Vertical Federated Learning: Concepts, Advances, and Challenges

Yang Liu, *Senior Member, IEEE*, Yan Kang, Tianyuan Zou, Yanhong Pu, *Member, IEEE*, Yuanqin He, Xiaozhou Ye, Ye Ouyang, *Fellow, IEEE*, Ya-Qin Zhang, *Fellow, IEEE*, and Qiang Yang, *Fellow, IEEE*

(Survey)

***Abstract*—Vertical Federated Learning (VFL) is a federated learning setting where multiple parties with different features about the same set of users jointly train machine learning models without exposing their raw data or model parameters. Motivated by the rapid growth in VFL research and real-world applications, we provide a comprehensive review of the concept and algorithms of VFL, as well as current advances and challenges in various aspects, including effectiveness, efficiency, and privacy. We provide an exhaustive categorization for VFL settings and privacy-preserving protocols and comprehensively analyze the privacy attacks and defense strategies for each protocol. In the end, we propose a unified framework, termed VFLow, which considers the VFL problem under communication, computation, privacy, as well as effectiveness and fairness constraints. Finally, we review the most recent advances in industrial applications, highlighting open challenges and future directions for VFL.**

***Index Terms*—Data privacy, feature-partitioned collaborative learning, vertical federated learning.**

I. Introduction

FEDERATED Learning (FL) [1] is a novel machine learning paradigm where multiple parties collaboratively build machine learning models without centralizing their data. The concept of FL was first proposed by Google in 2016 [2] to describe a cross-device scenario where millions of mobile devices are coordinated by a central server while local data are not transferred. This concept is soon extended to a cross-silo collaboration scenario among organizations [3], where a small number of reliable organizations join a federation to train a machine learning model. In [3], FL is, for the first time, categorized into three categories based on how data is partitioned in the sample and feature spaces: Horizontal Federated Learning (HFL), Vertical Federated Learning (VFL) and Federated Transfer Learning (FTL) (See Fig. 1).

- HFL refers to the FL setting where participants share the same feature space while holding different samples. For example, Google uses HFL to allow mobile phone users to use their dataset to collaboratively train a next-word prediction model [2].
- VFL refers to the FL setting where datasets share the same samples/users while holding different features. For example, Webank uses VFL to collaborate with an invoice agency to build financial risk models for their enterprise customers [4].
- FTL refers to the FL setting where datasets differ in both feature and sample spaces with limited overlaps. For example, EEG data from multiple subjects with heterogeneous distributions collaboratively build BCI models using FTL [5].

Due to their differences in data partitions, HFL and VFL adopt very different training protocols. Each party in HFL trains a local model and sends model updates (i.e., parameters or gradients) to a server, which aggregates the updates and sends the aggregating result back to each party. While in VFL, each party keeps its data and model local but exchanges intermediate computed results. The output of the HFL training procedure is a global model shared among all parties, while each party in the VFL owns a separate local model after training. During inference, each party in HFL uses the global model separately, while parties in VFL need to collaborate to make inferences. FL can also be categorized into ”cross-device” and ”cross-silo” settings [6]. The cross-device FL may involve a vast number of mobiles or edge devices as the participating parties, while the participating parties in the cross-silo FL are typically a limited number of organizations. HFL can be either cross-device or cross-silo FL, while VFL typically belongs to the cross-silo FL. We compare main differences between HFL, VFL, and FTL in Table I. Note that Table I compares the conventional cases of HFL, VFL, and

Manuscript received 22 November 2022; revised 9 December 2023; accepted 5 January 2024. Date of publication 26 January 2024; date of current version 10 June 2024. This work was supported in part by the National Key R&D Program of China under Grant 2022ZD0160504, and in part by Tsinghua-Toyota Joint Research Institute inter-disciplinary Program. Recommended for acceptance by G. Wang. *(Corresponding author: Yang Liu.)*

Yang Liu is with the Institute for AI Industry Research, Tsinghua University, Beijing 100190, China, and also with the Shanghai Artificial Intelligence Laboratory, Shanghai 200240, China (e-mail: liuy03@air.tsinghua.edu.cn).

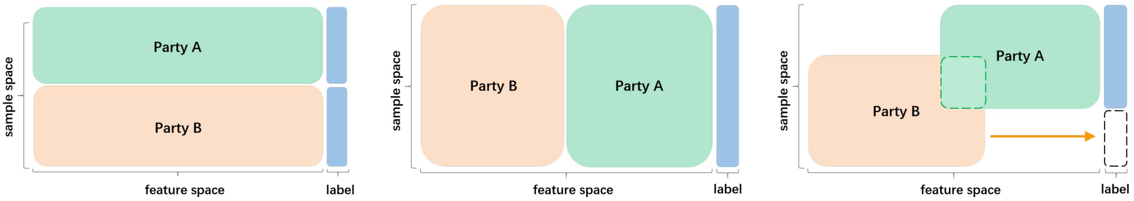
Yan Kang and Yuanqin He are with the Webank, Shenzhen 5180000, China (e-mail: yangkang@webank.com; yuanqinhe@webank.com).

Tianyuan Zou, Yanhong Pu, and Ya-Qin Zhang are with the Institute for AI Industry Research, Tsinghua University, Beijing 100190, China (e-mail: zty22@mails.tsinghua.edu.cn; puyanhong@air.tsinghua.edu.cn; zhangyaqin@tsinghua.edu.cn).

Xiaozhou Ye and Ye Ouyang are with the AsiaInfo Technologies, Beijing 100193, China (e-mail: yexz@asiainfo.com; ye.ouyang@asiainfo.com).

Qiang Yang is with the Webank, Shenzhen 5180000, China, and also with the Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong (e-mail: qyang@cse.ust.hk).

Digital Object Identifier 10.1109/TKDE.2024.3352628



(a) Horizontal Federated Learning (b) Vertical Federated Learning (c) Federated Transfer Learning

Fig. 1. Three categories of Federated Learning.

TABLE I
COMPARISON OF MAIN CHARACTERISTICS BETWEEN CONVENTIONAL HFL, VFL AND FTL

	HFL	**VFL**	FTL
Data is different in	Sample space	Feature space	Both
Scenarios	Cross-device/ Cross-silo	Cross-silo	Mostly Cross-silo
What is exchanged?	Model parameters	Intermediate results	Intermediate results
What is kept local?	Local data	Local data and model	Local data and model
Each party obtains	A shared global model	A local model	A local model
Collaborative Inference?	No	Yes	No

FTL. As this research area experiences explosive growth, some special cases may deviate from Table I.

The need for VFL has arisen and grown strongly in the industry in recent years. Companies and institutions owning only small and fragmented data have constantly been looking for compensating data partners to collaboratively develop artificial intelligence technology for maximizing data utilization [7], [8]. At the same time, data privacy and security regulations have been strengthened worldwide due to growing public concerns over data leakage and privacy breaches. Accordingly, many privacy-preserving projects and platforms supporting VFL have been developed in the past three years [9], [10], [11], [12], [13]. The number of commercialized projects and the economic values of VFL have grown significantly. Since in VFL, data parties are typically from different industrial segments, for example, a bank and a retailer, they are prone to collaborate rather than compete.

While the applications and research on VFL have grown dramatically in recent years, there lacks a comprehensive survey on the advances, challenges, and potential research directions of VFL. Existing FL surveys focus either on HFL [6], [7], [14] or a limited perspective of VFL [3], [15]. Therefore, we provide a comprehensive overview of the current progress in VFL. We propose an exhaustive categorization for VFL settings and privacy-preserving protocols and discuss possible routes for improving effectiveness, efficiency, and privacy. In the end, we propose a unified framework, termed VFLow, which is extended from the original VFL definition and takes into account communication, computation, effectiveness, privacy, and fairness constraints.

This paper is organized as follows: Section II overviews VFL's concepts and training procedures. Building on Sections II, III, IV, and V discuss the efficiency, effectiveness, privacy, and security aspects of VFL algorithms. Section VI discusses the challenges of data valuation, explainability, and fairness towards building a VFL ecosystem. Section VII introduces *VFLow*, a VFL optimization framework guiding the design and optimization of VFL algorithms, and Section VIII discusses application-oriented algorithms built on VFL. Finally, Section IX discusses open challenges and future directions. Fig. 2 dictates the relationships between sections in this work.

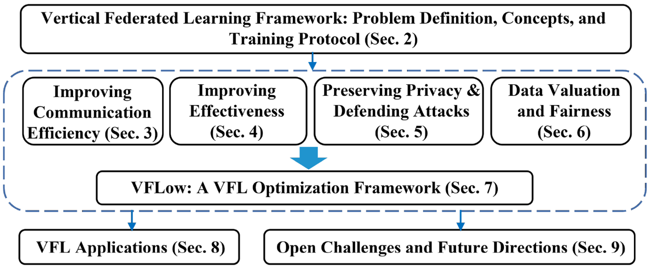

Vertical Federated Learning Framework: Problem Definition, Concepts, and Training Protocol (Sec. 2)
Improving Communication Efficiency (Sec. 3)
Improving Effectiveness (Sec. 4)
Preserving Privacy & Defending Attacks (Sec. 5)
Data Valuation and Fairness (Sec. 6)
VFLow: A VFL Optimization Framework (Sec. 7)
VFL Applications (Sec. 8)
Open Challenges and Future Directions (Sec. 9)

Fig. 2. Relationships between sections in this work.

II. VERTICAL FEDERATED LEARNING FRAMEWORK

In this section, we provide an overview of VFL formulation, variants, and algorithms.

A. *Problem Definition*

A VFL system aims to collaboratively train a joint machine learning (ML) model using a dataset $\mathcal{D} \triangleq \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ with N samples while preserving the privacy and safety of local data and models. We formulate the loss of VFL as follows.

$$\min_{\boldsymbol{\Theta}} \ell(\boldsymbol{\Theta}; \mathcal{D}) \triangleq \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{\Theta}; \mathbf{x}_i, y_i) + \lambda \sum_{k=1}^{K} \gamma(\boldsymbol{\Theta}) \quad (1)$$

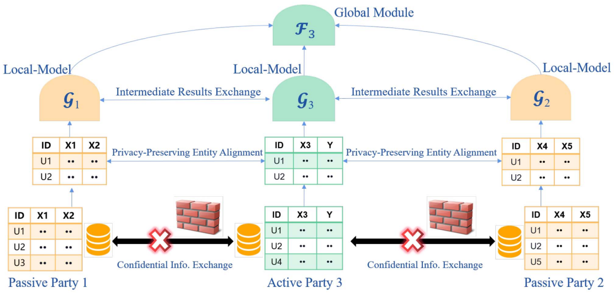
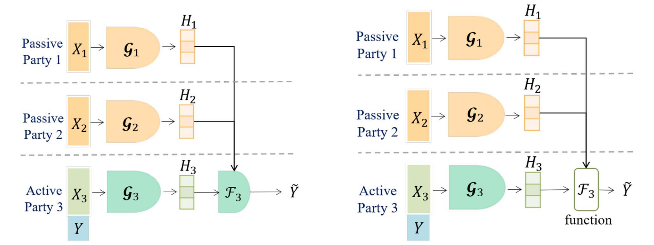


Fig. 3. Illustration of the VFL system with three parties (two passive parties and one active party). $\mathcal{G}_1$, $\mathcal{G}_2$, and $\mathcal{G}_3$ denote the local models of the three parties, respectively, and $\mathcal{F}_3$ denotes the global module owned by the active party. The VFL training protocol typically involves two steps: 1) the three parties align their samples via private entity alignment; 2) the three parties collaboratively train $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$ and $\mathcal{F}_3$ in a privacy-preserving manner (see Section II-B for details).



(a) The global module is trainable and the active party has features (**splitVFL**)

(b) The global module is non-trainable and the active party has features (**aggVFL**).

Fig. 4. Two major variants of VFL illustrated with one active party and two passive parties. These variants are defined based on whether the global model $\mathcal{F}_3$ is trainable or not.

where $\mathbf{\Theta}$ denote the joint ML model; $f(\cdot)$ and $\gamma(\cdot)$ denote the loss function and regularizer and λ is the hyperparatemer that controls the strength of γ.

VFL assumes that data are partitioned by feature space. Following [3], [16], each feature vector $\mathbf{x}_i \in \mathbb{R}^{1\times d}$ in $\mathcal{D}$ is distributed among K parties $\{\mathbf{x}_{i,k} \in \mathbb{R}^{1\times d_k}\}_{k=1}^{K}$, where d_k is the feature dimension of party k, for $k \in [K-1]$, and the K^{th} party has the label information $y_i = y_{i,K}$. We refer to the K^{th} party who owns the labels as *active party* while the rest of parties as *passive parties*. Each passive party k has dataset $\mathcal{D}_k \triangleq \{\mathbf{x}_{i,k}\}_{i=1}^{N}$, while the active party has dataset $\mathcal{D}_K \triangleq \{\mathbf{x}_{i,K}, y_{i,K}\}_{i=1}^{N}$. Without loss of generality, we decompose $\mathbf{\Theta}$ into local models $\mathcal{G}_k$ parameterized by θ_k, $k \in \{1,\ldots,K\}$, which operates only on local data, and a global module $\mathcal{F}_K$ parameterized by ψ_K, which is only accessible by the active party K. We rewrite the loss $f(\mathbf{\Theta}; \mathbf{x}_i, y_i)$ as:

$$
\begin{aligned}
& f(\mathbf{\Theta}; \mathbf{x}_i, y_i) \\
& = \mathcal{L}\left(\mathcal{F}_K\left(\psi_K; \mathcal{G}_1(\mathbf{x}_{i,1}, \theta_1), \ldots, \mathcal{G}_K(\mathbf{x}_{i,K}, \theta_K)\right), y_{i,K}\right) \quad (2)
\end{aligned}
$$

where $\mathcal{L}$ denotes the task loss (e.g., mean squared error loss, cross-entropy loss, and hinge loss).

Fig. 3 pictorially overviews the architecture and core components of a VFL system. Each party's local data are not exchanged during the collaboration. The local model $\mathcal{G}_k$ can take various forms including tree [17], linear and logistic regression [3], [16], [18], [19], [20], [21], support vector machine [22], [23], neural network (NN) [24], [25], [26], K-means [27] and EM algorithm [28].

The global module $\mathcal{F}_K$ can be either *trainable* [26], [29], [30] or *non-trainable* [26], [31]. If a *trainable* global module is in place, this VFL scenario is coincident with the vertical splitNN [32], where the whole model is split into different parties, thus we term it *splitVFL* (see Fig. 4(a)). If the global module is *non-trainable*, it serves as an aggregation function, such as the Sigmoid for NN or the optimal split finding function for tree, that aggregates parties' intermediate results. We term this scenario *aggVFL* (see Fig. 4(b)).

Another variant of VFL is when the active party provides no features. In this variant, the active party plays the role of a central server (like the one in HFL). We refer to the scenarios where the active party provides no features in splitVFL and aggVFL, respectively, as *splitVFL*$_c$ and *aggVFL*$_c$.

In a typical VFL system, passive parties communicate only with the active party, which serves as the coordinator that orchestrates the training and inference procedures. In some scenarios, a third party is involved and responsible for encryption and decryption [16].

B. VFL Training Protocol

In this section, we describe a general training protocol for VFL, which consists of two steps: 1) Privacy-Preserving Entity Alignment; 2)Privacy-preserving Training (Fig. 3).

Privacy-Preserving Entity Alignment: The very first step for a VFL system to start a collaborative training process is to align the data used for the training. This process can be referred to as entity alignment, which adopts private set intersection techniques to find the common sample IDs without revealing unaligned dataset. We discuss these techniques in Section V. Whereas conventional VFL frameworks mostly consider entity alignment with exact IDs, recent studies [33] also demonstrate a coupled design for fuzzy identifiers to enable one-to-many alignment, which could be an interesting future direction of VFL.

