2023 IEEE/CVF International Conference on Computer Vision (ICCV) | 979-8-3503-0718-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCV51070.2023.00480

Communication-Efficient Vertical Federated Learning with Limited Overlapping Samples

Jingwei Sun[1], Ziyue Xu[2], Dong Yang[2], Vishwesh Nath[2], Wenqi Li[2], Can Zhao[2], Daguang Xu[2], Yiran Chen[1], Holger R. Roth[2]
[1] Department of Electrical and Computer Engineering, Duke University
[2] NVIDIA
[1]{jingwei.sun, yiran.chen}@duke.edu,
[2]{ziyuex, dongy, vnath, wenqil, canz, daguangx, hroth}@nvidia.com

Abstract

Federated learning is a popular collaborative learning approach that enables clients to train a global model without sharing their local data. Vertical federated learning (VFL) deals with scenarios in which the data on clients have different feature spaces but share some overlapping samples. Existing VFL approaches suffer from high communication costs and cannot deal efficiently with limited overlapping samples commonly seen in the real world. We propose a practical VFL framework called ***one-shot VFL*** *that can solve the communication bottleneck and the problem of limited overlapping samples simultaneously based on semi-supervised learning. We also propose* ***few-shot VFL*** *to improve the accuracy further with just one more communication round between the server and the clients. In our proposed framework, the clients only need to communicate with the server once or only a few times. We evaluate the proposed VFL framework on both image and tabular datasets. Our methods can improve the accuracy by more than 46.5% and reduce the communication cost by more than 330× compared with state-of-the-art VFL methods when evaluated on CIFAR-10. Our code is available at https://nvidia.github.io/NVFlare/research/one-shot-vfl.*

1. Introduction

Federated Learning (FL) is a distributed learning method that enables multiple parties to collaboratively train a model without centralizing their raw data. Therefore, the clients can retain control over their own data assets. FL has received significant attention and has become a major research topic due to its capability to build real-world applications where datasets are isolated across different organizations/devices while preserving data governance and privacy [18, 14].

Existing approaches primarily focus on horizontal federated learning (HFL), where the data from different clients share the same feature space but have different samples [32]. One application of HFL is that smartphone users collaboratively train a next-word prediction model for the smart keyboard [10]. In HFL, the clients are expected to learn common knowledge from heterogeneous data distributions and produce a global model by aggregating the updates of local models. Hence, the main challenge of HFL is data distribution heterogeneity and under cross-device scenarios, limited computation resources.

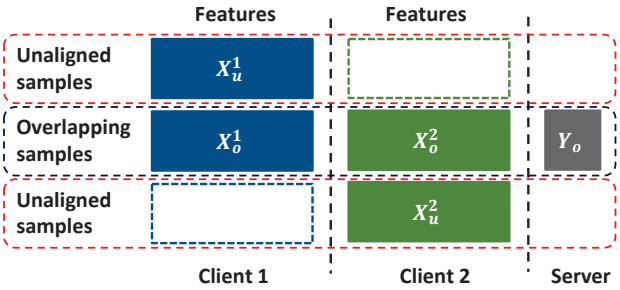

Figure 1. An example of data splitting in a two-client VFL setting.

Vertical federated learning (VFL), on the other hand, focuses on scenarios in which the data on clients have different feature spaces but share some overlapping samples [32]. In addition, the true labels can reside on a third-party server [25] as shown in Fig. 1. For example, a credit bureau collaborates with an e-commerce company and a bank to train a model to estimate a user's credit score. In this case, only the credit bureau has the credit score of the users which will not be shared with the e-commerce company and the bank. VFL is mostly deployed in cross-silo scenarios, and the computation power is usually not a major concern [14]. However, VFL faces two unique challenges. First, VFL requires the clients to communicate with the server for each iteration (rather than after several epochs under HFL) of training, which introduces extremely high

2380-7504/23/$31.00 ©2023 IEEE
DOI 10.1109/ICCV51070.2023.00480

communication costs. It is also notable that iterative communications require reliable communication channels between the server and the clients, which is usually expensive. In addition to the high communication cost, the other major challenge of VFL is that the number of overlapping samples may be limited. For example, two hospitals in different countries are not expected to have a large number of overlapping patients. The model trained with limited overlapping samples likely cannot achieve reliable performance.

Furthermore, VFL is currently not as well explored as HFL. Some existing works can reduce the communication cost by reducing the communication frequency or compressing the communicated data [20]. However, most methods only achieve limited reduction from one local update to multiple while still requiring heavy iterative communications. Other works focus on improving the performance with limited overlapping samples [15, 30]. Notably, both challenges are bottlenecks of applying VFL in realistic scenarios and leaving either one unsolved hinders the deployment of VFL in the real world. To the best of our knowledge, there is no work aiming at solving these two challenges simultaneously.

In this paper, we propose *one-shot* VFL, which is a communication-efficient VFL algorithm that can achieve high performance with minimal overlapping samples. In one-shot VFL, the clients are guided to conduct local semi-supervised learning (SSL) using both the overlapping samples and the unaligned samples to train well-performing feature extractors. Under one-shot setting, the clients only need to conduct two upload operations and one download operation for the training session, which drastically reduces the communication cost and frequency. We further propose *few-shot* VFL as an extension of one-shot VFL. Few-shot VFL expands the supervised dataset on clients to improve the performance of the local feature extractors. Compared with one-shot VFL, clients in few-shot VFL conduct one more time of uploading and downloading but can achieve better performance, especially when the number of overlapping samples is small.

Our key contributions are summarized as follows:

- We propose a communication-efficient VFL algorithm called *one-shot* VFL. To the best of our knowledge, *one-shot* VFL is the first algorithm that can simultaneously address the challenges of high communication cost and limited overlapping samples.
- We propose *few-shot* VFL that can improve the performance further under settings with minimal overlapping samples.
- We empirically evaluate the performance of *one-shot* VFL and *few-shot* VFL with different data modalities, including image data and tabular data. The results show that our methods improve the accuracy by more than 46.5% and reduce the communication cost by more than 330× compared with the state-of-the-art (SOTA) VFL methods on CIFAR-10.

