect. These methods, however, are computationally more expensive for the resource constrained clients, as distillation needs to be performed locally and perform worse than parameter averaging-based training after the same number of communication rounds. We want to highlight that improving communication efficiency is not a goal of our proposed

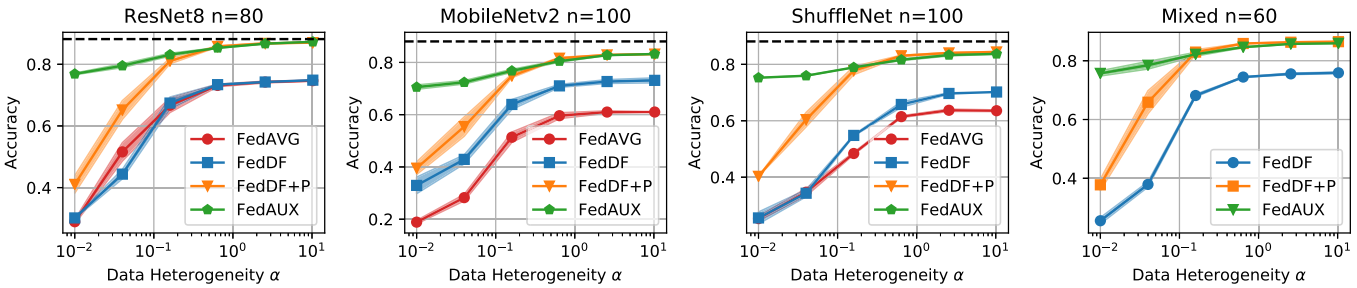


Fig. 6. Evaluation on different neural networks and client population sizes n. Accuracy achieved after $T = 100$ communication rounds by different FD methods at different levels of data heterogeneity α. STL-10 is used as auxiliary dataset. In the "Mixed" setting, one-third of the client population each trains on ResNet8, MobileNetv2, and Shufflenet, respectively. Black dashed line indicates centralized training performance.

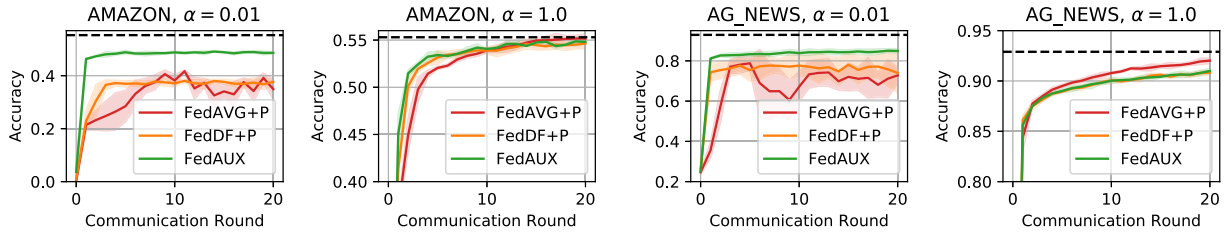


Fig. 7. Evaluating FEDAUX on NLP benchmarks. Performance of FEDAUX for different combinations of local datasets and heterogeneity levels α. Ten clients training TinyBERT at $\alpha = 0.01$ and $C = 100\%$. Bookcorpus is used as auxiliary dataset. Black dashed line indicates centralized training performance.

method, which relies on communication of full models and thus requires communication at the order of conventional parameter averaging-based methods.

Third, when combined with parameter averaging, it has been observed that FD methods achieve better performance than purely parameter averaging-based techniques. Lin *et al.* [15] and Chen and Chao [17] propose FL protocols, which are based on classical FEDAVG and perform ensemble distillation after averaging the received client updates at the server to improve performance. FEDBE, proposed by [17], additionally combines client predictions by means of a Bayesian model ensemble to further improve robustness of the aggregation. Our work primarily focuses on this latter aspect. Building upon the work of [15], we additionally leverage the auxiliary distillation data for unsupervised pre-training and weigh the client predictions in the distillation step according to their certainty scores to better cope with settings where the client's data generating distributions are statistically heterogeneous.

We also mention the related work by Guha *et al.* [37], which proposes a one-shot distillation method for convex models, where the server distills the locally optimized client models in a single round as well as the work of [38] which addresses privacy issues in FD. Federated one-shot distillation is also addressed in [39]. FD for edge-learning was proposed in [40].

B. Weighted Ensembles

FEDAUX leverages a weighted ensemble of client models to distill the locally acquired knowledge into a central server model. The ensemble weights are determined at an instance level, based on the certainty of each local model's prediction. The study of weighted ensembles started around the 1990s with the work by Hashem and Schmeiser [41], Perrone and Cooper [42], and Sollich and Krogh [43]. A weighted ensemble of models combines the output of the individual models by means of a weighted average in order to improve the overall generalization performance. The weights allow us to indicate the percentage of trust or expected performance for each individual model. See [44], [45] for an overview of ensemble methods. Instead of giving each client a static weight in the aggregation step of distillation, we weight the clients on an instance base as in [46], that is, each client's prediction is weighted using a data-dependent certainty score. We note that weighted combinations of weak classifiers are also commonly leveraged in centralized settings in the context of mixture of experts and boosting methods [47]–[49].