Privacy-Preserving Training by Exchanging Intermediate Results: After the alignment, participating parties can start training the VFL model using the aligned samples. The most common training protocol is using gradient descent [34], which requires parties to transmit local model outputs and corresponding gradients, together termed intermediate results, instead of local data. Algorithm 1 describes a general VFL training procedure based on neural networks using stochastic gradient descent (SGD). Specifically, each party k computes its local model output $H_k = \mathcal{G}_i(\mathbf{x}_k, \theta_k)$ on a mini-batch of samples $\mathbf{x}$ and sends H_k to the active party. With all the $\{H_k\}_{k=1}^{K}$, the active party computes the training loss following (1). Then, the active party computes the gradients $\frac{\partial \ell}{\partial \psi_K}$ of its global module ψ_K and updates ψ_K using $\frac{\partial \ell}{\partial \psi_K}$. Next, the active party computes the gradients $\frac{\partial \ell}{\partial H_k}$ for each party and transmits them back. Finally, each party k

Algorithm 1: A General VFL Training Procedure.

```
Input: learning rates $\eta_1$ and $\eta_2$
Output: Model parameters $\theta_1$, $\theta_2$... $\theta_K$, $\psi_K$
Party 1,2,...,$K$, initialize $\theta_1$, $\theta_2$,... $\theta_K$, $\psi_K$.
for each iteration $j = 1, 2, \ldots$ do
    Randomly sample a mini-batch of samples $\mathbf{x} \subset \mathcal{D}$
    for each party $k$=1,2,...,$K$ in parallel do
        Party $k$ computes $H_k = \mathcal{G}_k(\mathbf{x}_k, \theta_k)$;
        Party $k$ sends $\{H_k\}$ to party $K$;
    end for
    Active party $K$ updates $\psi_K^{j+1} = \psi_K^j - \eta_1 \frac{\partial \ell}{\partial \psi_K}$;
    Active party $K$ computes and sends $\frac{\partial \ell}{\partial H_k}$ to all other parties;
    for each party $k$=1,2,...,$K$ in parallel do
        Party $k$ computes $\nabla_{\theta_k} \ell$ with (3);
        Party $k$ updates $\theta_k^{j+1} = \theta_k^j - \eta_2 \nabla_{\theta_k} \ell$;
    end for
end for
```

computes the gradient of its local model θ_k as follows:

$$\nabla_{\theta_k} \ell = \frac{\partial \ell}{\partial \theta_k} = \sum_i \frac{\partial \ell}{\partial H_{i,k}} \frac{\partial H_{i,k}}{\partial \theta_k} \tag{3}$$

and updates θ_k. This procedure iterates until convergence.

To prevent privacy leakage from the intermediate results H_k and gradients $\frac{\partial \ell}{\partial H_k}$, Crypto-based privacy-preserving techniques such as Homomorphic Encryption (HE) (denoted as $[[\cdot]]$), Secure Multi-Party Computation (MPC) and Trusted Execution Environment (TEE) can be introduced into the VFL protocol to protect the crucial information from inner and outside attackers. For example, instead of sending H_k, each party k sends $[[H_k]]$ to the active party, who in turn sends $[[\frac{\partial \ell}{\partial H_k}]]$ back to each party. A third-party collaborator is often responsible for encryption and decryption. Other privacy-preserving techniques, such as Differential Privacy (DP) and Gradient Discretization (GD) can also be applied to enhance the privacy and security of the VFL system. We provide detailed comparisons of these techniques in Section V.

C. Tree-Based VFL

Tree-based VFL differs from the NN-based VFL in local models $\mathcal{G}_k, k \in \{1, \ldots, K\}$, the global module $\mathcal{F}_K$ as well as the training process at each party, while it complies with the architecture depicted in Fig. 3 and follows the general loss defined in (2) for conducting VFL training.

In tree-based VFL, the local model $\mathcal{G}_k$ at each party k consists of multiple partial tree models that each partial tree model, together with its counterparts from other parties, form a complete tree model. The $\mathcal{F}_K$ is an aggregation function that identifies the optimal feature split based on feature splitting information received from all parties. The literature has proposed various GBDT-based VFL algorithms [17], [35], [36], [37], [38], [39], [40]. Random Forest [41] (RF) is another popular tree-based ensemble algorithm that has been integrated into VFL.

III. Improving Communication Efficiency

In production VFL, network heterogeneity, long geographical distances, and the large size of encrypted data make the coordination a communication bottleneck. Thus, methods proposed to mitigate communication overhead typically involve reducing the cost of coordination and compressing the data transmitted between parties. We summarize these methods in Table II and discuss them in this section.

A. Multiple Client Updates

One way to save communication costs is by allowing parties to perform multiple local updates during each iteration.

FedBCD [16] allows each party to conduct multiple client updates before each communication to reduce the number of synchronizations, thereby mitigating the communication overhead. Flex-VFL [42] and AdaVFL [43] allow each party to conduct a different number of local updates by putting time constraints on each communication round and the whole training process, respectively. CELU-VFL [45] implements multiple local updates using cached statistics. Such methods typically require proper choice of training parameters, e.g. learning rate, to improve convergence and exhibit trade-off between computational resources and communication efficiency.

B. Asynchronous Coordination

The core idea of asynchronous coordination is that each party can upload and download intermediate results asynchronously to reduce idle time. However, asynchronous coordination may result in stale information, which may harm the overall model performance and jeopardize communication efficiency if the stale information is not dealt with properly.

GP-AVFL [46] and FDML [25] allow parties to update local models asynchronously. They mitigate the impact of stale information by predicting gradients and enforcing a bounded delay condition, respectively. AVFL [47] and T-VFL [48] achieve asynchronous training and address staleness by removing unstable and insignificant updates, respectively. VAFL [49] implements asynchronous coordination by utilizing a query-response strategy that decouples the coordination between the server and clients. AsySQN [51], VFB^2 [52], and FDSKL [53] all utilize a tree-structured communication scheme [67] to enhance the communication efficiency. FedGBF [54] and VF ^{2} Boost [55] improve efficiency by exploiting parallelism to build decision trees for boosting.

Asynchronous coordination may incur additional computation overhead for handling inconsistencies between the asynchronous updates. Thus, trade-offs between coordination and computation overhead should be carefully considered when applying asynchronous coordination methods.

C. One-Shot Communication

One-shot communication alleviates communication overhead by coordinating only once during the entire training procedure.

TABLE II

Summary of Existing Works That Aim to Improve the Communication Efficiency of VFL. In the Model Column, the LR, NN, XGB, GBT, KNN, and SVM Denote Logistic Regression, Neural Network, XGBoost, Gradient Boosting Decision Tree, K-Nearest Neighbor, and Support Vector Machine, Respectively. In the Convergence Rate Column, T Represents the Total Number of Local Iterations, and Δ Represents Stochastic Variance

Category	Existing Work	VFL Setting	Model	Convergence Rate	Core Method
Multiple Client Updates	FedBCD [18]	splitVFL	LR, NN	$O(1/\sqrt{T})$	Block coordinate descent w/ multiple local updates
	Flex-VFL [44]	splitVFL$_c$	NN	$O(1/\sqrt{T})$	Customized # of local updates constrained by time
	AdaVFL [45]	aggVFL$_c$	NN	$O(1/\sqrt{T})$	Customized # of local updates through optimization
	VIMADMM [46]	splitVFL	NN	-	Alternative direction method of multipliers
	CELU-VFL [47]	splitVFL	NN	$O(\Delta/\sqrt{T})$	Cache-based mechanism for local updates
Asynchronous Coordination	GP-AVFL [48]	aggVFL	LR, NN	-	Asynchronous training with gradient prediction
	AVFL [49]	aggVFL	LR	-	Backup-based straggler-resilient scheme
	T-VFL [50]	splitVFL$_c$	NN	$O(1/\sqrt{T})$	Channel-aware user scheduling poicy
	VAFL [51]	splitVFL$_c$	LR, NN	$O(1/\sqrt{T})$	Asynchronous query-response strategy
	FDML [27]	aggVFL$_c$	LR, NN	$O(1/\sqrt{T})$	Asynchronous local updates w/ bounded delay
	AFAP [52]	aggVFL	LR	$O(e^{-T})$	Tree-structured asynchronous communication (TSAC)
	AsySQN [53]	aggVFL	LR	$O(e^{-T})$	TSAC & quasi-Newton method
	VFB^2 [54]	aggVFL	LR	$O(e^{-T})$	TSAC & bi-level parallel update
	FDSKL [55]	aggVFL	LR	$O(1/T)$	TSAC & random features & doubly stochastic gradient
	FedGBF [56]	aggVFL	GBT	-	Build decision trees in parallel for learning GBDT
	VF^{2}Boost [57]	aggVFL	GBT	-	Concurrent protocol & customized Paillier HE
One-Shot Communication	FedOnce [58]	splitVFL	NN	-	Unsupervised learning by predicting noise
	AE-VFL [59]	splitVFL	NN	-	Unsupervised learning using autoencoder
	CE-VFL [60]	splitVFL	NN	-	Unsupervised learning using PCA & autoencoder
Compression	AVFL [49]	aggVFL	LR	-	Principle component analysis
	CE-VFL [60]	splitVFL	NN	-	Autoencoder and principle component analysis
	SecureBoost+ [37]	aggVFL	XGB	-	Encode encrypted first-order and second-order gradients into a single message
	eHE-SecureBoost [61]	aggVFL	XGB	-	
	C-VFL [62]	splitVFL	NN	$O(1/\sqrt{T})$	Arbitrary embedding compression scheme
	GP-AVFL+DESC [48]	aggVFL	LR, NN	-	Double-end sparse compression
Sample and Feature Selection	Coreset-VFL [63]	aggVFL$_c$	LR, K-Mean	-	Coreset to select samples
	FedSDG-FS [64]	splitVFL$_c$	NN	-	Stochastic dual-gate to select features
	SFS-VFL [65]	aggVFL$_c$	LR, KNN, SVM, GBT	-	Gini impurity to select features
	LESS-VFL [66]	splitVFL$_c$	NN	-	Group lasso regularization to select features
	FEAST [67]	aggVFL	LR, SVM, XGB, NN	-	Conditional mutual information to select features
	VFLFS [68]	splitVFL	NN	-	Use trainable transformation matrix to select features

All proposed one-shot communication methods follow a two-step training procedure: (1) All parties learn latent representations from their original data; (2) The active party trains the global model using these latent representations.

In FedOnce [56], each party extracts latent representations from its local data using an unsupervised learning method called NAT (Noise As Targets) [68]. Then, the active party trains a global model using its local features combined with latent representations passed from passive parties. AE-VFL [57] leverages autoencoder to extract latent representations from each party's local data, while CE-VFL [58] utilizes both Principal Component Analysis (PCA) and autoencoder to conduct the latent representation extraction.

A trade-off for one-shot methods is that sample-wise representations of original data are permanently passed on to the active party. Therefore, the privacy risks of revealing these representations need to be carefully evaluated. Besides, one-shot methods typically involve computationally expensive unsupervised learning of effective representations, therefore the trade-off between communication and computation is worth investigating.

D. Compression

Compression is a widely used approach in VFL to reduce the amount of data transmitted among parties. It can alleviate both communication and computation overheads, especially when expensive encryption operations (e.g., HE and MPC) are applied.

Neural network-based VFL algorithms naturally map high-dimensional input vectors to low-dimensional representations. Some works adopt specialized dimension-reduction techniques to compress data. AVFL [47] leverages Principle Component Analysis (PCA) to compress transmitted data, while CE-VFL [58] utilizes both PCA and Autoencoders to learn latent representations from raw data. Two follow-up works of SecureBoost, SecureBoost+ [35] and eHE-SecureBoost [59] encode encrypted first-order and second-order gradients into a single message to reduce the encryption operations and the size of data transmitted between parties, thereby saving communication bandwidth and computational costs. C-VFL [60] allows an arbitrary compression scheme to be applied to embeddings transmitted between parties to enhance communication efficiency. GP-AVFL [46] employs a double-end sparse compression (DESC) technique to save communication costs by squeezing the sparsity in transmitted information. Adaptive quantization techniques [69], [70], [71] may also be considered in future VFL research.

E. Sample and Feature Selection

Another approach to improve communication efficiency is to reduce the amount of data used for training and inference. For

TABLE III

SUMMARY OF EXISTING WORKS THAT AIM TO IMPROVE THE EFFECTIVENESS OF VFL. SEMI-SL, SELF-SL, KD, AND TL REPRESENT SEMI-SUPERVISED LEARNING, SELF-SUPERVISED LEARNING, KNOWLEDGE DISTILLATION, AND TRANSFER LEARNING, RESPECTIVELY. √ INDICATES ITS CORRESPONDING PORTION OF DATA (SEE FIG. 5) IS UTILIZED BY A SPECIFIC VFL ALGORITHM. NOTE THAT VFED-SSD HAS TWO OBJECTIVES: ONE IS TO BUILD A LOCAL PREDICTOR FOR THE ACTIVE PARTY, AND ANOTHER IS TO BUILD A JOINT PREDICTOR

Core Approach	Objective	Existing Work	Data Used					Method	Party
			Aligned		Unaligned				
			$\mathcal{D}^{au}$	$\mathcal{D}$	$\mathcal{D}_B^{uu}$	$\mathcal{D}_A^{uu}$	$\mathcal{D}_A^{ul}$		
Self-SL	Build a joint predictor	VFLFS [68]	-	√	√	√	-	Generative Models	≥ 2
		VFed-SSD [74]	√	√	-	-	-	Contrastive Learning	2
		FedHSSL [75]	√	√	√	√	-	Contrastive Learning	≥ 2
		SS-VFNAS [76]	-	√	√	√	-	Contrastive Learning	≥ 2
Semi-SL		FedCVT [77]	-	√	√	-	√	Feature & Label Estimation	2
		FedMC [78]	-	√	√	-	√	Data Collaboration	2
KD	Build a local predictor for active party A	VFedTrans [79]	√	-	-	-	√	FedSVD & Representation Distillation	≥ 2
		VFL-Infer [80]	-	√	-	-	-	Model Distillation	2
		VFed-SSD [74]	√	√	-	-	-	Model Distillation	2
		VFL-JPL [81]	-	√	-	-	√	Feature Estimation & Model Distillation	2
TL	Build a local predictor for passive party B	MMVFL [82]	-	√	-	-	-	Feature Selection & Label Transfer	≥ 2
		SFHTL [83]	-	√	√	-	√	Feature & Label Transfer	≥ 2
		SFTL [84], [85]	√	√	-	-	√	Feature Transfer	2
		PrADA [86]	√	√	-	-	-	Adversarial Domain Adaptation	3

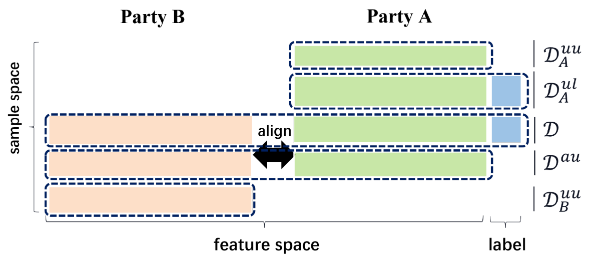


Fig. 5. Virtual dataset of a two-party VFL. $\mathcal{D}$ denotes the labeled and aligned samples used by the conventional VFL formulated in (1), whereas $\mathcal{D}^{au}$ denotes aligned but unlabeled samples. $\mathcal{D}_A^{uu}$ and $\mathcal{D}_B^{uu}$ denote unaligned and unlabeled samples of party A and party B, respectively. $\mathcal{D}_A^{ul}$ denotes unaligned and labeled samples of party A.

example, Coreset-VFL [61] constructs a coreset of samples to alleviate the communication burden, while FedSDG-FS [62], SFS-VFL [63], LESS-VFL [64], FEAST [65] and VFLFS [66] filter out unimportant features to save communication costs.