2. Background and Related Work

Vertical Federated Learning. Vertical federated learning (VFL) [25, 28] is the concept of collaboratively training a model on a dataset where the clients share some common samples but with different features on each client. VFL was first introduced in [11], where a federated logistic regression algorithm is proposed. SecureBoost [4] proposed a secure federated tree-boosting approach in the VFL setting and provided theoretical proof that it achieves the same level of accuracy as its centralized counterparts. Some other gradient boosting tree approaches for VFL include Pivot [29] and VF^{2}Boost [7]. A federated random forest was also studied in [21]. In addition to tree-based methods, other machine learning algorithms such as linear regression [34] and logistic regression [12, 20] have been investigated under VFL settings. However, these algorithms are usually incapable of handling complex tasks such as computer vision (CV) and natural language processing (NLP), in which Deep Neural Networks (DNN) are preferred. On the neural network side, SplitNN [28] was proposed to collaboratively train neural networks by splitting a neural network among participants and exchanging gradients and representations in each iteration. FATE [19] implemented a framework that supports DNN in VFL. Even though FATE improves the model's capacity in VFL by supporting DNN, it still requires frequent communication between the participants for each iteration of training and therefore incurs significant communication costs as in previous VFL methods.

To reduce the communication cost, FedBCD [20] was proposed to leverage stale gradients for conducting local training such that participants can decrease the communication frequency from one local update to multiple updates. However, frequent iterative communication is still required for the whole training process. The other challenge is that the common samples across clients are usually limited in the real world, and training under such constraints may not achieve acceptable accuracy. FedCVT [15] proposes to expand the training samples by estimating representations and labels but does not address the bottleneck of communication cost.

In contrast, our proposed one-shot VFL and few-shot VFL are capable of solving the challenges of communication cost and limited common samples simultaneously.

Privacy in VFL. Privacy has become a concern of VFL since it was proposed because the clients need to send representations to the server for training, and privacy protection in VFL is well-explored. [11] presents a secure protocol

that is managed by a third party, the coordinator, by employing privacy-preserving entity resolution and an additive homomorphic encryption scheme. To improve data privacy and model security, FATE [19] applies a hybrid encryption scheme in the forward and backward stages of training. To defend the label inference attack, [22] proposes manipulating the labels following certain rules, which can be seen as a variant of label differential privacy (label DP) [3, 8] in VFL. Our paper focuses on the performance and communication efficiency of VFL. However, our method does not require the clients and the server to share additional information compared with existing VFL methods and is orthogonal to existing privacy-protecting techniques, which can be directly applied to our method.

Semi-supervised Learning (SSL). SSL aims at training a model with partially labeled data, especially when the amount of labeled data is much smaller than the unlabeled ones. There have been many SSL algorithms proposed over the years. SSL algorithms can be broadly categorized as consistency regularization [1, 24], pseudo-label methods [17, 23, 9, 26], and generative models [16, 6]. Consistency regularization is based on the assumption that if a realistic perturbation is applied to a data point, the prediction conducted by the trained model should not change significantly. MixMatch [2] applies consistency regularization along with entropy minimization and generic regularization and can achieve similar accuracy as fully supervised training approaches. Pseudo-labeling has become a component of many recent SSL techniques [23]. Such methods leverage the trained model to generate pseudo labels for the unlabeled data so that the labeled training dataset is expanded. Generative models (e.g., VAE [16]) are trained to generate images from the data distribution and can be transferred to downstream tasks for a given task with targets.

Existing works [13, 35, 36, 31] apply SSL to FL to solve the real problem that the clients may not have enough labeled data. FSSL [13] learns inter-client consistency between multiple clients and splits model parameters for the server with labeled data and clients with unlabeled data separately. SemiFL [5] is the most recent work applying FixMatch [26] to FL to improve the generalization of the global model. However, these methods focus on Horizontal Federated Learning (HFL), where the clients have the same feature space. Our work focuses on VFL settings where most clients have only partial features and no labels. In addition, existing deep SSL methods focus on imaging applications, while VFL has more potential for other types of data, such as tabular or multi-modal models combining imaging with other data types.

3. Problem Definition

Suppose K clients and a server collaboratively train a model. There is an overlapped dataset[1] across all clients with size N_o: $\{x_{o,i}, y_{o,i}\}_{i=1}^{N_o}$. The feature vector $x_{o,i} \in \mathbb{R}^d$ is distributed among K clients $\{x_{o,i}^k \in \mathbb{R}^{d_k}\}_{k=1}^K$, where d_k is the feature dimension of client k. For simplicity, the aligned dataset $\{x_{o,i}^k \in \mathbb{R}^{d_k}\}_{i=1}^{N_o}$ on client k is denoted as X_o^k, and the set $\{X_o^k\}_{k\in[K]}$ is denoted as X_o. Besides X_o^k, each client k also possesses N_k local samples $\{x_{u,i}^k \in \mathbb{R}^{d_k}\}_{i=1}^{N_k}$ which is denoted as X_u^k that is "unaligned" with other clients. The server has the true label of the overlapping samples $\{y_{o,i}\}_{i=1}^{N_o}$ which is denoted as Y_o. An example of data splitting in the two-client setting is shown in Fig. 1.

Each client (says the k-th) learns a representation extractor $f_k(.;\theta_k)$ to extract representations and the server learns a classifier $f_c(.;\theta_c)$ to classify the representations uploaded by clients. The collaborative training problem can be formulated as

$$\min_{\Theta} \mathcal{L}(\Theta; X_o, Y_o) \triangleq \frac{1}{N_o} \sum_{i=1}^{N_o} g(\theta_1, ..., \theta_K, \theta_c; x_{o,i}, y_{o,i}), \quad (1)$$

where $\Theta = [\theta_1; ...; \theta_K; \theta_c]$. $g(.)$ denotes the loss function formulated as:

$$g(\theta_1, ..., \theta_K, \theta_c; x_{o,i}, y_{o,i}) = g\left(f_c\left(h_{o,i}^1 \circ ... \circ h_{o,i}^K\right), y_{o,i}\right), \quad (2)$$

where $\circ$ stands for the concatenation operation, and $h_{o,i}^k$ is the representation extracted by the local model on client k:

$$h_{o,i}^k = f_k(x_{o,i}^k; \theta_k). \quad (3)$$

For simplicity, the set of representations of the aligned data extracted by client k $\{h_{o,i}^k\}_{i=1}^{N_o}$ is denoted as H_o^k. The objective of each party k is to find the optimal θ_k without sharing local data $\{x_{o,i}^k\}_{i=1}^{N_o}$ and parameter θ_k. The objective of the server is to optimize θ_c without sharing θ_c and true labels Y_o.