C. Data Heterogeneity in FL

As we will demonstrate, FEDAUX excels, in particular, in situations where data is distributed heterogeneously among the clients. As the training data is generated independently on the participation devices, this type of statistical heterogeneity in the client data is very typical for FL problems [1]. It is well known that conventional FL algorithms like FEDAVG [1] perform best on statistically homogeneous data and suffer severely in this ("non-i.i.d.") setting [23], [24]. A number of different studies [11], [12], [17], [23] have tried to address this issue, but relevant performance improvements so far have only been possible under strong assumptions. For instance, [23] assume that the server has access to labeled public data from the same distribution as the clients. In contrast, we only assume that the server has access to unlabeled public data from a potentially deviating distribution. Other approaches [12] require high-frequent communication, with up to thousands of communication rounds, between the server and clients, which might be prohibitive in a majority of FL applications where communication channels are intermittent

and slow. In contrast, our proposed approach can drastically improve FL performance on non-i.i.d. data even after just one single communication round. For completeness, we note that there also exists a different line of research, which aims to address data heterogeneity in FL via meta- and multi-task learning. Here, separate models are trained for each client [50], [51] or clients are grouped into different clusters with similar distributions [52], [53].

D. Unlabeled Data in FL

FEDAUX, like all FD methods, leverages unlabeled auxiliary data during Federated training. To the best of our knowledge, there do not exist any prior studies on the use of unlabeled auxiliary data in FL outside of FD methods. Federated semi-supervised learning techniques [54], [55] assume that clients hold both labeled and unlabeled private data from the local training distribution. In contrast, we assume that the server has access to public unlabeled data that may differ in distribution from the local client data. Federated self-supervised representation learning [56] aims to train a feature extractor on private unlabeled client data. In contrast, we leverage self-supervised representation learning at the server to find a suitable model initialization.

VI. EXPERIMENTS

A. Setup

1) Datasets and Models: We evaluate FEDAUX and SOTA FL methods on both Federated image and text classification problems with large-scale convolutional and transformer models, respectively. For our image classification problems, we train ResNet- [57], MobileNet- [58], and ShuffleNet- [59]-type models on CIFAR-10 and CIFAR-100 and use STL-10, CIFAR-100, and SVHN as well as different subsets of ImageNet (Mammals, Birds, Dogs, Devices, Invertebrates, Structures)[1] as auxiliary data. In our experiments, we always use 80% of the auxiliary data as distillation data D_{distill} and 20% as negative data D^-. For our text classification problems, we train Tiny-Bert [60] on the AG-NEWS [61] and Multilingual Amazon Reviews Corpus [62] and use BookCorpus [63] as auxiliary data.

2) FL Environment and Data Partitioning: We consider FL problems with up to $n = 100$ participating clients. In all experiments, we split the training data evenly among the clients according to a Dirichlet distribution following the procedure outlined in [64] and illustrated in Fig. 5. This allows us to smoothly adapt the level of non-i.i.d.-ness in the client data using the Dirichlet parameter α. We experiment with values for α varying between 100.0 and 0.01. A value of $\alpha = 100.0$ results in an almost identical label distribution, while setting $\alpha = 0.01$ results in a split, where the vast majority of data on every client stems from one single class. See Supplementary Material A for a more detailed description of our data splitting procedure. We vary the client participation rate C in every round between 20% and 100%.

[1]The methodology for generating these subsets is described in Supplementary Materials D.

3) Pre-Training Strategy: For our image classification problems, we use contrastive representation learning as described in [19] for pre-training. We use the default set of data augmentations proposed in this article and train with the Adam optimizer, learning rate set to 10^{-3}, and a batch size of 512. For our text classification problems, we pre-train using self-supervised next-word prediction.

4) Training the Scoring Model and Privacy Setting: We set the default privacy parameters to $\lambda = 0.1$, $\varepsilon = 0.1$, and $\delta = 1e-5$, and solve (9) by running L-BFGS [65] until convergence ($\leq$1000 steps).

5) Baselines: We compare the performance of FEDAUX to state-of-the-art FL methods: FEDAVG [1], FEDPROX [11], FedDF [15], and FEDBE [17]. To clearly discern the performance benefits of the two components of FEDAUX (unsupervised pre-training and weighted ensemble distillation), we also report performance metrics on versions of these methods where the auxiliary data was used to pre-train the feature extractor h ("FEDAVG + P," "FEDPROX + P," "FEDDF + P," respectively, "FEDBE + P"). For FEDBE, we set the sample size to 10 as suggested in this article. For FEDPROX, we always tune the proximal parameter μ.

6) Optimization: On all image classification task, we use the very popular Adam optimizer [66], with a fixed learning rate of $\eta = 10^{-3}$ and a batch size of 32 for local training. Distillation is performed for one epoch for all methods using Adam at a batch size of 128 and fixed learning rate of $5e-5$. More detailed hyperparameter analysis in Supplementary Material F shows that this choice of optimization parameters is approximately optimal for all of the methods. If not stated otherwise, the number of local epochs E is set to 1.