IV. IMPROVING EFFECTIVENESS

Conventional VFL is only able to utilize aligned labeled samples. However, real-world applications often have limited *aligned samples*, especially as the number of parties grows. The availability of labeled samples is also scarce in many cases, resulting in unsatisfactory performance. Moreover, the collaborative inference is required since each party only has a sub-model after training.

To address these limitations, the literature has proposed various directions toward better utilizing available data to build a joint VFL model or helping participating parties build local predictors. For brevity, we discuss existing works through a two-party VFL setting, involving an *active party* A and a *passive party* B. To better explain these works, we depict a general virtual dataset formed by the two parties (see Fig. 5). We dissect this virtual dataset into several sub-datasets to illustrate which portions of the virtual dataset are utilized by a VFL algorithm to train models. We summarize existing works in Table III and explain them based on their respective learning approaches in the following.

A. Self-Supervised Approaches

Recently, self-supervised learning (Self-SL) has been introduced to VFL to improve the performance of the VFL model by exploiting unlabeled samples, which are not used in the conventional VFL. For illustrative purposes, we consider a two-party VFL scenario and rewrite (1) as follows:

$$\min_{\psi_A,\theta_A,\theta_B} \ell_{\text{VFL}}(\psi_A,\theta_A,\theta_B;\mathcal{D}) \tag{4}$$

Self-SL-based VFL approaches proposed in the literature typically train participating parties' models ψ_A, θ_A, and θ_B by minimizing a Self-SL loss based on unlabeled samples in addition to the main task loss defined in (4). We formulate a general Self-SL objective in VFL as follows:

$$\begin{aligned}&\tilde{\psi}_A,\tilde{\theta}_A,\tilde{\theta}_B\\&= \underset{\psi_A,\theta_A,\theta_B}{\text{argmin}}\ \ell_{\text{Self-SL}}(\psi_A,\theta_A,\theta_B;\mathcal{D}^{au},\mathcal{D}_A^{uu},\mathcal{D}_B^{uu})\end{aligned} \tag{5}$$

where $\ell_{\text{Self-SL}}$ is the self-supervised learning loss that optimizes ψ_A, θ_A and θ_B using unlabeled data.

VFed-SSD [72] pretrains local models ψ_A,θ_A and θ_B through (5) based on contrastive learning using positive and negative sample pairs formed by the matched pair detection technique from aligned data $\mathcal{D}^{au}$. Then, VFed-SSD finetunes pretrained models $\tilde{\psi}_A,\tilde{\theta}_A$ and $\tilde{\theta}_B$ through (4) based on labeled and aligned samples $\mathcal{D}$. FedHSSL [73] pretrains θ_A and θ_B through (5) based on contrastive learning leveraging cross-party views of aligned samples $\mathcal{D}^{au}$ and local views of unlabeled local samples $\mathcal{D}_A^{uu}$ and $\mathcal{D}_B^{uu}$. Then, FedHSSL finetunes ψ_A and pretrained models $\tilde{\theta}_A$ and $\tilde{\theta}_B$ through (4) based on $\mathcal{D}$. SS-VFNAS [74] leverages contrastive learning using local data $\mathcal{D}_A^{uu}$ and $\mathcal{D}_B^{uu}$ to first pretrain local models θ_A and θ_B, and then it performs a federated neural architecture search based on θ_A and θ_B. VFLFS [66] optimizes (4) and (5) in an end-to-end manner.

It trains local models θ_A and θ_B using autoencoders based on unaligned data $\mathcal{D}_A^{uu}$ and $\mathcal{D}_B^{uu}$, and simultaneously finetunes these local models and the global module ψ_A based on labeled aligned samples $\mathcal{D}$.

B. Semi-Supervised Approaches

Rather than boosting representation learning capability leveraging self-supervised learning, FedCVT [75] and FedMC [76] utilize semi-supervised learning approaches that augment labeled and aligned samples $\mathcal{D}$ to boost the performance of the VFL model. We formulate a general Semi-SL-based VFL objective as follows:

$$\min_{\psi_A,\theta_A,\theta_B,\tilde{\mathcal{D}}} \ell_{\text{VFL}}(\psi_A,\theta_A,\theta_B;\tilde{\mathcal{D}}) + \lambda \ell_{\text{Semi-SL}}(\psi_A,\theta_A,\theta_B;\mathcal{D},\mathcal{D}_A^{ul},\mathcal{D}_B^{uu}) \quad (6)$$

where $\ell_{\text{Semi-SL}}$ is the semi-supervised learning loss that aims to expand $\mathcal{D}$ by pseudo-labeling unlabeled samples or adding newly labeled samples while achieving maximal stability and precision on labeling newly added samples.

More specifically, FedCVT estimates representations for missing features and predicts pseudo-labels for unlabeled samples to obtain an expanded training set, denoted as $\tilde{\mathcal{D}}$. To improve the quality of $\tilde{\mathcal{D}}$, FedCVT cherry-picks pseudo-labeled samples added to $\tilde{\mathcal{D}}$ through an ensemble approach. Then, FedCVT uses $\tilde{\mathcal{D}}$ to conduct VFL through (6). FedMC [76] integrates data collaboration [84] into VFL to implement (6). It first uses $\mathcal{D}$ to form a latent subspace, in which the distance between each pair of unaligned samples from the active party and the passive party, respectively, is measured. Then, FedMC aligns two samples in a pair and adds aligned samples to $\mathcal{D}$ if their distance is less than a threshold to form an expanded training set $\tilde{\mathcal{D}}$. Next, FedMC trains the VFL model based on $\tilde{\mathcal{D}}$.

C. Knowledge Distillation-Based Approaches

In conventional VFL, the active party A cannot make inferences alone, which limits the availability of the active party's prediction service. Some studies [72], [77], [78], [79] propose methods to help party A build a local predictor instead of a VFL model while still benefiting from VFL training. To this end, they typically leverage Knowledge Distillation (KD) techniques to transfer knowledge of teacher models obtained through VFL to party A's local models for enhancing performance. We formulate a general knowledge distillation-based VFL objective as follows.

$$\min_{\psi_A^s,\theta_A^s} \ell_A(\psi_A^s,\theta_A^s;\mathcal{D}_A^{ul}) + \lambda \ell_{\text{KD}}(\psi_A^s,\theta_A^s,\psi_A^t,\theta_A^t,\theta_B^t;\mathcal{D}^{au}) \quad (7)$$

where ℓ_{KD} is the knowledge distillation loss that forces to transfer knowledge from teacher models ψ_A^t, θ_A^t and θ_B^t to party A's local models ψ_A^s and θ_A^s, ℓ_A is party A's task loss that optimizes ψ_A^s and θ_A^s based on labeled samples $\mathcal{D}_A^{ul}$, and γ is the hyperparameter that controls the strength of KD. ψ_A^t, θ_A^t and θ_B^t can be pretrained through (4) or (5).

VFL-Infer [78] and VFed-SSD [72] pretrain teacher models ψ_A^t, θ_A^t and θ_B^t through (4) and (5), respectively. Then, they both leverage teacher models to help party A train its local models ψ_A^s and θ_A^s through (7). By utilizing the feature estimation and ranking consistency restriction, VFL-JPL [79] trains a local model for the active party A through (7) based on knowledge distilled from teacher models pretrained through (4). VFedTrans [77] first learns federated representations from aligned samples $\mathcal{D}^{au}$ leveraging FedSVD [85], and then it utilizes autoencoders to transfer the knowledge encoded in the federated representations to the active party A's local models ψ_A^s and θ_A^s.

D. Transfer Learning-Based Approaches

Transfer-learning (TL) based VFL approaches [24], [80], [81], [82], [83] treat the active party A as the source domain with a large corpus of labeled samples and the passive party B as the target domain with only unlabeled samples or a limited amount of labeled samples. These approaches leverage VFL as the bridge to transfer knowledge from party A to party B. We formulate a general TL-based VFL objective as follows:

$$\min_{\phi_B,\theta_B} \ell_B(\phi_B;\theta_B;\mathcal{D}_B) + \lambda_1 \ell_A(\psi_A,\theta_A,\theta_B;\mathcal{D},\mathcal{D}_A^{ul}) + \lambda_2 \ell_{\text{TL}}(\theta_A,\theta_B;\mathcal{D}^{au},\mathcal{D}_A^{uu},\mathcal{D}_B^{uu}) \quad (8)$$

where ℓ_{TL} is the transfer learning loss that aims to reduce the domain discrepancy between source and target domains, and ℓ_A is the source party A's task loss that trains models using samples with labels of the source domain. ℓ_{TL} and ℓ_A together transfer the knowledge from the source domain to the target domain. The target party B utilizes its task loss ℓ_B to further adapt the transferred knowledge to its local task using samples $\mathcal{D}_B$ with labels of the target domain if $\mathcal{D}_B$ is available. ϕ_B is the target party B's local predictor. The target party B may or may not need the help of party A for inference, depending on the specific application of (8).

SFTL [24], a pioneering work exploring transfer learning in VFL, first trains feature extractors θ_A and θ_B to map two heterogeneous feature spaces into a common latent subspace through aligned samples $\mathcal{D}^{au}$. In this latent subspace, the passive party B's local models ϕ_B and θ_B are trained using data $\mathcal{D}_B$. Sharma et al. [82] leveraged a more efficient secure computation framework named SPDZ [86] to further enhance the efficiency of SFTL.

To support multi-party knowledge transfer, MMVFL [80] leverages consistency regularization to transfer label information from the active party to all passive parties so that each passive party can learn a local predictor with its pseudo-labeled samples. SFHTL [81] utilizes an autoencoder to learn local representations from each party and then aggregates local representations to form global representations, through which labels of the active party are propagated to each passive party. With labeled local samples, each party can train its local predictor independently. PrADA [83] addresses the label deficiency of VFL by transferring knowledge from a label-rich source domain to the target domain and leveraging the adversarial domain adaptation to minimize the discrepancy between the source and target domains.

V. Preserving Data Privacy and Defending Against Attacks

In a VFL system, privacy threats may emerge from the inside or the outside of the system, or both. If the attacker attempts to learn information about the private data of other parties without deviating from the VFL protocol, it is regarded as *honest-but-curious*. The attacker is regarded as *malicious* if it fails to adhere to the VFL protocol. In this section, we first review privacy-preserving protocols involved in the typical VFL framework (Section V-A and Section V-B), followed by discussions on emerging research on attacks and defense strategies (Sections V-C and V-D).

A. Private Entity Alignment

Private Set Intersection: (PSI) is the most common method for privacy-preserving entity alignment in VFL. In a PSI protocol, all parties cooperatively find the common ID intersection without revealing any information else. PSI protocols can be realized using various techniques, such as encryption and signature strategies [87] and oblivious transfer [88] etc. The standard PSI protocol is typically applied to a two-party VFL system. [89], [90] proposed methods for Entity Matching and PSI protocols that can be applied to multiple parties. PSI still reveals the common ID information. Several attempts have been made to enhance the privacy of the intersection ID set. [91] proposed an adapted PSI protocol for asymmetrical ID alignment using a Pohlig-Hellman encryption scheme and an obfuscate set to help protect the entity information of a weaker party with far fewer samples than the other party from being exposed. FLORIST [92] safeguards the entity membership information for all parties by using a union ID set and generating synthetic data for missing IDs in the union set. However, this method is limited to unbalanced binary classification tasks and incurs additional computational costs for generating and training the synthetic data.

B. Privacy-Preserving Training Protocols

VFL approaches proposed in the literature adopt various security definitions and privacy-preserving protocols. In this section, we summarize these protocols based on what is protected and exposed during VFL training and inference. We first provide the basic protocol of VFL. We then discuss other protocols which adopt either relaxed or enhanced privacy constraints. Fig. 6 illustrates these protocols.

Basic Protocol (P-1): Keeping private data and models local: All VFL participants keep their private data (e.g., labels and features), as well as the global module $\mathcal{F}_K$ and models $\{\mathcal{G}_k\}_{i=1}^{K}$ local during training and inference. Intermediate results are transmitted in plaintext for training and inference. We use this setting as our basic protocol (termed *P-1*). A case in point, during the training process of VFL (see Algorithm 1), each party k's intermediate results H_k and gradients $\frac{\partial \ell}{\partial H_k}$ instead of raw data are transmitted, preventing private data from being revealed. Liu et al. [16] provided security proof proving that private features $\mathbf{x}_k$ can not be exactly recovered in the P-1 protocol when no prior knowledge about data is available.

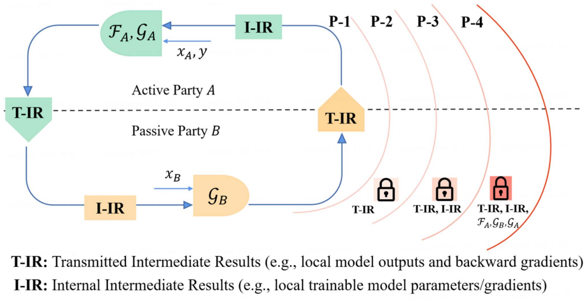

$\mathcal{F}_A, \mathcal{G}_A$
I-IR
x_A, y
P-1
P-2
P-3
P-4
T-IR
Active Party A
Passive Party B
T-IR
x_B
$\mathcal{G}_B$
I-IR
T-IR
T-IR, I-IR
T-IR, I-IR, $\mathcal{F}_A, \mathcal{G}_B, \mathcal{G}_A$
T-IR: Transmitted Intermediate Results (e.g., local model outputs and backward gradients)
I-IR: Internal Intermediate Results (e.g., local trainable model parameters/gradients)

Fig. 6. Conceptual view on the information flowing within and between an active party A and a passive party B during training to illustrate security protocols P-1, P-2, P-3 and P-4.

Relaxed Protocol (P-0): Nonprivate label or model: In literature and applications, there are also cases where this security assumption of *P-1* is relaxed, resulting in a few variants of protocols, including:

- Non-private Labels. These are cases where labels can be accessed by all parties for training, and the security model is to protect features only [25], [42], [60].
- Non-private global module or local models. These are cases where the global module [93] or local models [29], [94], [95], [96] are considered *white-boxed* to adversaries.

Since these variants relax the basic privacy requirement of VFL, we assign a lower level to them (*P-0*), and we use *P-0(y)* and *P-0(g)* to denote the non-private label and non-private model scenarios, respectively. Building on the basic protocol *P-1*, privacy-preserving techniques have been adopted to further protect the training procedure, resulting in protocols with enhanced privacy. Below we describe the most representative protocols based on what is exposed, in ascending order of privacy level.

Standard Protocol (P-2): Protecting transmitted intermediate results: In this protocol, P-1 is satisfied. In addition, the intermediate results transmitted between parties are protected by cryptography protocols, while other training information processed within each party is left in plaintext to balance privacy and efficiency. For example, HE [3], [97] can be adopted to encrypt sample-level outputs H_k and gradients $\frac{\partial \ell}{\partial H_k}$ transmitted between each passive party k and the active party to thwart privacy attacks. Batch-level gradients $\nabla_{\theta_k} \ell$ computed within party k are in plaintext for efficient training. The SecureBoost [17] is another example where HE is used to protect transmitted intermediate results, but the aggregated gradients are exposed to the active party.