4. Methods

To reduce the communication cost and improve the model performance under the settings with limited overlapping users, we propose two VFL methods called one-shot VFL and few-shot VFL, respectively.

4.1. One-shot Vertical Federated Learning

We first propose one-shot VFL, in which the clients expect to receive partial gradients from the server only once. The intuition of one-shot VFL is that we can extract sufficient information that will guide clients to conduct local

[1]We assume the alignment between overlapping samples is known as a priori. In some applications private set intersection could be used before running VFL to find the sample alignment.

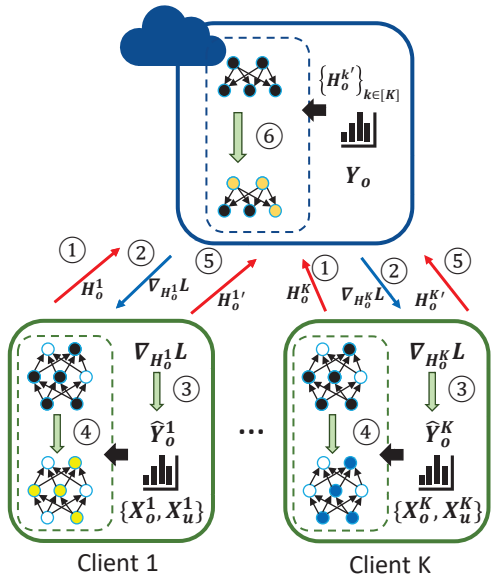

Figure 2. Workflow of one-shot VFL. The clients conduct two times of uploading and one time of downloading.

training from the received partial gradients. The workflow of one-shot VFL is shown in Fig. 2. First, the clients (e.g., the k-th) extract representations of the overlapping data H_o^k and send the representations to the server (①). Then, the server aligns and aggregates the received representations from all clients and computes loss with the true labels. After that, the server conducts back-propagation to compute the partial gradients of local representations $\nabla_{H_o^k} Loss$ and sends the partial gradients and the number of classes C of the global classification task to corresponding (i.e., the k-th) clients (②). After the k-th client receives the partial gradients, it conducts k-means on the partial gradients and assigns the overlapping samples X_o^k temporary labels $\hat{Y}_o^k$ using the clustering index of corresponding gradients (③). The intuition behind clustering is that the partial gradients of the same class should have similar directions while the partial gradients of different classes should have higher diversity. By clustering the partial gradients, the clients can infer information about true labels on the server to guide local training. With the temporary labels assigned to the overlapping samples, the clients conduct semi-supervised learning based on X_u^k and $\{X_o^k, \hat{Y}_o^k\}$ to get updated W_k' (④). After the k-th client completes the local semi-supervised learning, it derives new representations $H_o^{k'}$ by computing $f_k(X_o^k; W_k')$ and sends $H_o^{k'}$ to the server (⑤). Finally, the server aligns and aggregates the received new representations $\{H_o^{k'}\}_k$ to get new global representations H_o' and finetunes the classifier W_c using $\{H_o', Y_o\}$ (⑥).

The detailed algorithm of one-shot VFL is shown in Algorithm 1. It is notable that during the whole training process, the clients only need to upload representations to the server twice and download gradients from the server once, which is the reason we call it one-shot VFL. With such a significant reduction from iterative download/upload to one-shot, we overcome the communication bottleneck in VFL. Meanwhile, local SSL conducted by clients fully utilizes the data of users that are unique to each client and improves the performance under realistic settings with limited overlapping samples.

Algorithm 1 One-shot and few-shot VFL. $mode$ is "few_shot" if the server is executing few-shot VFL; X_o^k and X_u^k are aligned dataset and unaligned dataset of client k; Y_o is the set of true labels of overlapping samples on the server; C is the number of classes in the task; The K clients are indexed by k; B is the minibatch size; E_s and E_c are the number of epochs of the server and clients; η_s and η_c are learning rate of the server and clients; $g(.)$ and $l_{ssl}(.)$ are loss functions defined in Eq. (1) and Eq. (4); Uploading happens in ←; Downloading happens in ←.

Server executes:
initialize θ_c;
for each client $k \in [K]$ **in parallel do**
  $H_o^k \leftarrow f_k(X_o^k; \theta_k)$; ▷ ① in Fig. 2
end for
for $k = 1, ..., K$ **do**
  $\nabla_{H_o^k} Loss \leftarrow \nabla_{H_o^k} g\left(f_c\left(H_{o,i}^1 \circ ... H_{o,i}^K\right), Y_o\right)$; ▷ ② in Fig. 2
end for
for each client $k \in [K]$ **in parallel do**
  ClientUpdate($\nabla_{H_o^k} Loss, C$); ▷ ③④ in Fig. 2
  $H_o^k \leftarrow f_k(X_o^k; \theta_k)$; ▷ ⑤ in Fig. 2
  if $mode ==$ *"few_shot"* **then**
    $H_u^k \leftarrow f_k(X_u^k; \theta_k)$;
  end if
end for
if $mode ==$ *"few_shot"* **then**
  for $k = 1, ..., K$ **do**
    $\{\hat{p}_{u,i}^k\}_{i\in|H_u^k|} \leftarrow$ InferProb $\left(\{H_o^k\}_k, H_u^k\right)$; ▷ Defined in Alg. 2
    ClientUpdateFewshot($\{\hat{p}_{u,i}^k\}_{i\in|H_u^k|}$); ▷ Defined in Alg. 2
    $H_o^k \leftarrow f_k(X_o^k; \theta_k)$;
  end for
end if
$\mathcal{B}_\langle, \mathcal{B}_\dagger \leftarrow$ (split $H_o^1 \circ ... \circ H_o^K$ and Y_o into batches of size B);
for epoch i from 1 to E_s **do**
  for batch $b_h \in \mathcal{B}_\langle, b_y \in \mathcal{B}_\dagger$ **do**
    $\theta_c \leftarrow \theta_c - \eta_s \nabla_{\theta_c} g\left(f_c\left(b_h\right), b_y\right)$; ▷ ⑥ in Fig. 2
  end for
end for
ClientUpdate($\nabla_{H_o^k} Loss, C$):
$\hat{Y}_o^k \leftarrow k\text{-}means(\nabla_{H_o^k} Loss, C)$; ▷ ③ in Fig. 2
$\mathcal{B}_\sqcap, \mathcal{B}_\wr, \mathcal{B}_\dagger \leftarrow$ (split $X_u^k, X_o^k, \hat{Y}_o^k$ into batches of size B);
for epoch i from 1 to E_c **do**
  for batch $b_u \in \mathcal{B}_\sqcap, b_o \in \mathcal{B}_\wr, b_y \in \mathcal{B}_\dagger$ **do**
    $\theta_k \leftarrow \theta_k - \eta_c \nabla_{\theta_k} l_{ssl}\left(\theta_k; b_u, b_o, b_y\right)$; ▷ ④ in Fig. 2
  end for
end for