B. Evaluating FEDAUX on Common FL Benchmarks

We start out by evaluating the performance of FEDAUX on classic benchmarks for Federated image classification. Fig. 6 shows the maximum accuracy achieved by different FD methods after $T = 100$ communication rounds at different levels of data heterogeneity. As we can see, FEDAUX distinctively outperforms FEDDF on the entire range of data heterogeneity levels α on all benchmarks. For instance, when training ResNet8 with $n = 80$ clients at $\alpha = 0.01$, FEDAUX raises the maximum achieved accuracy from 30.4% to 78.1% (under the same set of assumptions). The two components of FEDAUX, unsupervised pre-training and weighted ensemble distillation, both contribute independently to the performance improvement, as can be seen when comparing with FEDDF + P, which only uses unsupervised pre-training. Weighted ensemble distillation as done in FEDAUX leads to greater or equal performance than equally weighted distillation (FEDDF + P) across all levels of data heterogeneity. The same overall picture can be observed in the "Mixed" setting where one-third of the client population each trains on ResNet8, MobileNetv2, and Shufflenet, respectively. (In this setting, parameter averaging is not possible and thus FEDAVG cannot be applied.) Detailed training curves are given in the Supplementary Material B.

Table I compares the performance of FEDAUX and baseline methods at different client participation rates C. We can see

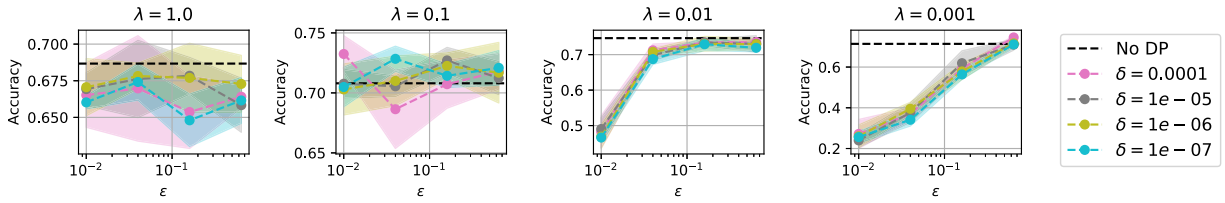


Fig. 8. Privacy analysis. Performance of FEDAUX for different combinations of the privacy parameters ε, δ, and λ. Forty clients training Resnet-8 for $T = 10$ rounds on CIFAR-10 at $\alpha = 0.01$ and $C = 40\%$. STL-10 is used as auxiliary dataset.

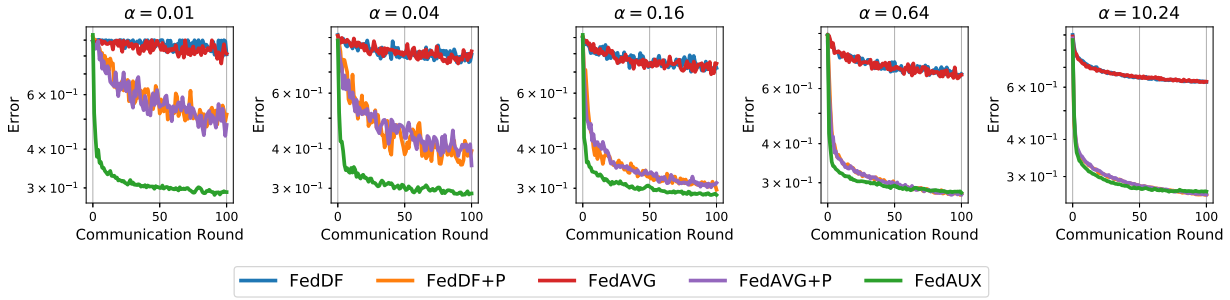


Fig. 9. Linear evaluation. Training curves for different FL methods at different levels of data heterogeneity α when only the classification head g is updated in the training phase. A total of $n = 80$ clients training ResNet8 on CIFAR-10 at $C = 40\%$, using STL-10 as auxiliary dataset.

that FEDAUX benefits from higher participation rates. In all scenarios, methods which are initialized using the pre-trained feature-extractor h_0 distinctively outperform their randomly initialized counterparts. In the i.i.d. setting at $\alpha = 100.0$, FEDAUX is mostly on par with the (improved) parameter averaging-based methods FEDAVG + P and FEDPROX + P, with a maximum performance gap of 0.8%. At $\alpha = 0.01$, on the other hand, FEDAUX outperforms all other methods with a margin of up to 29%.

C. Evaluating FEDAUX on NLP Benchmarks

Fig. 7 shows learning curves for Federated training of TinyBERT on the Amazon and AG-News datasets at two different levels of data heterogeneity α. We observe that FEDAUX significantly outperforms FEDDF + P as well as FEDAVG + P in the heterogeneous setting ($\alpha = 0.01$) and reaches 95% of its final accuracy after one communication round on both datasets, indicating suitability for one-shot learning. On more homogeneous data ($\alpha = 1.0$), FEDAUX performs mostly on par with pre-trained versions of FEDAVG and FEDDF, with a maximal performance gap of 1.1 % accuracy on the test set. We note that effects of data heterogeneity are less severe as in this setting as both the AG News and the Amazon dataset only have four and five labels, respectively, and an α of 1.0 already leads to a distribution where each client owns a subset of the private dataset containing all possible labels. Further details on our implementation can be found the Supplementary Material E.