Enhanced Protocol (P-3): Protecting entire training protocol: In this protocol, P-2 is satisfied. In addition, no training information is revealed to any party except for the resulting trained models. For example, batch-level information such as local model gradients $\nabla_{\theta_k} \ell$ and parameters θ_k can be protected by adopting Secure Multi-Party Computation (MPC) [21]. Most existing works focus on the *honest-but-curious* assumption, which assumes that the adversary follows the VFL protocol. To further handle malicious settings, more advanced

TABLE IV
SUMMARY OF EXISTING DATA INFERENCE ATTACKS IN VFL. A.P. REPRESENTS THE ATTACKING PHASE. IN THE A.P. COLUMN, TRG DENOTES THE TRAINING PHASE AND INF DENOTES THE INFERENCE PHASE

	Attacking Method	VFL Setting	Model	Against Protocol	A.P.	Auxiliary Requirement
Label Inference Attack	Direct Label Inference (DLI) [28], [33]	aggVFL	NN	P-1	TRG	–
	Norm Scoring (NS) [104]	splitVFL$_c$	NN	P-1	TRG	–
	Direction Scoring (DS) [104]	splitVFL$_c$	NN	P-1	TRG	–
	Residual Reconstruction (RR) [105]	aggVFL	LR	P-2	TRG	–
	Gradient Inversion (GI) [33]	aggVFL	NN	P-2	TRG	–
	Gradient Inversion with a Label Prior [106]	splitVFL$_c$	NN	P-2	TRG	Label prior distribution
	Passive Model Completion (PMC) [28]	splitVFL	NN	P-3	INF	Labeled data
	Active Model Completion (AMC) [28]	splitVFL	NN	P-3	INF	Labeled data
	Spectral Attack (SA) [107]	splitVFL$_c$	NN	P-3	INF	-
	Label-related Relation Inference (LRI) [96]	splitVFL$_c$	GNN	P-0(g)	INF	-
Feature Inference Attack	Binary Feature Inference (BFI) [108]	splitVFL	NN	P-1	TRG	Binary features
	Reverse Multiplication Attack (RMA) [109]	aggVFL	LR	P-2	TRG	Corrupted coordinator
	Protocol-aware Active Attack (PAA) [110]	aggVFL	LR	P-2	TRG	Victim has 1 feature
	Reverse Sum Attack (RSA) [109]	aggVFL	GBDT	P-2	TRG	–
	Equality Solving Attack (ESA) [97]	aggVFL	LR	P-0(g)	INF	–
	Path Restriction Attack (PRA) [97]	aggVFL	Tree	P-0(g)	INF	–
	Generative Regression Network (GRN) [97]	aggVFL	NN	P-0(g)	INF	–
	White-box Model Inversion (WMI) [98], [99]	aggVFL & splitVFL$_c$	LR & NN	P-0(g)	INF	–
	Black-box Model Inversion (BMI) [98], [99]	aggVFL & splitVFL$_c$	LR & NN	P-1	INF	Labeled data
	Catastrophic Data Leakage in VFL (CAFE) [31]	aggVFL$_c$	NN	P-0(g)	TRG	–
	Infer Attribute from Representation (IAR) [111]	aggVFL$_c$ & splitVFL$_c$	NN	P-0(g)	INF	Attribute data

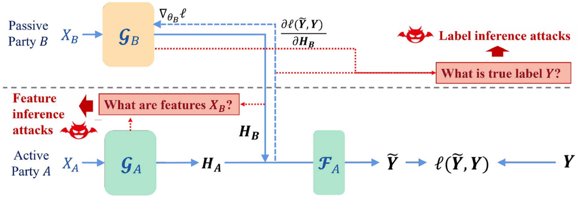


Fig. 7. Illustration of data inference attacks in VFL system. The active party A typically infers features or attributes of the passive party B, while the passive party B typically infers labels of the active party A.

privacy-preserving techniques such as SPDZ have also been integrated with VFL [82].

Strict Protocol (P-4): Protecting training protocol and learned models: This protocol further enhances the P-3 to protect final learned models using privacy-preserving techniques such as secret sharing [98] and hybrid schemes that combine HE and SS [99], [100]. It only reveals the final inference results but nothing else. This protocol addresses the emerging privacy challenge that the local model is exploited by its owner to infer private information about other parties [17], [26], [98]. However, it requires complex computations, which limits its efficiency and scalability.

C. Defending Against Data Inference Attacks

Both features and labels are typically considered private in a VFL system. Therefore, feature and label protections are critical research subjects for VFL. Fig. 7 illustrates data inference attacks in VFL.

1) Label Inference Attacks: In real-world scenarios, labels such as patients' diagnostic results and individual loan default records are considered sensitive information that only authorized institutions can access. A passive party B (i.e., the attacker) may try to infer the valuable label owned by the active party A using the information they accumulate during training or inference. It may follow the protocol *passively* under the *honest-but-curious* security assumptions or *actively* by tampering with the protocol under the *malicious* assumptions. The literature has proposed various label inference attacks under various security protocols, as summarized in Table IV.

Label inference attacks using sample-level gradients: When the VFL applies P-1 protocol, a passive party B (i.e., the attacker) has access to sample-level gradients $\frac{\partial\ell}{\partial H_B}$ sent backward from the active party A. The attacker can exploit this information to conduct Direct Label Inference (DLI) [26], [101]. DLI can achieve accuracy up to 100% if the active party adopts a *non-trainable* global module $\mathcal{F}_A$ such as a softmax function because the gradient vector for each sample has only one element that has an opposite sign against all the others, thereby disclosing the labels [101]. For special scenarios like binary classification, the attacker can deduce labels from sample-level gradients by mounting Norm Scoring (NS) or Direction Scoring (DS) attack [101] even when the global module $\mathcal{F}_A$ is a trainable model (e.g., neural network).

Label inference attacks using batch-level gradients: When the VFL applies the P-2 protocol, no intermediate result exchanged among parties is revealed to any party (e.g., encrypted by HE [97]). Thus, the passive party B (i.e., the attacker) cannot obtain sample-level gradients $\frac{\partial\ell}{\partial H_B}$, but it may have access to batch-level (i.e., local model) gradients $\nabla_{\theta_B}\ell$. Studies have shown that it is still possible to infer the true labels with high accuracy based on $\nabla_{\theta_B}\ell$ through gradient inversion attack (GI) [31], [103] and residue reconstruction attack (RR) [102]. The former predicts gradients $\nabla_{\theta_B}\hat{\ell}$ by minimizing the distance between $\nabla_{\theta_B}\hat{\ell}$ and the ground truth $\nabla_{\theta_B}\ell$ and latter solves a gradient matching problem to infer the plaintext values $\nabla_{\theta_B}\hat{\ell}$ from $[[\frac{\partial\ell}{\partial H_B}]]$.

Label inference attacks using trained models: When the VFL applies the P-3 protocol, no training information is revealed to any party but only the final trained local model. The P-3 protocol can be achieved through MPC-based VFL approaches [21], [100]. A possible label inference strategy is for a passive party to finetune its trained local model with an inference head using auxiliary labeled data, and then predict labels using the complete model (i.e., the finetuned local model with the inference head). This attack is called Passive Model Completion (PMC) [26], in which the passive party is semi-honest. An *active* version of model completion (AMC) is also proposed in [26], which leverages a malicious local optimizer instead of normal ones. MC relies heavily on the adequateness of the auxiliary data owned by the passive party as an attacker. Sun et al. [104] proposed a Spectral Attack (SA) that enables a passive party to predict labels by clustering outputs of the trained local model, thereby eliminating the dependency on auxiliary data. Qiu et al. [93] proposed a Label-related Relation Inference (LRI) attack targeting label-related relations in the graph owned by the active party, assuming the attacker has access to the global module and can obtain prediction results.

2) Feature Inference Attacks: An individual's original feature is at the heart of privacy protection because it contains sensitive information that is not allowed to be shared. Various attacking methods have been proposed to infer features from shallow models (e.g., logistic regression and decision trees) [94], [105], [106] and complex models (e.g., neural networks and random forests) [29], [94], [95], [96]. We summarize existing feature inference attacks in Table IV. These attacks are typically under the setting where the active party (with labels) A is the attacker who attempts to recover features of a passive party B. The attacker may or may not have the knowledge of the passive party's model parameters θ_B, which are, respectively, referred to as the *white-box* and *black-box* settings.

Feature inference attacks under white-box setting: Under the white-box setting, the attacker (i.e., the active party or the server) has access to its own model $\mathcal{G}_A$, the passive party's local model $\mathcal{G}_B$, the aligned data indices and possibly labels. In literature, there are mainly two ways to conduct white-box feature inference attacks: model inversion [95], [96] during the inference phase and gradient inversion during the training phase [29]. The core idea of white-box model inversion (WMI) is to optimize a variable $\hat{x}_B$ to approximate the passive party's real input data x_B such that the predicted output of the VFL federated model is close enough to the real output computed based on x_B. The VFL federated model can be a linear model, tree model, or neural network model, with model parameters fixed during the optimization.

He et al. [95] and Jiang et al. [96] proposed similar white-box model inversion attacks under the splitNN and aggVFL settings, respectively. Luo et al. [94] proposed three white-box feature inference attacks to learn $\hat{x}_B$ for three different models: an Equality Solving Attack (ESA) for attacking the logistic regression, a Path Restriction Attack (PRA) for the decision tree, and a Generative Regression Network (GRN) for the neural network and random forest. These attacks can be seen as specialized WMI attacks.

CAFE [29] extends GI to a white-box VFL setting, where the attacker has access to the passive party's model parameters, gradients, and the aligned data indices. With this knowledge, CAFE achieved state-of-the-art data recovery quality even with large batch sizes.

Feature inference attacks under black-box setting: Attackers under the black-box setting typically have some prior knowledge about the model or data of the passive party in order to conduct feature inference successfully.

Peng et al. [105] proposed a Binary Feature Inference attack (BFI) to reconstruct binary features from the passive party's local model output H_B (P-1 protocol), assuming the local model only has one fully connected layer. In addition, BFIA adopts the Leverage Score Sampling technique [114] to boost the attack efficiency. Weng et al. [106] and Hu et al. [107] proposed a Reverse Multiplication Attack (RMA) that solves linear equations with x_B being the only unknown variable and a Protocol-aware Active Attack (PAA) that first obtains passive party B's outputs and then deduce x_B by solving linear systems, respectively, to infer the private features x_B in the vertical logistic regression setting, which applies the P-2 protocol. Weng et al. [106] also proposed a Reserve Sum Attack (RSA) targeting SecureBoost by inferring the partial order of the passive party's input features. He et al. [95] proposed a black-box MI attack to learn features x_B of the passive party B under the splitNN setting by training a shadow model $\hat{\mathcal{G}}_B$ to mimic the local model $\mathcal{G}_B$ using auxiliary data. Jiang et al. [96] proposed a similar black-box MI method under the aggVFL setting.

Attribute Inference Attacks: Aside from original features, privacy-sensitive attributes not represented in training data may also be inferred through overlearned model [108].

In the rest of this subsection, we discuss defense strategies that alleviate the threat posed by these data leakage attacks.

3) Cryptographic Defense Strategies: Cryptographic Defense Strategies (CDS) use secure computations to evaluate functions on multiple parties in a way that only the necessary information is exposed to intended participants while preventing private data from being inferred by possible adversaries. Today, large-scale deployment of CDS to machine learning models, especially deep learning models, is still challenging. The focus of existing works in this direction is to improve the privacy-efficiency trade-off through the in-depth designing of privacy-preserving protocols. We adopt protocols defined in Section V-B as a vehicle to compare representative CDS, as listed in Table V. We consider a defense follow a particular protocol only when it satisfies all requirements of that protocol. A line of research works [3], [19], [21], [97], [100], [109], [110] focuses on designing CDS to protect the data privacy of vertical linear and logistic regressions. Gascon et al. [19] proposed a hybrid MPC protocol that combines Yao's garbled circuits with tailored protocols for securely solving vertical linear regression (GasconLR). Hardy et al. [97] proposed a HE-based scheme for training the vertical logistic regression (HardyLR). Follow-up works BaiduLR [109] and SecureLR [110] remove the coordinator from the training and inference procedure by relaxing either the efficiency or privacy constraint. HardyLR, BaiduLR, and SecureLR are vulnerable to privacy attacks targeting batch-level gradients (Section V-C1). To address this limitation, Chen et al. [21] proposed a hybrid defense, named CAESAR, that combines HE and MPC to encrypt all intermediate results during the training and inference phases except the resulting trained models. The HeteroLR module of FATE [100] extends CAESAR further to

TABLE V

SUMMARY OF CRYPTOGRAPHIC DEFENSE STRATEGIES. IN THE DEFENSE COLUMN, GC, SS, HE, FE, AND TEE DENOTE GARBLED CIRCUITS, SECRET SHARING, HOMOMORPHIC ENCRYPTION, FUNCTIONAL ENCRYPTION, AND TRUSTED EXECUTION ENVIRONMENT, RESPECTIVELY. IN THE ADVERSARIAL ASSUMPTION COLUMN, SH DENOTES SEMI-HONEST AND MA DENOTES MALICIOUS. IN THE PROTOCOL COLUMN, WE ASSIGN EACH DEFENSE WITH PROTOCOLS (SEE SECTION V-B) IT SATISFIES; "a" AND "p" DENOTE ACTIVE AND PASSIVE PARTIES, RESPECTIVELY

Defense Work	VFL Setting	Model	Defense	Protocol	Party	Require Coordinator	Adversarial Assumption
GasconLR [21]	aggVFL	LR	GC+SS	P-3	≥ 2	✓	SH
HardyLR [100]	aggVFL	LR	HE	P-2	≥ 2	✓	SH
BaiduLR [112]	aggVFL	LR	HE	P-2	≥ 2	✗	SH
SecureLR [113]	aggVFL	LR	HE+SS	P-2	≥ 2	✗	SH
CAESAR [23]	aggVFL	LR	HE+SS	P-3	$= 2$	✗	SH
HeteroLR [103]	aggVFL	LR	HE+SS	a :P-3, p :P-4	$= 2$	✗	SH
FedV [25]	aggVFL	LR & SVM	FE	P-2	≥ 2	✓	SH
SecureBoost [19]	aggVFL	XGB	HE	P-2	≥ 2	✗	SH
SecureBoost+ [37]	aggVFL	XGB	HE	P-2	≥ 2	✗	SH
SecureXGB [39]	aggVFL	XGB	HE+SS	P-3	$= 2$	✗	SH
MP-FedXGB [42]	aggVFL	XGB	SS	P-3	≥ 2	✓	SH
SecureGBM [38]	aggVFL	LGBM	HE	P-2	$= 2$	✗	SH
Pivot [101]	aggVFL	RF & GBDT	HE+SS	P-3	≥ 2	✗	SH, $\leq K$-1 colluded parties
Enhanced Pivot [101]	aggVFL	DT	HE+SS	P-4	≥ 2	✗	SH, $\leq K$-1 colluded parties
FedSGC [114]	aggVFL$_c$	GNN	HE	P-2	$= 2$	✗	SH
ACML [115]	splitVFL$_c$	NN	HE	P-1	$= 2$	✗	SH
PrADA [86]	splitVFL	NN	HE	P-1	≥ 2	✗	SH
BlindFL [102]	splitVFL	NN	HE+SS	a :P-2, p :P-4	$= 2$	✗	SH
SFTL [84]	aggVFL	NN	HE or SS	P-2 or P-3	$= 2$	✗	SH
SEFTL [85]	aggVFL	NN	HE+SPDZ	P-3	$= 2$	✗	MA,dishonest majority
N-TEE [116]	aggVFL	XGB	TEE	P-3	≥ 2	✗	SH

encrypt the passive party's local model after training.Designing CDS for vertical neural networks (VNN) is more challenging for both computation and communication. Therefore, current CDS for VNN either target shallow neural networks [24], [82], [99] or are tailored to protect specific intermediate results exposed to the adversary [83], [112] for balancing privacy and efficiency. SFTL [24] designs a HE-based protocol and an SS-based protocol, respectively, to encrypt information shared between two parties that adopt neural networks with one or two layers. The follow-up work [82] leverages SPDZ [86] to enhance the efficiency of SFTL further. BlindFL [99] builds privacy-preserving VNN models through a federated source layer (FSL), which leverages a hybrid scheme mixing HE and MPC to guarantee the privacy of original data. ACML [112] builds privacy-preserving SplitVFL and introduces a HE-equipped interactive layer between the active party and the passive party to protect the passive party's local model output. PrADA [83] extends the interactive layer of ACML to the splitVFL setting, in which the global module is a linear model and local models are neural networks. FedSGC [111] utilizes HE to protect transmitted graph structural information.