Local SSL. In one-shot VFL, the clients conduct semi-supervised learning (SSL) based on X_u^k and $\{X_o^k, \hat{Y}_o^k\}$. For different types of data, the detailed SSL algorithms are different. The training objective of the k-th client can be abstracted into

$$l_{ssl}\left(\theta_k; X_u^k, X_o^k, \hat{Y}_o^k\right) = l_s\left(\theta_k; X_o^k, \hat{Y}_o^k\right) + \lambda_u l_u\left(\theta_k; X_u^k\right), \tag{4}$$

where $l_s(.)$ is the supervised training loss, $l_u(.)$ is the unsupervised training loss, λ_u controls the trade-off between the supervised loss and unsupervised loss. In this paper, we focus on two types of data: image data and tabular data, which are common use cases of VFL. Plenty of SSL algorithms [26, 2] have been proposed for image recognition, and we apply FixMatch [26] for the clients conducting SSL on image data, which is a widely applied SSL algorithm in image recognition. On the other hand, in order to fit DL to the task of tabular data SSL, we modify the augmentation methods in FixMatch and propose our FixMatch-tab algorithm for local SSL of tabular data. We modify the weak augmentation

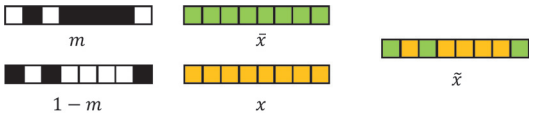

Figure 3. Features are randomly masked for data augmentation.

$\alpha(.)$ and strong augmentation $\mathcal{A}(.)$ in FixMatch to adapt it to the tabular data. For weak augmentation, we randomly generate a binary mask m with the same shape of the data point. Each element of m is sampled from a Bernoulli distribution. We replace the masked elements with the mean value of the corresponding elements of local data. For the strong augmentation, we add noise to the masked samples. Thus, when we train a data point x in FixMatch-tab, we first sample a binary mask m for both weak and strong augmentation and sample a noise vector n for strong augmentation where

$$\begin{aligned} m_i &\sim B(1, r_m), \\ n_i &\sim N(0, \sigma^2), \end{aligned} \tag{5}$$

r_m is the expected ratio of elements that are masked and σ^2 is the variance of the noise. Then, the weak augmentation α and strong augmentation $\mathcal{A}(.)$ for this sample in FixMatch-tab are formulated as

$$\begin{aligned} \alpha(x) &= m \otimes x + (1 - m) \otimes \bar{x}, \\ \mathbb{A}(x) &= \alpha(x) + n, \end{aligned} \tag{6}$$

where $\bar{x} = \frac{1}{N}\sum_{j\in[N]} x_j$ and N is the number of local samples.

4.2. Few-shot Vertical Federated Learning

Even though one-shot VFL can achieve high performance of the global model with extremely low communication cost, we consider improving the performance further by paying with a few more rounds of communication. The key factor to improve SSL performance is to have a larger labeled dataset. Thus, we propose to expand the supervised learning dataset on clients in VFL. One intuitive idea is to assign pseudo labels from local predictions to the unlabeled data points if those predictions have a high confidence. By doing this, we would only need to modify the local training procedure of one-shot VFL without introducing additional communication cost. However, there is one potential problem this method cannot solve. Considering a toy example shown in Fig. 4. Two clients participate in training an image classification task, and each client has access to half of each image. If the two shapes on the image are the same, the image is positive and negative if both shapes are different. If we want to improve the performance by enlarging the labeled dataset,

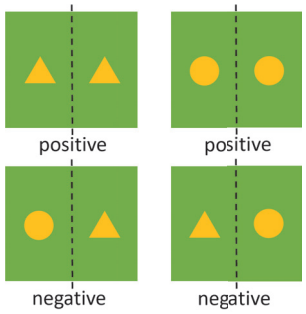

Figure 4. A client does not have enough information to generate reasonable pseudo labels from either the left or right part of the image alone during local semi-supervised learning (SSL).