D. Privacy Analysis of FEDAUX

Fig. 8 examines the dependence of FEDAUX' training performance of the privacy parameters ε, δ, and the regularization parameter λ. As we can see, performance comparable to non-private scoring is achievable at conservative privacy parameters ε and δ. For instance, at $\lambda = 0.01$ setting $\varepsilon = 0.04$ and $\delta = 10^{-6}$ reduces the accuracy from 74.6% to 70.8%. At higher values of λ, better privacy guarantees have an even less harmful effect, at the cost however of an overall degradation in performance. Throughout this empirical study, we have set the default privacy parameters to $\lambda = 0.1$, $\varepsilon = 0.1$, and $\delta = 1e - 5$. We also perform an empirical privacy analysis in the Supplementary Material H, which provides additional intuitive understanding and confidence in the privacy properties of our method.

E. Evaluating the Dependence on Auxiliary Data

Next, we investigate the influence of the auxiliary dataset D_{aux} on unsupervised pretraining, distillation, and weighted distillation, respectively. We use CIFAR-10 as training dataset and consider 8 different auxiliary datasets, which differ w.r.t. their similarity to this client training data—from more similar (STL-10, CIFAR-100) to less similar (Devices, SVHN).[2] Table II shows the maximum achieved accuracy after $T = 100$ rounds when each of these datasets is used as auxiliary data. As we can see, performance always improves when auxiliary data is used for unsupervised pre-training. Even for the highly dissimilar SVHN dataset (which contains images of house numbers) performance of FEDDF + P improves by 1% over FEDDF in both the i.i.d. and non-i.i.d. regime. For other datasets like Dogs, Birds, or Invertebrates, performance improves by up to 14%, although they overlap with only one single class of the CIFAR-10 dataset. The outperformance of FEDAUX on such a wide variety of highly dissimilar datasets suggest that beneficial auxiliary data should be available in

[2]The CIFAR-10 dataset contains images from the classes airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truc.

TABLE I

MAXIMUM ACCURACY ACHIEVED BY FEDAUX AND OTHER BASELINE FL METHODS AFTER $T = 100$ COMMUNICATION ROUNDS, AT DIFFERENT PARTICIPATION RATES C AND LEVELS OF DATA HETEROGENEITY α. TWENTY CLIENTS TRAINING RESNET-8 ON CIFAR-10. AUXILIARY DATA USED IS STL10. *METHODS ASSUME AVAILABILITY OF AUXILIARY DATA. †IMPROVED BASELINES

	$\alpha = 0.01$			$\alpha = 100.0$		
Method	$C = 0.2$	$C = 0.4$	$C = 0.8$	$C = 0.2$	$C = 0.4$	$C = 0.8$
FEDAVG [1]	19.9±0.7	23.6±2.0	28.9±2.0	81.3±0.1	82.2±0.0	82.3±0.1
FEDPROX [11]	28.4±2.5	34.0±1.9	42.0±1.0	81.4±0.1	82.3±0.2	82.0±0.3
FEDDF* [15]	25.0±0.8	27.8±0.8	30.6±0.3	80.8±0.1	81.4±0.3	81.5±0.3
FEDBE* [17]	20.9±0.6	25.7±1.4	29.1±0.1	81.4±0.7	82.0±0.1	82.2±0.2
FEDAVG+P*†	30.4±7.9	32.1±2.0	38.4±0.5	89.0±0.1	**89.5±0.1**	**89.6±0.1**
FEDPROX+P*†	42.8±2.7	43.1±0.2	49.0±0.7	88.9±0.0	89.1±0.1	89.4±0.0
FEDDF+P*†	28.8±3.0	39.3±3.6	48.1±1.1	88.8±0.0	88.9±0.1	88.9±0.1
FEDBE+P*†	30.2±2.2	29.8±0.8	37.7±0.0	**89.1±0.1**	89.5±0.2	89.5±0.0
FEDAUX*	**54.2±0.3**	**71.2±2.1**	**78.5±0.0**	88.9±0.0	89.0±0.0	89.0±0.1

TABLE II

MAXIMUM ACCURACY ACHIEVED BY FEDAUX AND OTHER BASELINE FL METHODS AFTER 100 COMMUNICATION ROUNDS, WHEN DIFFERENT SETS OF UNLABELED AUXILIARY DATA ARE USED FOR PRE-TRAINING AND/OR DISTILLATION. FORTY CLIENTS TRAINING RESNET-8 ON CIFAR-10 AT $C = 40\%$

		Auxiliary Data								
α	Method	STL-10	CIFAR-100	SVHN	Invertebr.	Birds	Devices	Dogs	Structures	Mammals
0.01	FEDDF	27.9±3.2	29.5±6.2	28.1±3.9	28.5±3.6	30.1±2.0	26.3±0.2	28.9±5.1	30.2±7.0	28.7±4.3
	FEDDF+P	43.0±5.2	41.6±1.1	29.6±3.4	38.8±6.5	41.4±5.9	35.9±4.9	41.1±7.3	36.7±7.1	39.4±2.3
	FEDAUX	**76.8±0.9**	**71.5±2.5**	**43.7±1.5**	**68.2±0.7**	**65.7±3.1**	**71.5±0.1**	**71.8±3.8**	**64.1±3.3**	**73.2±1.0**
100.00	FEDDF	79.3±0.7	79.9±0.1	80.9±0.1	80.2±0.1	80.2±0.4	79.4±0.3	79.7±0.4	80.1±0.2	80.0±0.1
	FEDDF+P	88.3±0.0	86.7±0.0	**81.7±0.2**	87.4±0.1	87.6±0.0	87.7±0.1	88.4±0.0	**87.4±0.1**	88.6±0.2
	FEDAUX	**88.5±0.0**	**86.7±0.1**	81.6±0.0	**87.8±0.1**	**87.8±0.1**	**87.8±0.0**	**88.6±0.0**	87.3±0.1	**88.8±0.2**