For tree-based VFL, SecureBoost [17], SecureBoost+ [35], SecureXGB [37], and MP-FedXGB [40] integrate XGBoost into VFL. SecureBoost and SecureBoost+ exploit additive homomorphic encryption (HE) to encrypt the information transmitted between parties to protect private data. SecureXGB protects all intermediate results through a hybrid scheme combining additive HE and secret sharing (SS), thereby enhancing the privacy level. MP-FedXGB proposed an SS scheme with distributed optimization to support more-than-two-party scenarios. SecureGBM [36] is a LightGBM-based VFL using additive HE to protect transmitted information. Pivot [98] utilizes SS mixed with additive HE to guarantee that no intermediate information is disclosed. It additionally proposed an enhanced protocol to conceal the values of leaf labels and split thresholds from all participating parties, as well as protocols to handle malicious parties. Targeting SecureBoost, N-TEE [113] introduces a feature inference attack leveraging the approximate distribution of feature values and proposed two countermeasures based on Trusted Execution Environment (TEE) to mitigate feature leakage.

CDS are typically applied to utility-critical applications, such as finance and healthcare, to achieve lossless model utility (i.e., performance) while maintaining an acceptable balance between privacy and efficiency. For applications in which efficiency is a major concern or CDS is not feasible, non-cryptographic defense strategies are preferred.

4) Non-Cryptographic Defense Strategies: Non-cryptographic Defense Strategies preserve privacy essentially by reducing the dependence between private data and the information exposed to the attacker. There are several representative ways to reduce such dependency, including adding noise, gradient discretization [119], gradient sparsification [120], [121] and their hybrid [122]. These methods typically exhibit a trade-off between utility and privacy.

Adding noise [26], [101], [123], [124] is a basic defense method for reducing leakage in FL. Noise following Laplace distribution or Gaussian distribution is commonly used. In VFL settings, it typically adds noise to the gradients or intermediate results shared with other parties to defend against label or feature leakages [26], [74]. [125] introduces a hybrid differentially private VFL method that adds Gaussian noise to all parties' intermediate results to achieve both local and joint differential privacy. [38], [39] applies differentially private noise to federated gradient-based decision trees in customized ways to achieve a

TABLE VI
SUMMARY OF EMERGING SPECIALIZED DEFENSE STRATEGIES FOR DEFENDING AGAINST DATA INFERENCE ATTACKS (SEE TABLE IV)

	Defense Work	VFL Setting	Model	Defense Scheme	Against Attack	Defending Party
Defenses against Label Inference Attack	MARVELL [104]	splitVFL$_c$	NN	Add Noise	NS, DS	Active party
	Max-Norm [104]	splitVFL$_c$	NN	Add Noise	NS, DS	Active party
	CAE [33]	aggVFL	NN	HE + Disguise Label	DLI, MC	Active party
	DCAE [33]	aggVFL	NN	HE + Disguise Label + DG	DLI, MC	Active party
	PELoss [118]	splitVFL$_c$	NN	Potential Energy Loss	MC	Active party
	dCorr [107]	splitVFL$_c$	NN	Minimize Correlation	SA	Active party
	RM [119]	aggVFL	LR	HE + Random Mask	RR	Active party
	FedPass [120]	splitVFL	NN	Passport	MC	Active party
Defenses against Feature Inference Attack	FG [31]	splitVFL	NN	Random Fake Gradients	CAFE	Passive party
	DRAVL [121]	splitVFL$_c$	NN	Adversarial Training	MI	Passive party
	MD [108]	splitVFL	NN	Masquerade	BFIA	Passive party
	DP-Paillier-MGD [110]	aggVFL	LR	HE + DP	PAA	Passive party
	FedPass [120]	splitVFL	NN	Passport	CAFE, MI	Passive party

good privacy-utility trade-off. Chen et al. [126] integrated GNN into the splitVFL setting and leverage DP-enhanced additive secret sharing to protect data privacy. Gradient Discretization (GD) [119] encodes originally continuous gradients into discrete ones, aiming to reduce the private information disclosed to the attacker. [26], [31] leverage a specialized version of GD, named DiscreteSGD, to defend against label inference attacks in VFL. Gradient Sparsification (GS) [120] removes a portion of the original gradients with small absolute values by setting them to 0 while preserving the convergence of the original VFL task. Similar to GD, GS leverages information reduction to mitigate privacy leakage and is effective in defending against various label inference attacks in VFL [26], [31].

A feasible direction to achieve better trade-offs between privacy and utility is designing specialized defense strategies tailored to specific data inference attacks.

5) Emerging Specialized Defense Strategies: Emerging specialized defense strategies are designed to thwart attacks that are difficult to defend against by traditional defense strategies. We compare representative emerging defense strategies in Table VI.

Defenses against label inference attacks: The MARVELL [101] is tailored to thwart binary label inference attacks by adding optimized noise to sample-level gradients. A heuristic Max-Norm defense [101] is also proposed for the same attack. Two label disguising methods, called Confusional AutoEncoder (CAE) and DiscreteSGD-enhanced CAE (DCAE) [31], are proposed to directly protect label information by encoding the original real label to soft fake labels with maximum confusion. PEloss [115] and dCorr [104] are two auxiliary losses that are proposed to defend against the Model Completion (MC) attack and Spectral Attack (SA), respectively. Both methods try to train the attacker's local model for a large generalization error. Random Masking (RM) [116] aims to defend against the Residue Reconstruction attack (RR) by injecting zeros into randomly selected positions of the HE-encrypted sample-level gradients. FedPass [117] leverages passport techniques to thwart both label and feature inference attacks. *Defenses against feature inference attacks.* Fake Gradients (FG) [29] is proposed to defend against Catastrophic Data Leakage in VFL (CAFE) by replacing the true gradients with randomly generated ones while keeping their corresponding positions. DRAVL [118] is to defend against Model Inversion (MI) through adversarial training. In [105], a Masquerade Defense (MD) is proposed to thwart the Binary Feature Inference attack (BFI) by misleading the attacker to focus on randomly generated binary features. DP-Paillier-MGD [107] is to thwart the Protocol-aware Active Attack (PAA) by masking encrypted sensitive information to prevent the attacker from learning the precise value of the passive party's output and, thereby, the private features. Adversarial training [108], [127] and mutual information regularization [108] are proposed to safeguard sensitive attributes of training samples.

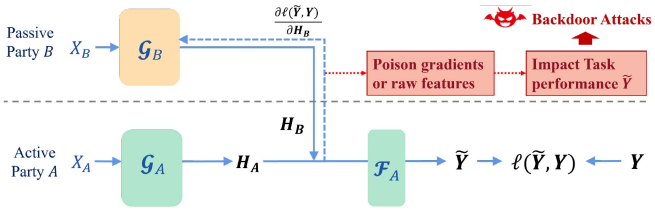


Fig. 8. Illustration of backdoor attacks in VFL. Passive parties are the backdoor attackers who aim to impact the task performance of the active party.

D. Defending Against Backdoor Attacks

Different from data inference attacks, whose target is to invade privacy and steal data, the target of malicious backdoor attacks is to mislead the VFL model or harm its overall performance on the original task. Typically, passive parties are the backdoor attackers, while the active party is the victim since only the active party has labels. In this section, we summarize existing backdoor attacks and defenses.

1) Backdoor Attacks: Existing research on backdoor attacks can be divided into two main categories, *targeted* and *non-targeted*, depending on whether the attacker has a determinant backdoor target or not. Fig. 8 illustrates backdoor attacks, and Table VII summarizes the settings and methods for backdoor attacks.

Targeted backdoor attacks: The attacker secretly trains a model that achieves high performance on both the original and the targeted backdoor tasks. The objective function of targeted

TABLE VII
SUMMARY OF EXISTING BACKDOOR ATTACKS IN LITERATURE. A.P. REPRESENTS THE ATTACKING PHASE

	Attacking Method	VFL Setting	Against Protocol	A.P.	Auxiliary Requirement
Targeted Backdor Attack	Label Replacement Backdoor (LRB) [131]	aggVFL	P-2	TRG	$\geq$ 1 label of clean samples
	Adversarial Dominant Input (ADI) [132]	VLR/splitVFL$_c$	P-0(g)/P-1	INF	a few samples of other party
Non-targeted Backdoor Attack	Adversarial attack [32], [133]	splitVFL/aggVFL	P-1	TRG	–
	Missing attack [32]	splitVFL/aggVFL	P-3	TRG	–
	Graph-Fraudster [134]	splitVFL	P-2	INF	–

backdoor attacks can be written as follows:

$$\min_{\Theta} \mathcal{L}_{\text{BD}}(\Theta;\mathcal{D}) \triangleq \frac{1}{N_{cln}} \sum_{i\in\mathcal{D}_{cln}} \ell(\tilde{y}_i, y_i) + \frac{1}{N_{poi}} \sum_{i\in\mathcal{D}_{poi}} \ell(\tilde{y}_i, \tau)$$

where $\tilde{y}_i$ is the prediction for sample x_i, subscripts $_{cln}$ and $_{poi}$ are short for "clean" and "poisoned" respectively, τ denotes the target label chosen by the attacker.

Liu et al. [128] proposed a Label Replacement Backdoor attack (LRB) in which the attacker replaces the gradients of a triggered sample with the ones of a clean sample of the targeted class. Pang et al. [129] introduced the Adversarial Dominating Input (ADI), which is an input sample with features that override all other features and lead to certain model output.

Non-targeted backdoor attacks: Non-targeted backdoor attacks aim to hurt the convergence or the performance of the original task by using adversarial samples [30], [130], noisy samples or missing features [30]. An adversarial sample is generated using the Fast Gradient Sign Method, in which a perturbation $\Delta x_i = \epsilon \text{sign}(\frac{\partial \ell}{\partial x_i})$ is added to the original sample x_i where ϵ is the magnitude of the perturbation [130]. Multiple research works [30], [130] demonstrate the effectiveness of this kind of attack in its misleading performance. If Δx_i is simply a randomly generated perturbation, the attack is referred to as the noisy-sample attack.

The missing-feature attack simulates real-world VFL scenarios with unstable network [30] in which, for example, the local model output of a passive party may fail to reach the active party for collaboration.

2) Defense Strategies: Traditional defense strategies such as adding noise and GS are effective in defending against targeted and non-targeted backdoor attacks [30], [31]. However, these defenses suffer from trade-offs between main task accuracy and backdoor task accuracy. On the other hand, cryptographic defense strategies are generally ineffective for defending against backdoor attacks because they preserve the computed outputs and thus do not impact the backdoor training objectives. In [128], the authors show that gradient-replacement backdoor attacks can still survive in HE-protected VFL protocols.

Therefore, emerging defense strategies have been proposed to further improve the effectiveness of defenses. For example, CAE and DCAE both show promising effectiveness in defending against targeted backdoor attacks [31]. RVFR [30] is put forward to defend against both target and non-target backdoor attacks in VFL scenarios by robust feature subspace recovery. We compare these defenses in Table VIII.

TABLE VIII
SUMMARY OF DEFENSE STRATEGIES FOR DEFENDING AGAINST BACKDOOR ATTACKS

Defense Work	VFL Setting	Defense Scheme	Against Attack
DP [33]	aggVFL	Add Noise	Targeted
GS [33]	aggVFL	Sparsify Gradient	Targeted
CAE [33]	aggVFL	HE + Disguise Label	Targeted
DCAE [33]	aggVFL	HE + Disguise Label + DG	Targeted
RVFR [32]	splitVFL	Robust Feature Sub-space Recovery	Targeted & Non-targeted

In sum, research works on defending against backdoor attacks in VFL are still at an early stage. It is worth exploring effective defense strategies while maintaining good model utility.

VI. DATA VALUATION AND FAIRNESS

VFL opens up new opportunities for cross-institution and cross-industry collaborations. As industrial use cases grow, a critical challenge for establishing a stable and sustainable federation among parties is the lack of fair data valuation and incentive design to allocate profits. In addition, a responsible VFL framework should also address various bias problems towards certain groups of people. In this section, we discuss the research progress for data valuation, explainability, and fairness for VFL.

A. Data Valuation

Currently, most research works on data valuations for FL framework still focus on HFL scenarios [132], [133], while data valuations on VFL are much less studied. [134], [135] are among the earliest works that proposed contribution evaluation frameworks for VFL using Shapley valuations on features. Shapley-based approaches typically adopt model performance gain as a key metric to measure data value. [136] proposed a model-free approach that uses conditional mutual information for Shapley to evaluate the feature importance and data values in VFL. [137] proposed an embedding-based Shapley evaluation method for VFL and applied this method to both asynchronous and synchronous settings. [138] focused on party-level evaluation from a mutual information (MI) perspective and adopted such evaluations to select important participants to improve the scalability of VFL. However, Shapley-based and MI-based evaluations are computationally challenging, which makes them difficult to apply to real-world cases. Improving the efficiency of Shapley calculations is an important research direction.

TABLE IX
COMMONLY USED DATASETS IN VFL

Dataset	Data Type
Tabular	Income [146], Bank [147], Credit Card [148]
	Give Me Some Credit [149], MIMIC III [150]
	Breast Cancer [151], Diabetes [152], Avazu [153]
	Criteo [154], Vehicle [155], Drive [156], Cover type [157]
	NUSWIDE [158], Handwritten [159], Epsilon [160]
Image	BHI [161], CheXpert [162], Modelnet [163]
Graph	Cora [164], Citeseer [165], PubMed [166]
Text	Yahoo Answers [167], News20 [168]

B. Explainability

In fields that are highly regulated, such as financial and medical fields, making the trained VFL model explainable to authorities and compliance is of paramount importance. Currently, only a limited amount of works are proposed to address the explainability of VFL. For example, [139] proposed an explainable VFL framework using credibility assessment and counterfactual analysis to control data quality and explain counterfactual instances. [140] designed a VFL scheme based on logistic regression with bounded constraints for interpretable scorecards in credit scoring. [83] proposed a feature grouping method that converts original features with low explainability into explainable feature groups to enhance the explainability of VFL prediction models. While designing VFL with explainability is an important research topic, how to reconcile privacy-preserving and explainability in VFL is also a crucial research direction because the two objectives may contradict each other.

C. Fairness

Machine learning models trained in a collaborative setting may inherit bias towards certain user groups. Addressing the fairness problem in VFL is an emerging research topic. FairVFL [141] is a framework to use adversarial learning to remove bias for the fairness-sensitive features in a privacy-preserving VFL setting. [142] provided a fairness objective in VFL and developed an asynchronous gradient coordinate-descent ascent algorithm to solve it. The core challenge for addressing fairness in VFL is to identify fairness-sensitive features and perform collaborative debias training while preserving data privacy and protocol efficiency.

D. Datasets

We list datasets commonly used in existing VFL works in Table IX. Most of the datasets used in VFL research works are tabular datasets from the domains of Finance, Healthcare, and Advertising. NUSWIDE and Vehicle datasets consist of multi-modal features that can naturally simulate the two-party VFL scenario, whereas other datasets require manual partition. In summary, more practical datasets and high-quality benchmarks are still called for to facilitate industrial applications and academic research in VFL.

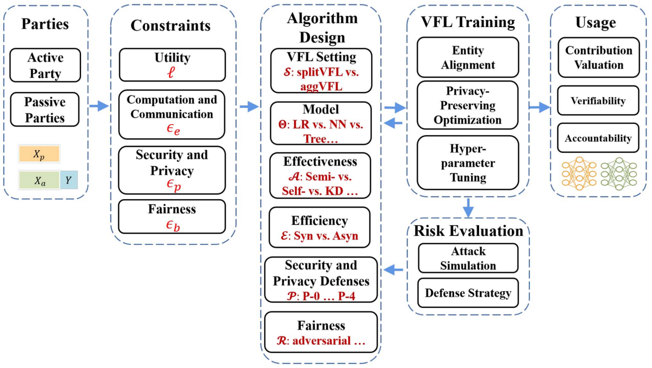


Fig. 9. VFLow: A Framework for setting up, designing and optimizing VFL algorithms.