Algorithm 2 Local SSL training process with expanded labeled dataset in few-shot VFL. $H_o^{[K]\setminus\{k\}}$ stands for $H_o^1 \circ, ..., \circ H_o^{k-1} \circ H_o^{k+1} \circ, ..., \circ H_o^K$

InferProb$\left(\{H_o^k\}_k, H_u^k\right)$**:**
for $k = 1, ..., K$ **do** ▷ This is executed only by once
  $\theta_c^k \leftarrow \arg\min_\theta g\left(f_c^k\left(H_o^k; \theta\right), Y_o\right)$; ▷ Optimize with SGD, ② in Fig. 5
end for
$\theta_c \leftarrow$ execute line 22-27 in Alg. 1; ▷ ② in Fig. 5
$\hat{H}_u^{[K]\setminus\{k\}} \leftarrow T\left(H_u^k, H_o^k, H_o^{[K]\setminus\{k\}}\right)$; ▷ ③ in Fig. 5
for $i = 1, ..., |H_u^k|$ **do**
  $\hat{p}_{u,i}^k \leftarrow$ compute Eq. (8) and Eq. (9); ▷ ④ in Fig. 5
end for
return $\{\hat{p}_{u,i}^k\}_{i\in|H_u^k|}$;
ClientUpdateFewshot$(\{\hat{p}_{u,i}^k\}_{i\in|X_u^k|})$**:**
$X_{uc}^k \leftarrow$ sample from X_u^k with probability $\{\hat{p}_{u,i}^k\}_{i\in|X_u^k|}$; ▷ ⑤ in Fig. 5
$\hat{Y}_{uc}^k \leftarrow f_k\left(X_{uc}^k; \theta_k\right)$; ▷ ⑤ in Fig. 5
$X_o^{k\prime} \leftarrow X_o^k \cup X_{uc}^k$; $\hat{Y}_o^{k\prime} \leftarrow \hat{Y}_o^k \cup \hat{Y}_{uc}^k$;
$X_u^{k\prime} \leftarrow X_o^k \setminus X_{uc}^k$;
$\mathcal{B}_\sqcap, \mathcal{B}_l, \mathcal{B}_\dagger \leftarrow$ (split $X_u^{k\prime}, X_o^{k\prime}, \hat{Y}_o^{k\prime}$ into batches of size B);
for epoch i from 1 to E_c **do**
  for batch $b_u \in \mathcal{B}_\sqcap, b_o \in \mathcal{B}_l, b_y \in \mathcal{B}_\dagger$ **do**
    $\theta_k \leftarrow \theta_k - \eta_c \nabla_{\theta_k} l_{ssl}\left(\theta_k; b_u, b_o, b_y\right)$; ▷ ⑥ in Fig. 5
  end for
end for

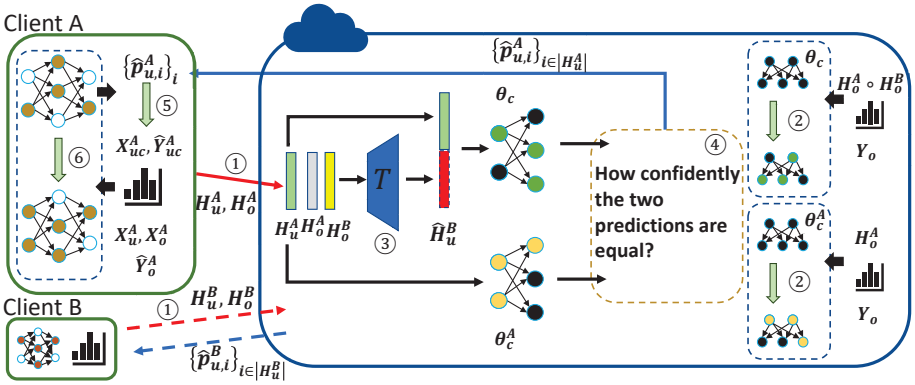

Figure 5. The server judges whether local samples contain enough information to generate accurate pseudo labels in few-shot learning. The clients conduct local SSL with the expanded labeled dataset.

the key is to generate pseudo labels with high accuracy. However, in this toy example, it is impossible for the clients to generate reasonable pseudo-labels based on only half of the images since they do not have enough information to infer the true label of the image.

To solve this problem, we propose few-shot VFL as shown in Algorithm 1. The main difference with one-shot VFL is that the server in few-shot VFL estimates the missing part of the representations for each client's unaligned data which shall expand their labeled datasets. The detailed pipeline of lines 17-18 in Algorithm 1 is shown in Fig. 5. In this section, we focus on VFL with two clients (A & B), but the proposed method can be naturally extended to the scenario where there are more than two clients. When the server receives the representations H_u^A of client A's unaligned data (① in Fig. 5), it estimates the missing representations $\hat{H}_u^B$ of corresponding samples on client B (does not exist for unaligned data) with a transform layer T (③ in Fig. 5) as

$$\hat{H}_u^B = T(H_u^A, H_o^A, H_o^B). \tag{7}$$

The details of $T(.)$ will be introduced later. Then for each unaligned sample (says the i-th), the server produces predictions $\{\hat{y}_{u,i}^A, \hat{y}_{u,i}^{A,B}\}$ and probabilities $\{p_{u,i}^A, p_{u,i}^{A,B}\}$ following

$$\begin{aligned} \hat{y}_{u,i}^A &= \arg\max_j f_c^A(H_{u,i}^A; \theta_c^A)_j, \\ p_{u,i}^A &= \max_j f_c^A(H_{u,i}^A; \theta_c^A)_j, \\ \hat{y}_{u,i}^{A,B} &= \arg\max_j f_c(H_{u,i}^A \circ \hat{H}_{u,i}^B; \theta_c^{A,B})_j, \\ p_{u,i}^{A,B} &= \max_j f_c(H_{u,i}^A \circ \hat{H}_{u,i}^B; \theta_c^{A,B})_j. \end{aligned} \tag{8}$$

where $f_c^A(.)$ is an auxiliary classifier whose input is $h_{u,i}^A$. θ_c^A and $\theta_c^{A,B}$ are trained (② in Fig. 5) based on the overlapping samples. If the predictions $\{\hat{y}_{u,i}^A, \hat{y}_{u,i}^{A,B}\}$ based on local and estimated global representations are the same with high confidence, the local representation $h_{u,i}^A$ contains enough information and should be given a pseudo label during the local SSL on client A. To reduce the noise from misleading pseudo labels, the server sets a probability $\hat{p}_{u,i}^A$ for each unaligned sample to be given a pseudo-label during local training following

$$\hat{p}_{u,i}^A = \mathbb{1}\left(\hat{y}_{u,i}^A = \hat{y}_{u,i}^{A,B}\right) \mathbb{1}\left(p_{u,i}^A > t\right) \mathbb{1}\left(p_{u,i}^{A,B} > t\right) p_{u,i}^{A,B}. \tag{9}$$

The intuition is that the higher the confidence, the local representation $h_{u,i}^A$ is predicted to be the same label as the global representation $h_{u,i}^A \circ \hat{h}_{u,i}^B$, the larger a probability $\hat{p}_{u,i}^A$ is given to the i-th unaligned sample on client A for assigning a pseudo label during local training. In the following, we will introduce the representation transform layer $T(.)$ and local SSL with probability set $\{\hat{p}_{u,i}^A\}_i$.