TABLE III

ONE-SHOT PERFORMANCE OF DIFFERENT FL METHODS. MAXIMUM ACCURACY ACHIEVED AFTER $T = 1$ COMMUNICATION ROUNDS AT PARTICIPATION-RATE $C = 100\%$. EACH CLIENT TRAINS FOR $E = 40$ LOCAL EPOCHS

	MobileNetv2, $n = 100$				Shufflenet, $n = 100$			
Method	$\alpha = 0.01$	$\alpha = 0.04$	$\alpha = 0.16$	$\alpha = 10.24$	$\alpha = 0.01$	$\alpha = 0.04$	$\alpha = 0.16$	$\alpha = 10.24$
FEDAVG	10.3±0.0	13.6±2.3	23.6±0.0	30.5±0.9	12.1±0.8	17.4±0.4	28.2±0.8	37.8±0.7
FEDPROX	11.6±0.8	14.3±1.4	23.7±0.3	30.5±0.5	12.9±1.7	18.9±0.2	29.4±0.3	38.9±0.5
FEDDF	16.8±4.2	29.5±3.8	37.7±1.1	40.4±0.5	16.0±5.1	27.3±0.1	38.7±0.2	45.5±0.5
FEDAVG+P	24.3±1.1	44.0±4.4	57.6±3.7	69.9±0.0	25.5±1.4	44.2±0.1	62.9±1.6	71.9±0.1
FEDPROX+P	27.2±2.2	43.4±3.6	56.9±3.9	70.0±0.1	28.4±0.2	47.1±1.5	63.3±1.2	71.9±0.1
FEDDF+P	46.7±5.6	61.1±1.3	67.6±0.5	71.2±0.1	40.4±2.7	59.4±0.8	68.8±0.2	72.7±0.0
FEDAUX	**64.8±0.0**	**65.5±1.0**	**68.2±0.2**	**71.3±0.1**	**66.9±0.6**	**68.6±0.4**	**70.8±0.3**	**72.9±0.1**

the majority of practical FL problems and also has positive implications from the perspective of privacy. Interestingly, the performance of FEDDF seems to only weakly correlate with the performance of FEDDF + P and FEDAUX as a function of the auxiliary dataset. This suggests that the properties, which make a dataset useful for distillation, are not the same ones that make it useful for pre-training and weighted distillation. Investigating this relationship further is an interesting direction of future research.

F. FEDAUX *in Hardware-Constrained Settings*

1) Linear Evaluation: In settings where the FL clients are hardware-constrained mobile or IoT devices, local training of entire deep neural networks like ResNet8 might be infeasible. We therefore also consider the evaluation of different FL methods, when only the linear classification head g is updated during the training phase. Fig. 9 shows the training curves in this setting when clients hold data from the CIFAR-10 dataset. We see that in this setting, performance of FEDAUX is high, independent of the data heterogeneity levels α, suggesting that in the absence of non-convex training dynamics, our proposed scoring method actually yields robust weighted ensembles in the sense of [25]. We note that FEDAUX also trains much more smoothly, than all other baseline methods.

2) One-Shot Evaluation: In many FL applications, the number of times a client can participate in the Federated training

TABLE IV
QUALITATIVE COMPARISON: COMPLEXITY, COMMUNICATION OVERHEAD, AND PRIVACY LOSS AFTER T COMMUNICATION ROUNDS AS WELL AS IMPLICIT ASSUMPTIONS MADE BY DIFFERENT FL METHODS

	FEDAVG	FEDDF, FEDBE	FEDAUX (preparation phase)	FEDAUX (training phase)
Operations (Clients)	Local Training ($\times T$)	Local Training ($\times T$)	Solve λ-strongly convex ERM (9)	Local Training ($\times T$)
Operations (Server)	Model Averaging ($\times T$)	Model Averaging, Distillation ($\times T$)	Self-Supervised Pre-training of h_0, Computation of certainty scores s_i	Model Averaging, Distillation ($\times T$)
Communication Clients $\rightarrow$ Server	Model Parameters $\lvert\theta_i\rvert$ ($\times T$)	Model Parameters $\lvert\theta_i\rvert$ ($\times T$)	Scoring Models w_i^*	Model Parameters $\lvert\theta_i\rvert$ ($\times T$)
Communication Server $\rightarrow$ Clients	Model Parameters $\lvert\theta\rvert$ ($\times T$)	Model Parameters $\lvert\theta^{\mathcal{R}[i]}\rvert$ ($\times T$)	Negative Data D^-, Feature Extractor h_0	Model Parameters $\lvert\theta^{\mathcal{R}[i]}\rvert$ ($\times T$)
Privacy Loss	Privacy loss of communicating θ_i ($\times T$)	Privacy loss of communicating θ_i ($\times T$)	(ε, δ)-DP	Privacy loss of communicating θ_i ($\times T$)
Assumptions	No Assumptions	Auxiliary Data	Auxiliary Data	Auxiliary Data

is restricted by communication, energy, and/or privacy constraints [37], [67]. To study these types of settings, we investigate the performance of FEDAUX and other FL methods in Federated one-shot learning where we set $T = 1$ and $C = 100\%$. Table III compares performance in this setting for $n = 100$ clients training MobileNetv2 (resp. ShuffleNet). FEDAUX outperforms the baseline methods in this setting at all levels of data heterogeneity α.