VII. VFLow: A VFL Optimization Framework

We propose a comprehensive VFL optimization framework consisting of major considerations for setting up and optimizing a VFL algorithm, as illustrated in Fig. 9. We termed this framework *VFLow*.

In VFLow, we take into account major constraints, including utility, privacy, efficiency, and fairness, to guide the design of a VFL algorithm from aspects of the model architecture and partition settings, effectiveness and efficiency improving strategies, privacy defense strategies, as well as fairness improving strategies covered in this work. In addition, VFLow consists of a separate risk evaluation module that comprehensively evaluates data attacks and defense strategies. Finally, for model usage, party contributions, accountability, and verifiability tools are necessary for a sustainable and trustworthy federation (also see Section IX). We further extend the objective function formulated in (1) to a more general meta-objective, in which we want to minimize the main task loss (i.e., maximize utility) constrained by privacy, efficiency (i.e., communication and computation), and fairness:

$$\min_{\Theta} \ell(\Theta; \mathcal{S}, \mathcal{A}, \mathcal{E}, \mathcal{P}, \mathcal{R}, \mathcal{D})$$

$$s.t.\ \ M_p(\Theta; \mathcal{K}, \mathcal{P}) \leq \epsilon_p, M_e(\Theta; \mathcal{E}, \mathcal{P})) \leq \epsilon_e, M_b(\mathcal{R}, \mathcal{D}) \leq \epsilon_b$$

where Θ and $\mathcal{S}$ denote specific models and a VFL setting, respectively; $\mathcal{A}$ denotes an effectiveness improving strategy, $\mathcal{P}$ denotes a privacy defense strategy, $\mathcal{K}$ denotes the collection of attack algorithms, $\mathcal{E}$ denotes an efficiency improving strategy, and $\mathcal{R}$ denotes a fairness improving strategy. M_p denotes a measurement for measuring privacy leakage imposed by attacks $\mathcal{K}$ against the defense strategy $\mathcal{P}$. M_e is the efficiency measure, typically with respect to communication load and computation resources. M_b measures the system bias. ϵ_p, ϵ_e, and ϵ_b are constraints for privacy leakage, efficiency cost, and bias, respectively. This optimization problem can be considered as a constrained multi-objective federated learning problem [166]. Such formulation brings about a set of solutions, each of which is an optimal trade-off between multiple objectives and thus provides stakeholders with flexible decision options.

VIII. Applications

Recommendation systems: are typically adopted in VFL to support advertising applications. Federated bandit can be used as a promising technique [167], [168], [169] for FL. Shmueli et al. [170] proposed a privacy-preserving collaborative filtering protocol. Atarashi et al. [171] proposed a higher-order factorization machine in the VFL setting. Recommendation systems can be built between two platforms holding different rating data. Cui et al. [172] proposed a cross-platform recommendation based on secure computation protocols. Zhang et al. [173] proposed a VFL recommendation based on clustering and latent factor model to reduce the dimension of the matrix and improve the recommendation accuracy. To achieve privacy-preserving recommendations based on the personal data cloud, Yuan et al. [174] proposed a hybrid federated learning recommendation algorithm named HyFL. Many internet companies have adopted VFL to support advertising. ByteDance developed a tree-based VFL algorithm based on the Fedlearner framework, which significantly improves its advertising efficiency [175]. Based on the VFL module in its 9N-FL framework, JD established a joint model for advertising, which has promoted the cumulative increase of all participating parties' income [176]. Tencent applied its Angel PowerFL platform to establish a VFL federation between advertisers and advertising platforms to boost model accuracy [177]. Based on the trusted intelligent computing service framework (TICS), Huawei applied VFL to advertising [178] to leverage user profile and behavior data dispersed in different platforms.

Finance: is another major application that new VFL approaches have been rapidly developed. For example, a gradient-based method for traditional scorecard model training is proposed in [140]. In [21], a secure large-scale sparse logistic regression algorithm is designed and applied to financial risk control. Kang et al. [83] developed a fine-grained adversarial domain adaptation algorithm to address the label deficiency issue in the financial field. Long et al. [179] discussed the applications and open challenges for FL in open banking. Wang et al. [135] provided an overview of the use cases of FL in the insurance industry. WeBank uses customers' credit data and invoice information from partner companies to jointly build a risk control VFL model [4]. *Healthcare* has been very active in applied research in VFL. A privacy-preserving logistic regression is proposed in [107] and applied to clinical diagnosis. Chen et al. [49] proposed an asynchronous VFL framework and verified the effectiveness of this framework on the public health care dataset MIMIC-III. In [180], the authors applied VFL to cancer survival analysis to predict the likelihood of patients surviving time after diagnosis and to analyze which features might be associated with the chance of survival. [57] proposed an efficient VFL method using autoencoders to predict hearing impairment after surgery based on a vestibular schwannoma dataset. Song et al. [181] applied VFL to the joint modeling between mobile network operators (MNOs) and health care providers (HP).

Emerging applications: have also been exploited in recent years for discovering novel data utilization in fields such as electric vehicles and wireless communications. Teimoori et al. [182] proposed a VFL algorithm to locate charging stations for electric vehicles while protecting user privacy. [183] discussed the opportunities for VFL to be utilized in 5 G wireless networks. [184] proposed a VFL-based cooperative sensing scheme for cognitive radio networks. [185] developed a VFL framework for optical network disaggregation. [186] applied VFL to collaborative power consumption predictions in smart grid applications. [187] proposed VFL modelings for predicting failures in intelligent manufacturing.

MultiModal Tasks: are performed when participants in VFL hold data from multiple modalities, such as vision, language, and sense. Liu et al. [188] proposed an aimNet that helps the FL model learn better representations from textual and visual features through multi-task learning. Liang et al. [74] proposed a self-supervised vertical federated neural architecture search approach that automatically optimizes each party's local model for the best performance of the VFL model, given that participating parties hold heterogeneous image data. *Vertical federated graph learning (VFGL)* algorithms are proposed to leverage features, relations, and labels that belong to the same group of people but are dispersed among different organizations. VFGNN [126] and FedVGCN [189] perform node classification on the scenario where all parties share the same set of nodes, but each party only owns partial features and relations of these nodes. FedSGC [111] performs node classification on another scenario where one party has only graph structural information while other parties have only node features.

IX. Open Challenges and Future Direction

In this section, we discuss some of the major open challenges facing the development of VFL frameworks and propose possible paths in the future.

Interoperability: With the rapid development of VFL projects in real-world scenarios, the lack of interoperability of existing frameworks has become a new pain point for its industrial growth. Different platforms adopt different sets of secure computation and privacy-preserving training protocols, making cross-platform collaboration difficult and turning data silos into platform silos. One possible route to solve this challenge is to enforce the interoperability of platforms by developing algorithm and architecture standards so that platforms can connect with others more readily. Another route is to develop seed projects to support fundamental functionalities and modules for interoperability as a plug-in tool for diverse platforms.

Trustworthy VFL: To be trustworthy, VFL frameworks must appropriately reflect characteristics such as privacy and security, effectiveness, efficiency, fairness, explainability, robustness, and verifiability. Data needs to be protected in transit and at rest with clear security and privacy definitions and scopes. Despite recent research efforts, there is still a lack of universally effective defense strategies that are lossless and highly efficient. The trade-off between utility-privacy-efficiency [190] is still the focus of future studies. Applying multi-objective optimization techniques [166] in VFLow is a promising research direction towards trustworthy VFL [191]. In addition, the path toward a trustworthy FL framework is for the trained models to be verifiable and auditable. One possible route is for the released trained models in VFL to be protected by verifiable intellectual property (IP) protection methods [192] in an efficient manner. Blockchain is leveraged to address the issue that a vanilla FL framework heavily relies on a central server, which means the system is vulnerable to this party's mal-behavior. How to integrate Blockchain into VFL frameworks to improve the overall security and robustness is an interesting future direction.

Automated VFL: Automated machine learning (AutoML) is of great interest in alleviating human effort and achieving satisfactory model performance [193]. Neural Architecture Search (NAS) techniques have been explored in VFL [74], [194]. For VFL, participants without labels can not perform training or evaluation locally. Thus, their hyperparameters are nested in the collaborative training. This unique characteristic makes AutoML in the VFL setting more challenging. By utilizing Blockchain, exchanging parties' local model updates is made possible on this network without the need of a central server.

X. Concluding Remarks

Vertical federated learning has become an attractive solution for solving industrial feature-partitioned data silo problems. . Despite its practical usefulness, evidenced by a growing number of VFL projects and use cases, the breadth and depth of the research advances still lag behind those of HFL. We present an extensive categorization of research advances and challenges in VFL and propose a novel framework towards comprehensively formulating relevant aspects of VFL. We hope this work will encourage future research efforts to address challenges in this area.

References

[1] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated learning," *Synth. Lectures Artif. Intell. Mach. Learn.*, vol. 13, no. 3, pp. 1–207, 2019.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist.*, PMLR, 2017, pp. 1273–1282.

[3] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, Jan. 2019.

[4] Y. Cheng, Y. Liu, T. Chen, and Q. Yang, "Federated learning for privacy-preserving ai," *Commun. ACM*, vol. 63, no. 12, pp. 33–36, 2020.

[5] C. Ju, D. Gao, R. Mane, B. Tan, Y. Liu, and C. Guan, "Federated transfer learning for EEG signal classification," in *Proc. IEEE 42nd Annu. Int. Conf. Eng. Med. Biol. Soc.*, 2020, pp. 3040–3045.

[6] P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, no. 1/2, pp. 1–210, 2021.

[7] Q. Li et al., "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Trans. Knowl. Data Eng.*, 2021.

[8] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Comput. Ind. Eng.*, vol. 149, 2020, Art. no. 106854.

[9] Y. Liu, T. Fan, T. Chen, Q. Xu, and Q. Yang, "Fate: An industrial grade platform for collaborative learning with data protection," *J. Mach. Learn. Res.*, vol. 22, no. 226, pp. 1–6, 2021.

[10] D. Romanini et al., "PyVertical: A vertical federated learning framework for multi-headed splitNN," 2021, arXiv: *2104.00489*.

[11] Bytedance. Fedlearner: Vertical federated GBDT model. 2023. [Online]. Available: https://github.com/bytedance/fedlearner/tree/master/example/tree_model#readme

[12] C. He et al., "FedML: A research library and benchmark for federated machine learning," 2020, arXiv: *2007.13518*.

[13] Q. Li, Y. Cai, Y. Han, C. M. Yung, T. Fu, and B. He, "FedTree: A federated learning system for trees," in *Proc. Mach. Learn. Syst.*, 2023, vol. 5.

[14] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 12, pp. 9587–9603, Dec. 2023.

[15] N.-P. Tran, N.-N. Dao, T.-V. Nguyen, and S. Cho, "Privacy-preserving learning models for communication: A tutorial on advanced split learning," in *Proc. 13th Int. Conf. Inf. Commun. Technol. Convergence*, 2022, pp. 1059–1064.

[16] Y. Liu et al., "FedBCD: A communication-efficient collaborative learning framework for distributed features," *IEEE Trans. Signal Process.*, vol. 70, pp. 4277–4290, 2022.

[17] K. Cheng et al., "SecureBoost: A lossless federated learning framework," *IEEE Intell. Syst.*, vol. 36, no. 6, pp. 87–98, Nov./Dec. 2021.

[18] C. Gratton, N. K. Venkategowda, R. Arablouei, and S. Werner, "Distributed ridge regression with feature partitioning," in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, 2018, pp. 1423–1427.

[19] A. Gascón et al., "Secure linear regression on vertically partitioned datasets," *IACR Cryptol. ePrint Arch.*, vol. 2016, 2016, Art. no. 892.

[20] A. P. Sanil, A. F. Karr, X. Lin, and J. P. Reiter, "Privacy preserving regression modelling via distributed computation," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, New York, NY, USA, 2004, pp. 677–682.

[21] C. Chen et al., "When homomorphic encryption marries secret sharing: Secure large-scale sparse logistic regression and applications in risk control," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2021, pp. 2652–2662.

[22] H. Yu, J. Vaidya, and X. Jiang, "Privacy-preserving SVM classification on vertically partitioned data," in *Proc. Pacific-Asia Conf. Knowl. Discov. Data Mining*, Springer, 2006, pp. 647–656.

[23] R. Xu, N. Baracaldo, Y. Zhou, A. Anwar, J. Joshi, and H. Ludwig, "FedV: Privacy-preserving federated learning over vertically partitioned data," in *Proc. 14th ACM Workshop Artif. Intell. Secur.*, 2021, pp. 181–192.

[24] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, "A secure federated transfer learning framework," *IEEE Intell. Syst.*, vol. 35, no. 4, pp. 70–82, Jul./Aug. 2020.

[25] Y. Hu, D. Niu, J. Yang, and S. Zhou, "FDML: A collaborative machine learning framework for distributed features," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, New York, NY, USA, 2019, pp. 2232–2240.

[26] C. Fu et al., "Label inference attacks against vertical federated learning," in *Proc. 31st USENIX Secur. Symp.*, Boston, MA, USENIX Association, 2022, pp. 1397–1414.

[27] Z. Li, T. Wang, and N. Li, "Differentially private vertical federated clustering," 2022, arXiv: *2208.01700*.

[28] W. Ou, J. Zeng, Z. Guo, W. Yan, D. Liu, and S. Fuentes, "A homomorphic-encryption-based vertical federated learning scheme for rick management," *Comput. Sci. Inf. Syst.*, vol. 17, no. 3, pp. 819–834, 2020.

[29] X. Jin, P.-Y. Chen, C.-Y. Hsu, C.-M. Yu, and T. Chen, "CAFE: Catastrophic data leakage in vertical federated learning," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 994–1006, 2021.

[30] J. Liu, C. Xie, K. Kenthapadi, O. O. Koyejo, and B. Li, "RVFR: Robust vertical federated learning via feature subspace recovery," in *Proc. NeurIPS Workshop New Front. Federated Learn.*, 2021.

[31] T. Zou et al., "Defending batch-level label inference and replacement attacks in vertical federated learning," *IEEE Trans. Big Data*, early access, Jul. 19, 2022, doi: 10.1109/TBDATA.2022.3192121.

[32] I. Ceballos et al., "SplitNN-driven vertical partitioning," 2020, arXiv: *2008.04137*.

[33] Z. Wu, Q. Li, and B. He, "A coupled design of exploiting record similarity for practical vertical federated learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 21087–21100.

[34] L. Wan, W. K. Ng, S. Han, and V. C. S. Lee, "Privacy-preservation for gradient descent methods," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, New York, NY, USA, 2007, pp. 775–783.

[35] W. Chen, G. Ma, T. Fan, Y. Kang, Q. Xu, and Q. Yang, "SecureBoost: A high performance gradient boosting tree framework for large scale vertical federated learning," 2021, arXiv: *2110.10927*.

[36] Z. Feng et al., "SecureGBM: Secure multi-party gradient boosting," in *Proc. IEEE Int. Conf. Big Data*, 2019, pp. 1312–1321.

[37] W. Fang et al., "Large-scale secure XGB for vertical federated learning," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, New York, NY, USA, 2021, pp. 443–452.

[38] Z. Tian, R. Zhang, X. Hou, J. Liu, and K. Ren, "FederBoost: Private federated learning for GBDT," 2020, arXiv: *2011.02796*.

[39] X. Li et al., “OpBoost: A vertical federated tree boosting framework based on order-preserving desensitization,” in *Proc. 49th Int. Conf. Very Large Data Bases*, 2022, pp. 202–215.

[40] L. Xie, J. Liu, S. Lu, T.-H. Chang, and Q. Shi, “An efficient learning framework for federated XGBoost using secret sharing and distributed optimization,” *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 5, pp. 1–28, Sep. 2022.