Efficient Representation Estimation. We design the representation transform layer utilizing a *scaled dot product attention* (SDPA) function formulated as

$$\begin{aligned} \hat{H}_u^B &= T(H_u^A, H_o^A, H_o^B) \\ &= softmax(\frac{H_u^A \otimes {H_o^A}^T}{\sqrt{d}}) \otimes H_o^B, \end{aligned} \tag{10}$$

where $\otimes$ is matrix multiplication operator, and d is the dimension of representation. With $T(.)$, Each missing representation is estimated through the weighted sum over representations of overlapped samples. The weight matrix $W_A = softmax(\frac{H_u^A \otimes {H_o^A}^T}{\sqrt{d}})$ reflects the similarity between the representation to be estimated and the aligned representations in client A.

We apply the SDPA function rather than a generative model (e.g., GAN) to estimate representations for two reasons. First, the generative model has to be trained on the representations of overlapping samples. However, the number of overlapping samples could be too small in real life to train a generator with good performance. Second, when there are K clients, the server needs to train K generators, which introduces heavy computational overhead. By applying the SDPA function to estimate representations, we can overcome the problem of limited overlapping samples and improve the computational efficiency of our estimation.

Local SSL with Expanded Supervised Dataset. After the client A receives $\{\hat{p}^A_{u,i}\}$ (④ in Fig. 5), it samples a subset denoted as X^A_{uc} from X^A_u with probabilities $\{\hat{p}^A_{u,i}\}$. For each sample $x^A_{uc,i}$ in X^A_{uc}, client A assigns the pseudo label $\hat{y}^A_{uc,i}$ as the prediction of the local model that was learned using SSL on client A. For simplicity, the set of pseudo labels $\hat{y}^A_{uc,i}$ is denoted as $\hat{Y}^A_{uc}$. In such a way, client A expands the supervised data via SSL. Hence, the objective of SSL (⑥ in Fig. 5) on client A can be formulated as

$$\begin{aligned} & l_{ssl}\left(\theta; X^A_u \backslash X^A_{uc}, X^A_o \cup X^A_{uc}, \hat{Y}^A_o \cup \hat{Y}^A_{uc}\right) \\ =& l_s\left(\theta; X^A_o \cup X^A_{uc}, \hat{Y}^A_o \cup \hat{Y}^A_{uc}\right) + \lambda_u l_u\left(\theta; X^A_u \backslash X^A_{uc}\right). \end{aligned} \tag{11}$$

5. Evaluation

5.1. Experimental setup

We evaluate our proposed one-shot VFL and few-shot VFL on both image and tabular data. As stated before, we focus on and evaluate two-client scenarios in this paper, which is a common experimental setup in most VFL literature[20, 21]. We compare our methods with two SOTA VFL methods aiming at reducing communication cost and solving the problem of limited overlapping samples, respectively.

Baselines. We compare our proposed algorithm with vanilla VFL and two SOTA VFL methods. (1) **FedBCD [20]** aims at reducing the communication cost. In vanilla VFL, clients conduct one iteration of training after one time of inter-party communication. FedBCD allows clients to conduct multiple iterations of local training using the stale partial gradients of representations received in the last communication. (2) **FedCVT [15]** is a semi-supervised learning approach that improves the performance of VFL using limited overlapping samples. FedCVT leverages representation estimation and pseudo-label prediction to expand the training set to improve the model's representation learning. However, it still suffers from high communication cost.

Datasets. To evaluate our VFL methods under different VFL settings, we use both image data and tabular data for experiments. We use CIFAR-10 for image classification and UCI_credit_card dataset [33] for prediction of default of credit card clients. For CIFAR-10, we follow [20, 15] to split an image into two halves. For UCI_credit_card dataset, we follow FATE [19] to assign ten attributes to one client and the rest to the other client. To mimic the settings in which limited samples are overlapping, we randomly sample N_o samples from the dataset as the aligned dataset. For the rest of the samples we evenly and randomly separate them into two sets, and one client has access to the assigned attributes/halves of images of one set.

Hyperparameter Configurations. To evaluate our methods under settings with different sizes of overlapping samples, we set $N_o = \{256, 512, 1024, 2048\}$ for CIFAR-10 and $N_o = \{1000, 2000\}$ for UCI_default_credit. We set B as 32 for both datasets. Learning rates η_s and η_c are set to be 0.01. For FedBCD, we set Q as 5. We set σ as 0.1 and r_m as 0.2 for tabular data augmentation. For CIFAR-10, we allow the baselines to continue training even after convergence to try to achieve a decent accuracy. For UCI_default_credit, we stop the training of baselines when there is no improvement in accuracy in the last 20 rounds. We use WideResNet20 as the backbone model for CIFAR-10 and a two-layer MLP for UCI_default_credit.

Evaluation metrics. (1) **Utility metric (Accuracy & AUC):** We use the test data accuracy of the classifier on the server to measure the performance of VFL on CIFAR-10. For UCI_default_credit, we apply Area under the ROC Curve (AUC) as the utility metric. A smaller accuracy or AUC means a less practical utility. (2) **Communication metric (Communication cost/times):** We use the times of communication between the clients and the server and the total data volume of communication cost to evaluate the communication efficiency of VFL.

5.2. Experimental results

Accuracy v.s. Communication Cost. The results of CIFAR-10 are shown in Tab. 1. It is shown that compared with vanilla VFL, one-shot VFL improves the accuracy by more than 45% while reducing communication cost by more than 330×. FedBCD reduces communication cost compared with Vanilla VFL. However, it does not improve the accuracy of the model, and the communication reduction is not comparable with one-shot VFL. It is notable that FedCVT cannot achieve significant accuracy improvement compared with vanilla VFL, because the true label is extremely limited in our realistic setting. With extremely limited true labels, it is hard for the server of FedCVT to conduct SSL using only the estimated representations and pseudo labels.