VII. DISCUSSION AND QUALITATIVE COMPARISON WITH BASELINE METHODS

The experiments performed in the previous section demonstrate that FEDAUX outperforms state-of-the-art FL methods by wide margins, in particular, if the training data is distributed in a heterogeneous way among the clients. In Table IV, we additionally provide a qualitative comparison between FEDAUX and the baseline methods FEDAVG and FEDDF. We can note the following.

A. Client Workload

Compared with FEDAVG and FEDDF, FEDAUX additionally requires the clients to once solve the λ-strongly convex ERM (9). For this problem, linearly convergent algorithms are known [65] and thus the computational overhead (and energy consumption) is negligible compared with the complexity of multiple rounds of locally training deep neural networks.

B. Server Workload

FEDAUX also adds computational load to the server for self-supervised pre-training and computation of the certainty scores s_i. As the server is typically assumed to have massively stronger computational resources than the clients, this can be neglected.

C. Communication Client $\rightarrow$ Server

Once, in the preparation phase of FEDAUX, the scoring models w_i^* need to be communicated from the clients to the server. The overhead of communicating these H-dimensional vectors, where H is the feature dimension, is negligible compared to the communication of the full models f_i.

D. Communication Server $\rightarrow$ Clients

FEDAUX also requires the communication of the negative data D^- and the feature extractor h_0 from the server to the clients. The overhead of sending h_0 is lower than sending the full model f, and thus the total downstream communication is increased by less than a factor of $(T + 1)/T$. The overhead of sending D^- is small (in our experiments $|D^-| = 0.2|D_{\text{aux}}|$) and can be further reduced by sending extracted features $\{|h_0^P(x)|x \in D^-\}$ instead of the full data. For instance, in our experiments with ResNet-8 and CIFAR-100, we have $|D^-| = 12\,000$ and $h_0^P(x) \in \mathbb{R}^{512}$, resulting in a total communication overhead of $12\,000 \times 512 \times 4B = 24.58$ MB for D^-. For comparison, the total communication overhead of once sending the parameters of ResNet-8 (needs to be done T times) is 19.79 MB.

E. Privacy Loss

Communicating the scoring models w_i^* incurs additional privacy loss for the clients. Using our proposed sanitation mechanism, this process is made (ε, δ)-differentially private. Our experiments in Section VI-D demonstrate that FEDAUX can achieve drastic performance improvements, even under conservative privacy constraints. All empirical results reported are obtained with (ε, δ) DP at $\varepsilon = 0.1$ and $\delta = 10^{-5}$.

F. Assumptions

Finally, FEDAUX makes the additional assumption that unlabeled auxiliary data is available to the server. This assumption is made by all FD methods including FEDDF.

In conclusion, FEDAUX requires comparable resources as state-of-the-art FD methods and has similar privacy properties, while at the same time achieving significantly better performance.

VIII. CONCLUSION AND FUTURE WORK

In this work, we have explored FL in the presence of unlabeled auxiliary data, an assumption made in the quickly

growing area of FD. By leveraging auxiliary data for unsupervised pre-training and certainty weighted ensemble distillation, we were able to demonstrate that this assumption is rather strong and can lead to drastically improved performance of FL algorithms. As we have seen, these performance improvements can be obtained even if the distribution of the auxiliary data is highly divergent from the client data distribution and are maintained when the certainty scores are obfuscated using a strong DP mechanism. Additionally, our detailed qualitative comparison with baseline methods revealed that FEDAUX incurs only marginal excess computation and communication overhead.

On a more fundamental note, the dramatic performance improvements observed in FEDAUX call into question the common practice of comparing FD-based methods (which assume auxiliary data) with parameter averaging-based methods (which do not make this assumption) [15], [17] and thus have implications for the future evaluation of FD methods in general.

An interesting direction of future research would be to explore how well FD methods and FEDAUX, in particular, fare if only synthetically generated auxiliary data is available for distillation and/or pre-training. First studies already show promising results in this direction [68]. Another interesting direction to explore would be the extension of our proposed privacy mechanism from Section III-C to the training phase to fully quantify the privacy loss of the FEDAUX method. Furthermore, certainty estimates of client predictions as provided by FEDAUX could also be used to detect anomalous client behavior and thus increase adversarial robustness [69], [70]. Finally, certainty estimates could also be used to group the client population into clusters in the spirit of [53] for improved performance under structured heterogeneity of the client data.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2017, pp. 1273–1282.

[2] Q. Li *et al.*, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," 2019, *arXiv:1907.09693*.

[3] U. Ahmed, G. Srivastava, and J. C.-W. Lin, "A federated learning approach to frequent itemset mining in cyber-physical systems," *J. Netw. Syst. Manage.*, vol. 29, no. 4, pp. 1–17, Oct. 2021.