[41] T. K. Ho, “Random decision forests,” in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, 1995, pp. 278–282.

[42] T. Castiglia, S. Wang, and S. Patterson, “Flexible vertical federated learning with heterogeneous parties,” 2022, arXiv: *2208.12672*.

[43] J. Zhang et al., “Adaptive vertical federated learning on unbalanced features,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 12, pp. 4006–4018, Dec. 2022.

[44] C. Xie, P.-Y. Chen, C. Zhang, and B. Li, “Improving privacy-preserving vertical federated learning by efficient communication with admm,” 2022, arXiv: *2207.10226*.

[45] F. Fu, X. Miao, J. Jiang, H. Xue, and B. Cui, “Towards communication-efficient vertical federated learning training via cache-enabled local updates,” *Proc. VLDB Endowment*, vol. 15, no. 10, pp. 2111–2120, Jun. 2022.

[46] M. Li, Y. Chen, Y. Wang, and Y. Pan, “Efficient asynchronous vertical federated learning via gradient prediction and double-end sparse compression,” in *Proc. 16th Int. Conf. Control, Automation, Robot. Vis.*, 2020, pp. 291–296.

[47] D. Cai et al., “Accelerating vertical federated learning,” *IEEE Trans. Big Data*, early access, Jul. 21, 2022, doi: 10.1109/TBDATA.2022.3192898.

[48] Z. Zhang, G. Zhu, and S. Cui, “Low-latency cooperative spectrum sensing via truncated vertical federated learning,” 2022, arXiv: *2208.03694*.

[49] T. Chen, X. Jin, Y. Sun, and W. Yin, “VAFL: A method of vertical asynchronous federated learning,” in *Proc. ICML Workshop Federated Learn. User Privacy Data Confidentiality*, 2020.

[50] B. Gu, A. Xu, Z. Huo, C. Deng, and H. Huang, “Privacy-preserving asynchronous vertical federated learning algorithms for multiparty collaborative learning,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6103–6115, Nov. 2022.

[51] Q. Zhang et al., “AsySQN: Faster vertical federated learning algorithms with better computation resource utilization,” in *Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, New York, NY, USA, 2021, pp. 3917–3927.

[52] Q. Zhang, B. Gu, C. Deng, and H. Huang, “Secure bilevel asynchronous vertical federated learning with backward updating,” in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 10896–10904.

[53] B. Gu, Z. Dang, X. Li, and H. Huang, “Federated doubly stochastic kernel learning for vertically partitioned data,” in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, New York, NY, USA, 2020, pp. 2483–2493.

[54] Y. Han, P. Du, and K. Yang, “FedGBF: An efficient vertical federated learning framework via gradient boosting and bagging,” 2022, arXiv: *2204.00976*.

[55] F. Fu et al., “VF2Boost: Very fast vertical federated gradient boosting for cross-enterprise learning,” in *Proc. Int. Conf. Manage. Data*, 2021, pp. 563–576.

[56] Z. Wu, Q. Li, and B. He, “Practical vertical federated learning with unsupervised representation learning,” *IEEE Trans. Big Data*, Jun. 06, 2022, doi: 10.1109/TBDATA.2022.3180117.

[57] D. Cha et al., “Implementing vertical federated learning using autoencoders: Practical application, generalizability, and utility study,” *JMIR Med. Inform.*, vol. 9, no. 6, 2021, Art. no. e26598.

[58] A. Khan, M. ten Thij, and A. Wilbik, “Communication-efficient vertical federated learning,” *Algorithms*, vol. 15, no. 8, 2022, Art. no. 273.

[59] W. Xu, H. Fan, K. Li, and K. Yang, “Efficient batch homomorphic encryption for vertically federated XGBoost,” 2021, arXiv: *2112.04261*.

[60] T. J. Castiglia, A. Das, S. Wang, and S. Patterson, “Compressed-VFL: Communication-efficient learning with vertically partitioned data,” in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2022, pp. 2738–2766.

[61] L. Huang, Z. Li, J. Sun, and H. Zhao, “Coresets for vertical federated learning: Regularized linear regression and k-means clustering,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 29566–29581.

[62] A. Li et al., “FedSDG-FS: Efficient and secure feature selection for vertical federated learning,” in *Proc. IEEE Int. Conf. Comput. Commun.*, 2023, pp. 1–10.

[63] R. Zhang, H. Li, M. Hao, H. Chen, and Y. Zhang, “Secure feature selection for vertical federated learning in ehealth systems,” in *Proc. IEEE Int. Conf. Commun.*, 2022, pp. 1257–1262.

[64] T. Castiglia, Y. Zhou, S. Wang, S. Kadhe, N. Baracaldo, and S. Patterson, “LESS-VFL: Communication-efficient feature selection for vertical federated learning,” in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 3757–3781.

[65] R. Fu, Y. Wu, Q. Xu, and M. Zhang, “FEAST: A communication-efficient federated feature selection framework for relational data,” *Proc. ACM Manage. Data*, vol. 1, no. 1, pp. 1–28, 2023.

[66] S. Feng, “Vertical federated learning-based feature selection with non-overlapping sample utilization,” *Expert Syst. Appl.*, vol. 208, 2022, Art. no. 118097.

[67] G.-D. Zhang, S.-Y. Zhao, H. Gao, and W.-J. Li, “Feature-distributed SVRG for high-dimensional linear classification,” 2018, arXiv: *1802.03604*.

[68] P. Bojanowski and A. Joulin, “Unsupervised learning by predicting noise,” in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 517–526.

[69] Y. Mao et al., “Communication-efficient federated learning with adaptive quantization,” *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, pp. 1–26, Aug. 2022.

[70] D. Jhunjhunwala, A. Gadhikar, G. Joshi, and Y. C. Eldar, “Adaptive quantization of model updates for communication-efficient federated learning,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 3110–3114.

[71] G. Yan, T. Li, S.-L. Huang, T. Lan, and L. Song, “AC-SGD: Adaptively compressed SGD for communication-efficient distributed learning,” *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2678–2693, Sep. 2022.

[72] W. Li et al., “VFed-SSD: Towards practical vertical federated advertising,” in *Proc. Int. Workshop Trustworthy Federated Learn. Conjunction IJCAI*, 2022.

[73] Y. He, Y. Kang, J. Luo, L. Fan, and Q. Yang, “A hybrid self-supervised learning framework for vertical federated learning,” 2022, arXiv: *2208.08934*.

[74] Y. Liu et al., “Cross-silo federated neural architecture search for heterogeneous and cooperative systems,” in *Federated and Transfer Learning*. Berlin, Germany: Springer, 2023, pp. 57–86.

[75] Y. Kang, Y. Liu, and X. Liang, “FedCVT: Semi-supervised vertical federated learning with cross-view training,” *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, pp. 1–16, May 2022.

[76] Y. Yang, X. Ye, and T. Sakurai, “Multi-view federated learning with data collaboration,” in *Proc. 14th Int. Conf. Mach. Learn. Comput.*, New York, NY, USA, 2022, pp. 178–183.

[77] C. Huang, L. Wang, and X. Han, “Vertical federated knowledge transfer via representation distillation for healthcare collaboration networks,” in *Proc. ACM Web Conf.*, New York, NY, USA, 2023, pp. 4188–4199.

[78] Z. Ren, L. Yang, and K. Chen, “Improving availability of vertical federated learning: Relaxing inference on non-overlapping data,” *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, pp. 1–20, 2022.

[79] W. Li, Q. Xia, H. Cheng, K. Xue, and S.-T. Xia, “Vertical semi-federated learning for efficient online advertising,” in *Proc. Int. Workshop Trustworthy Federated Learn. Conjunction IJCAI*, 2023.

[80] S. Feng and H. Yu, “Multi-participant multi-class vertical federated learning,” 2020, arXiv: *2001.11154*.

[81] S. Feng, B. Li, H. Yu, Y. Liu, and Q. Yang, “Semi-supervised federated heterogeneous transfer learning,” *Knowl.-Based Syst.*, vol. 252, 2022, Art. no. 109384.

[82] S. Sharma, C. Xing, Y. Liu, and Y. Kang, “Secure and efficient federated transfer learning,” in *Proc. IEEE Int. Conf. Big Data*, 2019, pp. 2569–2576.

[83] Y. Kang, Y. He, J. Luo, T. Fan, Y. Liu, and Q. Yang, “Privacy-preserving federated adversarial domain adaptation over feature groups for interpretability,” *IEEE Trans. Big Data*, early access, Jul. 11, 2022, doi: 10.1109/TBDATA.2022.3188292.

[84] A. Imakura and T. Sakurai, “Data collaboration analysis framework using centralization of individual intermediate representations for distributed data sets,” *ASCE-ASME J. Risk Uncertainty Eng. Syst.*, vol. 6, no. 2, 2020, Art. no. 04020018.

[85] D. Chai et al., “Practical lossless federated singular vector decomposition over billion-scale data,” in *Proc. 28th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, New York, NY, USA, 2022, pp. 46–55.

[86] I. Damgård, V. Pastro, N. Smart, and S. Zakarias, “Multiparty computation from somewhat homomorphic encryption,” in *Proc. Annu. Cryptol. Conf.*, Springer, 2012, pp. 643–662.

[87] G. Liang and S. S. Chawathe, “Privacy-preserving inter-database operations,” in *Proc. Int. Conf. Intell. Secur. Informat.*, Springer, 2004, pp. 66–82.

[88] B. Pinkas, T. Schneider, and M. Zohner, “Faster private set intersection based on OT extension,” in *Proc. 23rd USENIX Secur. Symp.*, San Diego, CA: USENIX Association, 2014, pp. 797–812.

[89] Z. Zhou, Y. Tian, and C. Peng, “Privacy-preserving federated learning framework with general aggregation and multiparty entity matching,” *Wireless Commun. Mobile Comput.*, vol. 2021, 2021, Art. no. 6692061.

[90] L. Lu and N. Ding, “Multi-party private set intersection in vertical federated learning,” in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun.*, 2020, pp. 707–714.

[91] Y. Liu, X. Zhang, and L. Wang, “Asymmetrical vertical federated learning,” 2020, arXiv: *2004.07427*.

[92] J. Sun et al., “Vertical federated learning without revealing intersection membership,” 2021, arXiv: *2106.05508*.

[93] P. Qiu et al., “Your labels are selling you out: Relation leaks in vertical federated learning,” *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 5, pp. 3653–3668, Sep./Oct. 2023.

[94] X. Luo, Y. Wu, X. Xiao, and B. C. Ooi, “Feature inference attack on model predictions in vertical federated learning,” in *Proc. IEEE 37th Int. Conf. Data Eng.*, 2021, pp. 181–192.

[95] Z. He, T. Zhang, and R. B. Lee, “Model inversion attacks against collaborative inference,” in *Proc. 35th Annu. Comput. Secur. Appl. Conf.*, 2019, pp. 148–162.

[96] X. Jiang, X. Zhou, and J. Grossklags, “Comprehensive analysis of privacy leakage in vertical federated learning during prediction,” *Proc. Privacy Enhancing Technol.*, vol. 2022, no. 2, pp. 263–281, 2022.

[97] S. Hardy et al., “Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption,” 2017, arXiv: *1711.10677*.

[98] Y. Wu, S. Cai, X. Xiao, G. Chen, and B. C. Ooi, “Privacy preserving vertical federated learning for tree-based models,” *Proc. VLDB Endowment*, vol. 13, no. 12, pp. 2090–2103, 2020.

[99] F. Fu, H. Xue, Y. Cheng, Y. Tao, and B. Cui, “BlindFL: Vertical federated machine learning without peeking into your data,” in *Proc. Int. Conf. Manage. Data*, 2022, pp. 1316–1330.

[100] Webank, “Federated logistic regression,” 2022. [Online]. Available: https://github.com/FederatedAI/FATE/blob/master/doc/federatedml_component/logistic_regression.md

[101] O. Li et al., “Label leakage and protection in two-party split learning,” 2021, arXiv: *2102.08504*.

[102] J. Tan, L. Zhang, Y. Liu, A. Li, and Y. Wu, “Residue-based label protection mechanisms in vertical logistic regression,” 2022, arXiv: *2205.04166*.

[103] S. Kariyappa and M. K. Qureshi, “Exploit: Extracting private labels in split learning,” in *Proc. IEEE Conf. Secure Trustworthy Mach. Learn.*, 2023, pp. 165–175.

[104] J. Sun, X. Yang, Y. Yao, and C. Wang, “Label leakage and protection from forward embedding in vertical federated learning,” 2022, arXiv: *2203.01451*.

[105] P. Ye, Z. Jiang, W. Wang, B. Li, and B. Li, “Feature reconstruction attacks and countermeasures of DNN training in vertical federated learning,” 2022, arXiv: *2210.06771*.

[106] H. Weng, J. Zhang, F. Xue, T. Wei, S. Ji, and Z. Zong, “Privacy leakage of real-world vertical federated learning,” 2020, arXiv: *2011.09290*.

[107] Y. Hu et al., “Is vertical logistic regression privacy-preserving? a comprehensive privacy analysis and beyond,” 2022, arXiv: *2207.09087*.

[108] C. Song and V. Shmatikov, “Overlearning reveals sensitive attributes,” in *Proc. Int. Conf. Learn. Representations*, 2020.

[109] S. Yang, B. Ren, X. Zhou, and L. Liu, “Parallel distributed logistic regression for vertical federated learning without third-party coordinator,” in *Proc. Workshop Federated Mach. Learn. User Privacy Data Confidentiality*, 2019.

[110] D. He, R. Du, S. Zhu, M. Zhang, K. Liang, and S. Chan, “Secure logistic regression for vertical federated learning,” *IEEE Internet Comput.*, vol. 26, no. 2, pp. 61–68, Mar./Apr. 2022.

[111] W. D. Tsz-Him Cheung and S. Li, “FedSGC: Federated simple graph convolution for node classification,” in *Proc. Int. Joint Conf. Artif. Intell. Workshops*, 2021.

[112] Y. Zhang and H. Zhu, “Additively homomorphical encryption based deep neural network for asymmetrically collaborative machine learning,” 2020, arXiv: *2007.06849*.

[113] J. G. Chamani and D. Papadopoulos, “Mitigating leakage in federated learning with trusted hardware,” in *Proc. Privacy Preserving Mach. Learn. Workshop NeurIPS*, 2020.

[114] M. W. Mahoney, “Randomized algorithms for matrices and data,” *Found. Trends Mach. Learn.*, vol. 3, no. 2, pp. 123–224, Feb. 2011.

[115] F. Zheng, C. Chen, B. Yao, and X. Zheng, “Making split learning resilient to label leakage by potential energy loss,” 2022, arXiv: *2210.09617*.

[116] J. Tan, L. Zhang, Y. Liu, A. Li, and Y. Wu, “Residue-based label protection mechanisms in vertical logistic regression,” 2022. [Online]. Available: https://arxiv.org/abs/2205.04166

[117] H. Gu, J. Luo, Y. Kang, L. Fan, and Q. Yang, “FedPass: Privacy-preserving vertical federated deep learning with adaptive obfuscation,” in *Proc. 32nd Int. Joint Conf. Artif. Intell.*, 2023, pp. 3759–3767.

[118] J. Sun, Y. Yao, W. Gao, J. Xie, and C. Wang, “Defending against reconstruction attack in vertical federated learning,” 2021, arXiv: *2107.09898*.

[119] N. Dryden, T. Moon, S. A. Jacobs, and B. Van Essen, “Communication quantization for data-parallel training of deep neural networks,” in *Proc. 2nd Workshop Mach. Learn. HPC Environments*, 2016, pp. 1–8.

[120] A. F. Aji and K. Heafield, “Sparse communication for distributed gradient descent,” 2017, arXiv: *1704.05021*.

[121] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, “Deep gradient compression: Reducing the communication bandwidth for distributed training,” 2017, arXiv: *1712.01887*.