Few-shot VFL improves the accuracy further with higher communication cost compared with one-shot VFL. However, the communication reduction is still significant compared with baselines. In both one-shot VFL and few-shot VFL, clients train representation extractors first, then the server trains the classifier. To improve the accuracy further, we conduct end-to-end vanilla VFL after completing few-shot VFL to finetune the global model on CIFAR-10. It is shown that end-to-end finetuning can improve the accuracy further. Even though the finetuning requires more communication rounds, it is still much more efficient compared with the baselines and offers the clients an option to further improve the performance.

The results of UCI_default_credit dataset are shown in Fig. 6. Even though the task of credit card default detection

Table 1. Results of accuracy and communication on CIFAR-10. Best accuracy is shown in **bold** and best communication cost in ***bold italic***.

Overlap size	256			512			1024			2048		
	Acc (%)	Comm times	Comm cost (MB)	Acc (%)	Comm times	Comm cost (MB)	Acc (%)	Comm times	Comm cost (MB)	Acc (%)	Comm times	Comm cost (MB)
Vanilla VFL	31.47	8000	262	35.33	16000	524	42.71	32000	1047	50.75	64000	2094
FedCVT [15]	31.83	8000	262	35.12	16000	524	42.38	32000	1047	48.2	64000	2094
FedBCD [20]	31.45	1600	53	35.43	3200	105	41.93	6400	209	49.75	12800	419
One-shot VFL	78.23	***3***	***0.79***	81.12	***3***	***1.6***	85.25	***3***	***3.1***	86.13	***3***	***6.3***
Few-shot VFL	78.93	5	26.4	83.03	5	27.2	85.68	5	28.7	87.23	5	31.9
Few-shot VFL +finetune	**80.37**	805	52.6	**84.05**	965	58.6	**86.35**	1805	87.7	**87.49**	2005	97.4

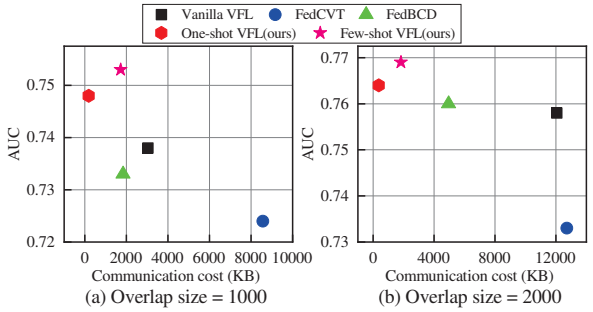

Figure 6. Compared results of AUC v.s. communication cost on UCI_default_credit.

is much simpler than image classification, which does not require a large amount of data to learn, our one-shot VFL can still achieve AUC higher than all the baselines in both settings. One-shot VFL can reduce the communication cost by more than 32×, 33×, and 10× compared with Vanilla VFL, FedCVT, and FedBCD, respectively, under the setting with 2000 overlapping samples. Few-shot VFL increases the communication cost slightly compared with one-shot VFL, but it can improve the AUC further.

Accuracy v.s. Times of Communication. Besides the communication cost, the times of communication needed between the clients are also an important metric to evaluate the communication efficiency. If the clients are required to communicate with the server continually (e.g., vanilla VFL), a stable and reliable communication channel between the server and a client will be necessary. In addition, if a client cannot upload its update to the server in one round of training due to network outage, all the other clients have to wait for it, which is extremely detrimental to the robustness and efficiency of the system. As shown in Tab. 1 and Fig. 7, only three times of communications are needed for one-shot VFL, and the clients conduct SSL locally without waiting for the response from the server, which improves the efficiency significantly. FedBCD reduces the times of communication, but it is still not comparable to one-shot and few-shot VFL. In addition, continual communication between clients are still required for FedBCD, which cannot solve the bottleneck of communication efficiently. Few-shot VFL improves the accuracy further with only two additional times of communication between the clients and the server. During finetuning, the clients need to continually communicate with the server for multiple rounds. However, the clients do not need to communicate during as many rounds as for the other baselines. In practice, our one-shot and few-shot VFL can be used as pre-training techniques to achieve higher performance while significantly reducing the communication cost.

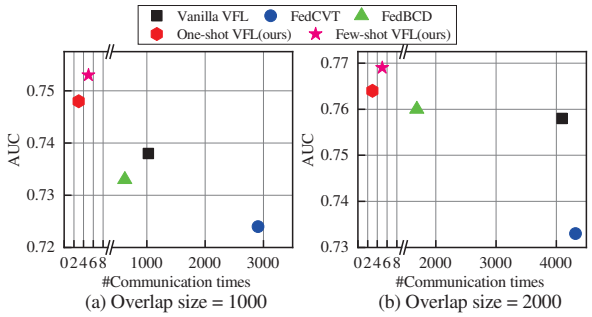

Figure 7. Compared results of AUC v.s. number of communication times on UCI_default_credit.

6. Conclusions and Future Work

In this paper, we propose one-shot VFL that applies SSL to solve two critical problems of VFL: high communication cost and limited overlapping samples common in the real world. In one-shot VFL, the clients only conduct two times of uploading and one time of downloading and can achieve higher accuracy than the SOTA VFL approaches. We also propose few-shot VFL to improve the performance further by paying with one more round of communication. We evaluate our methods on imaging data and tabular data, and the results demonstrate that our methods can improve the model performance under the settings with limited overlapping samples and reduce the communication cost significantly.

In the future, we will evaluate our VFL methods in multi-modal settings combining different data types. In addition,

we note that data privacy preservation is a significant concern of deploying FL in real life [27, 29]. Our methods do not require the clients and the server to share additional information compared with existing VFL methods besides the number of classes, which is common knowledge for both the server and clients in most settings. Existing defense methods [22, 8] can be directly incorporated into our approaches to improve privacy. In this paper, we follow most previous VFL literature [20, 19] and evaluate on two-client scenarios. We will explore multi-party settings in future work. Our code is publicly available[2].