[4] D. Połap, G. Srivastava, and K. Yu, "Agent architecture of an intelligent medical system based on federated learning and blockchain technology," *J. Inf. Secur. Appl.*, vol. 58, May 2021, Art. no. 102748.

[5] M. J. Sheller *et al.*, "Federated learning in medicine: Facilitating multi-institutional collaborations without sharing patient data," *Sci. Rep.*, vol. 10, no. 1, pp. 1–12, Dec. 2020.

[6] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Gener. Comput. Syst.*, vol. 115, pp. 619–640, Feb. 2021.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 248–255.

[8] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," 2016, *arXiv:1609.07843*.

[9] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 4615–4625.

[10] S. Reddi *et al.*, "Adaptive federated optimization," 2020, *arXiv:2003.00295*.

[11] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst. (MLSys)*, 2020, pp. 1–22.

[12] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.

[13] S. Itahara, T. Nishio, Y. Koda, M. Morikura, and K. Yamamoto, "Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-IID private data," 2020, *arXiv:2008.06180*.

[14] F. Sattler, A. Marban, R. Rischke, and W. Samek, "CFD: Communication-efficient federated distillation via soft-label quantization and delta coding," *IEEE Trans. Netw. Sci. Eng.*, early access, May 19, 2021, doi: 10.1109/TNSE.2021.3081748.

[15] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 33, 2020, pp. 1–26.

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

[17] H.-Y. Chen and W.-L. Chao, "FedBE: Making Bayesian model ensemble applicable to federated learning," 2020, *arXiv:2009.01974*.

[18] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[19] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 1597–1607.

[20] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 9929–9939.

[21] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol. (NAACL-HLT)*, vol. 1, 2019, pp. 4171–4186.

[22] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[23] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," 2018, *arXiv:1806.00582*.

[24] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-iid data," in *Proc. 8th Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–26.

[25] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation with multiple sources," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 21, 2008, pp. 1041–1048.

[26] J. Hoffman, M. Mohri, and N. Zhang, "Algorithms and theory for multiple-source adaptation," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 31, 2018, pp. 8256–8266.

[27] L. Oala, C. Heiß, J. Macdonald, M. März, G. Kutyniok, and W. Samek, "Detecting failure modes in image reconstructions with interval neural network uncertainty," *Int. J. Comput. Assist. Radiol. Surgery*, vol. 4, pp. 1–9, Sep. 2021.

[28] L. Ruff *et al.*, "A unifying review of deep and shallow anomaly detection," *Proc. IEEE*, vol. 109, no. 5, pp. 756–795, May 2021.

[29] L. Ruff *et al.*, "Deep semi-supervised anomaly detection," 2019, *arXiv:1906.02694*.

[30] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.

[31] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *J. Mach. Learn. Res.*, vol. 12, pp. 1069–1109, Mar. 2011.

[32] M. Abadi *et al.*, "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 308–318.

[33] D. Li and J. Wang, "FedMD: Heterogenous federated learning via model distillation," 2019, *arXiv:1910.03581*.

[34] H. Chang, V. Shejwalkar, R. Shokri, and A. Houmansadr, "Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer," 2019, *arXiv:1912.11279*.

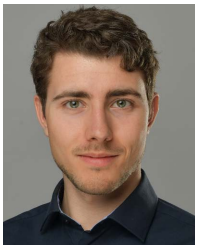
[35] Y. Li, W. Zhou, H. Wang, H. Mi, and T. M. Hospedales, "FedH2L: Federated learning with model and statistical heterogeneity," 2021, *arXiv:2101.11296*.

[36] H. Seo, J. Park, S. Oh, M. Bennis, and S.-L. Kim, "Federated knowledge distillation," 2020, *arXiv:2011.02367*.

[37] N. Guha, A. Talwalkar, and V. Smith, "One-shot federated learning," 2019, *arXiv:1902.11175*.

[38] L. Sun and L. Lyu, "Federated model distillation with noise-free differential privacy," 2020, *arXiv:2009.05537*.