[122] R. Shokri and V. Shmatikov, “Privacy-preserving deep learning,” in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 1310–1321.

[123] L. Zhu, Z. Liu, and S. Han, “Deep leakage from gradients,” in *Proc. Adv. Neural Inf. Process. Syst.*, Curran Associates, Inc., 2019, pp. 14774–14784.

[124] C. Dwork, “Differential privacy: A survey of results,” in *Proc. Int. Conf. Theory Appl. Models Comput.*, Springer, 2008, pp. 1–19.

[125] C. Wang, J. Liang, M. Huang, B. Bai, K. Bai, and H. Li, “Hybrid differentially private federated learning on vertically partitioned data,” 2020, arXiv: *2009.02763*.

[126] C. Chen et al., “Vertically federated graph neural network for privacy-preserving node classification,” in *Proc. 31st Int. Joint Conf. Artif. Intell.*, 2022, pp. 1959–1965.

[127] J. Jia and N. Z. Gong, “AttriGuard: A practical defense against attribute inference attacks via adversarial machine learning,” in *Proc. 27th USENIX Secur. Symp.*, 2018, pp. 513–529.

[128] Y. Liu, Z. Yi, and T. Chen, “Backdoor attacks and defenses in feature-partitioned collaborative learning,” 2020, arXiv: *2007.03608*.

[129] Q. Pang, Y. Yuan, and S. Wang, “Attacking vertical collaborative learning system using adversarial dominating inputs,” 2022, arXiv: *2201.02775*.

[130] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” *Stat*, vol. 1050, 2015, Art. no. 20.

[131] J. Chen, G. Huang, H. Zheng, S. Yu, W. Jiang, and C. Cui, “Graph-fraudster: Adversarial attacks on graph neural network-based vertical federated learning,” *IEEE Trans. Comput. Social Syst.*, vol. 10, no. 2, pp. 492–506, Apr. 2023.

[132] T. Wang, J. Rausch, C. Zhang, R. Jia, and D. Song, “A principled approach to data valuation for federated learning,” in *Federated Learning*, Berlin, Germany: Springer, 2020, pp. 153–167.

[133] Z. Liu, Y. Chen, H. Yu, Y. Liu, and L. Cui, “GTG-Shapley: Efficient and accurate participant contribution evaluation in federated learning,” *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, pp. 1–21, 2022.

[134] G. Wang, C. X. Dang, and Z. Zhou, “Measure contribution of participants in federated learning,” in *Proc. IEEE Int. Conf. Big Data*, 2019, pp. 2597–2604.

[135] G. Wang, “Interpret federated learning with shapley values,” 2019, arXiv: *1905.04519*.

[136] X. Han, L. Wang, and J. Wu, “Data valuation for vertical federated learning: An information-theoretic approach,” 2021, arXiv: *2112.08364*.

[137] Z. Fan, H. Fang, Z. Zhou, J. Pei, M. P. Friedlander, and Y. Zhang, “Fair and efficient contribution valuation for vertical federated learning,” 2022, arXiv: *2201.02658*.

[138] J. Jiang et al., “VF-PS: How to select important participants in vertical federated learning, efficiently and securely?,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 2088–2101.

[139] P. Chen, X. Du, Z. Lu, J. Wu, and P. C. Hung, “Evfl: An explainable vertical federated learning for data-oriented artificial intelligence systems,” *J. Syst. Archit.*, vol. 126, 2022, Art. no. 102474.

[140] F. Zheng et al., “A vertical federated learning method for interpretable scorecard and its application in credit scoring,” 2020, arXiv: *2009.06218*.

[141] T. Qi et al., “Fairvfl: A fair vertical federated learning framework with contrastive adversarial learning,” *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 7852–7865, 2022.

[142] C. Liu, Z. Zhou, Y. Shi, J. Pei, L. Chu, and Y. Zhang, “Achieving model fairness in vertical federated learning,” 2021, arXiv: *2109.08344*.

[143] R. Kohavi, “Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid,” in *Proc. 2nd Int. Conf. Knowl. Discov. Data Mining*, 1996, pp. 202–207.

[144] S. Moro, P. Cortez, and P. Rita, “A data-driven approach to predict the success of bank telemarketing,” *Decis. Support Syst.*, vol. 62, pp. 22–31, 2014.

[145] I.-C. Yeh and C.-h. Lien, "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 2473–2480, 2009.

[146] Kaggle, "Give me some credit dataset," 2011. [Online]. Available: https://www.kaggle.com/c/GiveMeSomeCredit

[147] A. E. Johnson et al., "MIMIC-III, a freely accessible critical care database," *Sci. Data*, vol. 3, 2016, Art. no. 160035.

[148] W. N. Street, W. H. Wolberg, and O. L. Mangasarian, "Nuclear feature extraction for breast tumor diagnosis," in *Proc. Biomed. Image Process. Biomed. Visual.*, 1993, pp. 861–870.

[149] J. W. Smith, J. E. Everhart, W. Dickson, W. C. Knowler, and R. S. Johannes, "Using the ADAP learning algorithm to forecast the onset of diabetes mellitus," in *Proc. Annu. Symp. Comput. Appl. Med. Care*, 1988, pp. 261–265.

[150] Kaggle, "Avazu dataset," [Online]. Available: https://www.kaggle.com/c/avazu-ctr-prediction

[151] Criteo-Labs, "Criteo dataset," [Online]. Available: https://labs.criteo.com/2014/02/download-kaggle-display-advertising-challenge-dataset/

[152] M. F. Duarte and Y. H. Hu, "Vehicle classification in distributed sensor networks," *J. Parallel Distrib. Comput.*, vol. 64, no. 7, pp. 826–838, 2004.

[153] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[154] J. A. Blackard and D. J. Dean, "Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables," *Comput. Electron. Agriculture*, vol. 24, no. 3, pp. 131–151, 1999.

[155] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng, "NUS-WIDE: A real-world web image database from national university of Singapore," in *Proc. ACM Conf. Image Video Retrieval*, Santorini, Greece., 2009, pp. 1–9.

[156] M. van Breukelen, R. P. Duin, D. M. Tax, and J. Den Hartog, "Handwritten digit recognition by combined classifiers," *Kybernetika*, vol. 34, no. 4, pp. 381–386, 1998.

[157] PASCAL-Challenge-2008, "Epsilon dataset," 2008. [Online]. Available: https://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets/

[158] P. Mooney, "Breast histopathology images," 2017. [Online]. Available: https://www.kaggle.com/datasets/paultimothymooney/breast-histopathology-images

[159] J. Irvin et al., "Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 590–597.

[160] Z. Wu et al., "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1912–1920.

[161] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore, "Automating the construction of internet portals with machine learning," *Inf. Retrieval*, vol. 3, pp. 127–163, 2000.

[162] C. L. Giles, K. D. Bollacker, and S. Lawrence, "Citeseer: An automatic citation indexing system," in *Proc. 3rd ACM Conf. Digit. Libraries*, 1998, pp. 89–98.

[163] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI Mag.*, vol. 29, no. 3, pp. 93–93, 2008.

[164] S. Rakshit, "Yahoo answers dataset," [Online]. Available: https://www.kaggle.com/soumikrakshit/yahoo-answers-dataset

[165] S. S. Keerthi, D. DeCoste, and T. Joachims, "A modified finite newton method for fast solution of large scale linear svms," *J. Mach. Learn. Res.*, vol. 6, no. 3, pp. 341–361, 2005.

[166] Y. Kang et al., "Optimizing privacy, utility and efficiency in constrained multi-objective federated learning," 2023, arXiv: *2305.00312*.

[167] Z. Liu, L. Song, and C. Fragouli, "Federated multi-armed bandits with vector rewards for aspect-based recommendations," in *Proc. IEEE Glob. Commun. Conf.*, 2022, pp. 1079–1084.

[168] T. Li and L. Song, "Privacy-preserving communication-efficient federated multi-armed bandits," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 3, pp. 773–787, Mar. 2022.

[169] Z. Zhu, J. Zhu, J. Liu, and Y. Liu, "Federated bandit: A gossiping approach," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 5, no. 1, pp. 1–29, Feb. 2021. [Online]. Available: https://doi.org/10.1145/3447380

[170] E. Shmueli and T. Tassa, "Secure multi-party protocols for item-based collaborative filtering," in *Proc. 11th ACM Conf. Recommender Syst.*, New York, NY, USA, 2017, pp. 89–97.

[171] K. Atarashi and M. Ishihata, "Vertical federated learning for higher-order factorization machines," in *Proc. Adv. Knowl. Discov. Data Mining*, 2021, pp. 346–357.

[172] J. Cui, C. Chen, L. Lyu, C. Yang, and W. Li, "Exploiting data sparsity in secure cross-platform social recommendation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 10524–10534.

[173] J. Zhang and Y. Jiang, "A vertical federation recommendation method based on clustering and latent factor model," in *Proc. Int. Conf. Electron. Inf. Eng. Comput. Sci.*, 2021, pp. 362–366.

[174] H. Yuan, C. Ma, Z. Zhao, X. Xu, and Z. Wang, "A privacy-preserving oriented service recommendation approach based on personal data cloud and federated learning," in *Proc. IEEE Int. Conf. Web Serv.*, 2022, pp. 322–330.

[175] F. Cai, "Bytedance breaks federated learning: Open source fedlearner framework, 209% increase in advertising efficiency," Accessed: Mar. 15, 2021, 2020. [Online]. Available: https://www.jiqizhixin.com/articles/2020--11-03-9

[176] Y. Hou, "JD's exploration and practice of large-scale federated learning," Accessed: May 31, 2021, 2021. [Online]. Available: https://zhuanlan.zhihu.com/p/376697402

[177] Y. Lin, "The practice of federated learning in tencent wesee advertising," 2021. [Online]. Available: https://cloud.tencent.com/developer/article/1872819

[178] Y. Wu, "Huawei's exploration and application in federated advertising algorithm," 2022. [Online]. Available: https://zhuanlan.zhihu.com/p/558684266

[179] G. Long, Y. Tan, J. Jiang, and C. Zhang, "Federated learning for open banking," in *Federated Learning*. Berlin, Germany: Springer, 2020, pp. 240–254.

[180] T. Rooijakkers, "CONVINCED—Enabling privacy-preserving survival analyses using multi-party computation," 2020.

[181] Y. Song et al., "Federated learning application on telecommunication-joint healthcare recommendation," in *Proc. IEEE 21st Int. Conf. Commun. Technol.*, 2021, pp. 1443–1448.

[182] Z. Teimoori, A. Yassine, and M. S. Hossain, "A secure cloudlet-based charging station recommendation for electric vehicles empowered by federated learning," *IEEE Trans. Ind. Inform.*, vol. 18, no. 9, pp. 6464–6473, Sep. 2022.

[183] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 46–51, Jun. 2020.

[184] Y. Zhang, Q. Wu, and M. Shikh-Bahaei, "Vertical federated learning based privacy-preserving cooperative sensing in cognitive radio networks," in *Proc. IEEE Globecom Workshops*, 2020, pp. 1–6.

[185] N. Hashemi, P. Safari, B. Shariati, and J. K. Fischer, "Vertical federated learning for privacy-preserving ML model development in partially disaggregated networks," in *Proc. Eur. Conf. Opt. Commun.*, 2021, pp. 1–4.

[186] H. Liu, X. Zhang, X. Shen, and H. Sun, "A federated learning framework for smart grids: Securing power traces in collaborative learning," 2021, arXiv: *2103.11870*.

[187] N. Ge, G. Li, L. Zhang, and Y. Liu, "Failure prediction in production line based on federated learning: An empirical study," *J. Intell. Manuf.*, vol. 33, pp. 2277–2294, 2021.

[188] F. Liu, X. Wu, S. Ge, W. Fan, and Y. Zou, "Federated learning for vision-and-language grounding problems," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, pp. 11572–11579, 2020.

[189] X. Ni, X. Xu, L. Lyu, C. Meng, and W. Wang, "A vertical federated learning framework for graph convolutional network," 2021, arXiv: *2106.11593*.

[190] X. Zhang, Y. Kang, K. Chen, L. Fan, and Q. Yang, "Trading off privacy, utility and efficiency in federated learning," *ACM Trans. Intell. Syst. Technol.*, vol. 14, no. 6, pp. 1–32, 2022.

[191] Z. Ren et al., "SecureBoost hyperparameter tuning via multi-objective federated learning," in *Proc. Int. Workshop Trustworthy Federated Learn. Conjunction IJCAI*, 2023.

[192] B. Li, L. Fan, H. Gu, J. Li, and Q. Yang, "FediPR: Ownership verification for federated deep neural network models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4521–4536, Apr. 2023.

[193] Q. Yao et al., "Taking human out of learning applications: A survey on automated machine learning," Oct. 2018, *arXiv:1810.13306*.

[194] H. Zhu, H. Zhang, and Y. Jin, "From federated learning to federated neural architecture search: A survey," *Complex Intell. Syst.*, vol. 7, no. 2, pp. 639–657, 2021.

Yang Liu (Senior Member, IEEE) is an associate professor with the Institute for AI Industry Research of Tsinghua University. Her research work was recognized with multiple awards, such as AAAI Innovation Award. She co-authored several books on the topic of federated learning and is recognized by MIT Technology Review as the “Privacy-Preserving Computation Tech Innovators China”.

Yan Kang is currently a research team lead with the AI department of WeBank, Shenzhen, China. His works focus on the research and implementation of privacy-preserving machine learning and federated learning. His research was authored or coauthored in well-known conferences and journals including *IEEE Instruction Set Processor*, *IEEE Transactions on Big Data*, IJCAI, ICDE, and *ACM Transactions on Intelligent Systems and Technology*.

Tianyuan Zou is currently working toward the PhD degree with the Institute for AI Industry Research (AIR), Tsinghua University, China, majoring in computer science and technology. Her current research interest focuses on the privacy and safety in federated learning.

Yanhong Pu (Member, IEEE) is currently working toward the EngD degree with the College of Intelligence and Computing, Tianjin University, China. He is currently a scientific research engineer with Institute for AI Industry Research (AIR), Tsinghua University, China. He current research interest focuses on the privacy and safety in federated learning.

Yuanqin He received the BS degree from Shanghai Jiao Tong University, and the PhD degree in physics from the Technical University of Munich. He is currently a researcher with WeBank. His research interests include machine learning and federated learning.

Xiaozhou Ye is currently the chief scientist of AsiaInfo Technologies. Before joining AsiaInfo, he was a professor with the Institute of Acoustics, Chinese Academy of Sciences, and deputy director of an National Engineering Technology Research Center of China. His interests now mainly focus on telecommunication and artificial intelligence.

Ye Ouyang (Fellow, IEEE) is chief technology officer and senior vice president with Asiainfo Technologies. Prior to AsiaInfo, he has been a Verizon fellow and senior manger with Verizon. His R&D interests include innovation and commercialization in the interdisciplinary space of artificial intelligence, data science, and mobile communications from 2G to 6G. He is an Verizon fellow.

Ya-Qin Zhang (Fellow, IEEE) has been a chair professor and dean of Institute for AI Industry Research (AIR), Tsinghua University since 2020. He was a senior executive of several high-tech companies during his 30-year tenure, including the President of Baidu Inc, and Corporate vice president of Microsoft. He was elected to the Chinese Academy of Engineering (CAE), the American Academy of Arts and Sciences (AAA&S), and the Australian Academy of Technology and Engineering (ATSE). He is a fellow of CAAI.

Qiang Yang (Fellow, IEEE) is a fellow of Royal Society of Canada (RSC) and Canadian Academy of Engineering (CAE), Chief Artificial Intelligence Officer of WeBank, a chair professor with the Computer Science and Engineering Department, Hong Kong University of Science and Technology (HKUST). He is a fellow of AAAI, ACM, CAAI, IAPR, and AAAS. His research interests are artificial intelligence, transfer learning and federated learning.