7. Acknowledgements

This work was supported in part by NSF CNS-2112562 and IIS-2140247. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of NSF and its contractors.

References

[1] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 27, 2014. 3

[2] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019. 3, 5

[3] Kamalika Chaudhuri and Daniel Hsu. Sample complexity bounds for differentially private learning. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 155–186. JMLR Workshop and Conference Proceedings, 2011. 3

[4] Kewei Cheng, Tao Fan, Yilun Jin, Yang Liu, Tianjian Chen, Dimitrios Papadopoulos, and Qiang Yang. Secureboost: A lossless federated learning framework. *IEEE Intelligent Systems*, 36(6):87–98, 2021. 2

[5] Enmao Diao, Jie Ding, and Vahid Tarokh. Semifl: Communication efficient semi-supervised federated learning with unlabeled clients. *arXiv preprint arXiv:2106.01432*, 2021. 3

[6] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016. 3

[7] Fangcheng Fu, Yingxia Shao, Lele Yu, Jiawei Jiang, Huanran Xue, Yangyu Tao, and Bin Cui. Vf2boost: Very fast vertical federated gradient boosting for cross-enterprise learning. In *Proceedings of the 2021 International Conference on Management of Data*, pages 563–576, 2021. 2

[8] Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, and Chiyuan Zhang. Deep learning with label differential privacy. *Advances in neural information processing systems*, 34:27131–27145, 2021. 3, 9

[9] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004. 3

[2] https://nvidia.github.io/NVFlare/research/one-shot-vfl

[10] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018. 1

[11] Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Richard Nock, Giorgio Patrini, Guillaume Smith, and Brian Thorne. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *arXiv preprint arXiv:1711.10677*, 2017. 2

[12] Yaochen Hu, Peng Liu, Linglong Kong, and Di Niu. Learning privately over distributed features: An admm sharing approach. *arXiv preprint arXiv:1907.07735*, 2019. 2

[13] Wonyong Jeong, Jaehong Yoon, Eunho Yang, and Sung Ju Hwang. Federated semi-supervised learning with inter-client consistency & disjoint learning. *arXiv preprint arXiv:2006.12097*, 2020. 3

[14] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021. 1

[15] Yan Kang, Yang Liu, and Xinle Liang. Fedcvt: Semi-supervised vertical federated learning with cross-view training. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4):1–16, 2022. 2, 7, 8

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[17] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013. 3

[18] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020. 1

[19] Yang Liu, Tao Fan, Tianjian Chen, Qian Xu, and Qiang Yang. Fate: An industrial grade platform for collaborative learning with data protection. *J. Mach. Learn. Res.*, 22(226):1–6, 2021. 2, 3, 7, 9

[20] Yang Liu, Yan Kang, Xinwei Zhang, Liping Li, Yong Cheng, Tianjian Chen, Mingyi Hong, and Qiang Yang. A communication efficient collaborative learning framework for distributed features. *arXiv preprint arXiv:1912.11187*, 2019. 2, 7, 8, 9

[21] Yang Liu, Yingting Liu, Zhijie Liu, Yuxuan Liang, Chuishi Meng, Junbo Zhang, and Yu Zheng. Federated forest. *IEEE Transactions on Big Data*, 2020. 2, 7

[22] Yang Liu, Zhihao Yi, Yan Kang, Yuanqin He, Wenhan Liu, Tianyuan Zou, and Qiang Yang. Defending label inference and backdoor attacks in vertical federated learning. *arXiv preprint arXiv:2112.05409*, 2021. 3, 9

[23] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018. 3

[24] 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. 3

[25] Daniele Romanini, Adam James Hall, Pavlos Papadopoulos, Tom Titcombe, Abbas Ismail, Tudor Cebere, Robert Sandmann, Robin Roehm, and Michael A Hoeh. Pyvertical: A vertical federated learning framework for multi-headed splitnn. *arXiv preprint arXiv:2104.00489*, 2021. 1, 2

[26] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 3, 5

[27] Jiankai Sun, Xin Yang, Yuanshun Yao, and Chong Wang. Label leakage and protection from forward embedding in vertical federated learning. *arXiv preprint arXiv:2203.01451*, 2022. 9

[28] Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. *arXiv preprint arXiv:1812.00564*, 2018. 2

[29] Yuncheng Wu, Shaofeng Cai, Xiaokui Xiao, Gang Chen, and Beng Chin Ooi. Privacy preserving vertical federated learning for tree-based models. *arXiv preprint arXiv:2008.06170*, 2020. 2, 9

[30] Zhaomin Wu, Qinbin Li, and Bingsheng He. Practical vertical federated learning with unsupervised representation learning. *IEEE Transactions on Big Data*, 2022. 2

[31] Dong Yang, Ziyue Xu, Wenqi Li, Andriy Myronenko, Holger R Roth, Stephanie Harmon, Sheng Xu, Baris Turkbey, Evrim Turkbey, Xiaosong Wang, et al. Federated semi-supervised learning for covid region segmentation in chest ct using multi-national data from china, italy, japan. *Medical image analysis*, 70:101992, 2021. 3

[32] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. *Federated Learning*. Morgan & Claypool Publishers, 2019. 1

[33] I-Cheng Yeh and Che-hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert systems with applications*, 36(2):2473–2480, 2009. 7

[34] Qingsong Zhang, Bin Gu, Cheng Deng, and Heng Huang. Secure bilevel asynchronous vertical federated learning with backward updating. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10896–10904, 2021. 2

[35] Zhengming Zhang, Yaoqing Yang, Zhewei Yao, Yujun Yan, Joseph E Gonzalez, Kannan Ramchandran, and Michael W Mahoney. Improving semi-supervised federated learning by reducing the gradient diversity of models. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 1214–1225. IEEE, 2021. 3

[36] Yuchen Zhao, Hanyang Liu, Honglin Li, Payam Barnaghi, and Hamed Haddadi. Semi-supervised federated learning for activity recognition. *arXiv preprint arXiv:2011.00851*, 2020. 3