[39] Y. Zhou, G. Pu, X. Ma, X. Li, and D. Wu, "Distilled one-shot federated learning," 2020, *arXiv:2009.07999*.
[40] J.-H. Ahn, O. Simeone, and J. Kang, "Wireless federated distillation for distributed edge learning with heterogeneous data," in *Proc. IEEE 30th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2019, pp. 1–6.
[41] S. Hashem and B. Schmeiser, "Approximating a function and its derivatives using mse-optimal linear combinations of trained feedforward neural networks," in *Proc. World Congr. Neural Netw.*, vol. 1, 1993, pp. 617–620.
[42] M. P. Perrone and L. N. Cooper, "When networks disagree: Ensemble methods for hybrid neural networks," in *Neural Networks for Speech and Image Processing*, R. J. Mammone, Ed. London, U.K.: Chapman and Hall, 1993.
[43] P. Sollich and A. Krogh, "Learning with ensembles: How overfitting can be useful," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 8, 1995, pp. 190–196.
[44] A. J. C. Sharkey, "On combining artificial neural nets," *Connection Sci.*, vol. 8, nos. 3–4, pp. 299–314, Dec. 1996.
[45] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *J. Artif. Intell. Res.*, vol. 11, pp. 169–198, Aug. 1999.
[46] D. Jimenez, "Dynamically weighted ensemble neural networks for classification," in *Proc. IEEE Int. Joint Conf. Neural Netw. World Congr. Comput. Intell.*, vol. 1, May 1998, pp. 753–756.
[47] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty years of mixture of experts," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 8, pp. 1177–1193, Aug. 2012.
[48] S. Masoudnia and R. Ebrahimpour, "Mixture of experts: A literature survey," *Artif. Intell. Rev.*, vol. 42, no. 2, pp. 275–293, 2014.
[49] R. E. Schapire, "A brief introduction to boosting," in *Proc. 16th Int. Joint Conf. Artif. Intell. (IJCAI)*, 1999, pp. 1401–1406.
[50] V. Smith, C. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 30, 2017, pp. 4424–4434.
[51] H. Wu, C. Chen, and L. Wang, "A theoretical perspective on differentially private federated multi-task learning," 2020, *arXiv:2011.07179*.
[52] A. Ghosh, J. Hong, D. Yin, and K. Ramchandran, "Robust federated learning in a heterogeneous environment," 2019, *arXiv:1906.06629*.
[53] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3710–3722, Aug. 2021.
[54] Z. Zhang, Y. Yang, Z. Yao, Y. Yan, J. E. Gonzalez, and M. W. Mahoney, "Improving semi-supervised federated learning by reducing the gradient diversity of models," 2020, *arXiv:2008.11364*.
[55] W. Jeong, J. Yoon, E. Yang, and S. Ju Hwang, "Federated semi-supervised learning with inter-client consistency & disjoint learning," 2020, *arXiv:2006.12097*.
[56] F. Zhang *et al.*, "Federated unsupervised representation learning," 2020, *arXiv:2010.08982*.
[57] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[58] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
[59] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 6848–6856.
[60] X. Jiao *et al.*, "TinyBERT: Distilling BERT for natural language understanding," in *Proc. Conf. Empirical Methods Natural Lang. Process., Findings (EMNLP)*, 2020, pp. 4163–4174.
[61] X. Zhang, J. J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 28, 2015, pp. 649–657.
[62] P. Keung, Y. Lu, G. Szarvas, and N. A. Smith, "The multilingual Amazon reviews corpus," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 4563–4568.
[63] Y. Zhu *et al.*, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 19–27.
[64] T.-M. Harry Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," 2019, *arXiv:1909.06335*.
[65] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Program.*, vol. 45, no. 1, pp. 503–528, 1989.
[66] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[67] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and U. Erlingsson, "Scalable private learning with PATE," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–34.
[68] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," 2021, *arXiv:2105.10056*.
[69] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 634–643.
[70] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated learning-based anomaly detection for IoT security attacks," *IEEE Internet Things J.*, early access, May 5, 2021, doi: 10.1109/JIOT.2021.3077803.



Felix Sattler received the B.Sc. degree in mathematics, the M.Sc. degree in computer science, and the M.Sc. degree in applied mathematics from Technische Universität Berlin, Berlin, Germany, in 2016 and 2018, respectively.

He is currently with the Machine Learning Group, Fraunhofer Heinrich Hertz Institute, Berlin. His research interests include efficient and robust machine learning, federated learning, and multi-task learning.

Tim Korjakow received the B.Sc. degree in computer science from Technische Universität Berlin, Berlin, Germany, in 2019.

He currently works with the Machine Learning Group, Fraunhofer Heinrich Hertz Institute, Berlin. His research interests include distributed machine learning, neural networks, and interpretability methods.

Roman Rischke received the M.Sc. degree in business mathematics from Technische Universität Berlin, Berlin, Germany, in 2012, and the Dr.rer.nat. degree in mathematics from Technische Universität München, Munich, Germany, in 2016.

He currently works as a Post-Doctoral Researcher with the Department of Artificial Intelligence, Fraunhofer Heinrich Hertz Institute, Berlin. His research interests include discrete optimization under data uncertainty, efficient and robust machine learning, and federated learning.

Wojciech Samek (Member, IEEE) studied computer science at the Humboldt University of Berlin, Berlin, Germany, from 2004 to 2010. He received the Ph.D. degree (Hons.) from the Technical University of Berlin, Berlin, in 2014.

He was a Visiting Researcher with the NASA Ames Research Center, Mountain View, CA, USA. In 2014, he founded the Machine Learning Group, Fraunhofer HHI, which he has directed until 2020. He is an Associated Faculty at the Berlin Institute for the Foundation of Learning and Data (BIFOLD), the ELLIS Unit Berlin, and the DFG Graduate School BIOQIC. He is currently the Head of the Department of Artificial Intelligence and the Explainable AI Group, Fraunhofer Heinrich Hertz Institute, Berlin. His research interests include deep learning, explainable AI, neural network compression, and federated learning.

Dr. Samek is an Elected Member of the IEEE MLSP Technical Committee. During his studies, he was awarded scholarships from the German Academic Scholarship Foundation and the DFG Research Training Group GRK 1589/1. He has been serving as an AC for NAACL 2021, was a recipient of multiple best paper awards, including the 2020 Pattern Recognition Best Paper Award, and a part of the MPEG-7 Part 17 standardization. He is an Editorial Board Member of *Pattern Recognition*, *PLoS ONE*, and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS).