Vertical Federated Learning: Concepts, Advances and Challenges

Yang Liu[1]*, Yan Kang[2], Tianyuan Zou[1], Yanhong Pu[1], Yuanqin He[2], Xiaozhou Ye[3], Ye Ouyang[3], Ya-Qin Zhang[1] and Qiang Yang[2,4]

[1] *Institute for AI Industry Research, Tsinghua University, Beijing, China.*
[2] *Webank, Shenzhen, China.*
[3] *AsiaInfo Technologies, Beijing, China.*
[4] *Hong Kong University of Science and Technology, Hong Kong, China.*

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. 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.

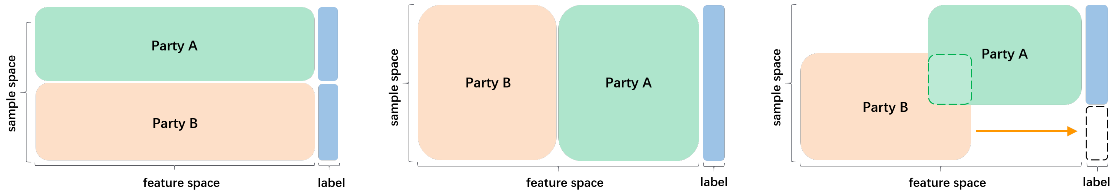
1 Introduction

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 space: Horizontal Federated Learning (HFL), Vertical Federated Learning (VFL) and Federated Transfer Learning (FTL) (See Figure 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].

*corresponding author, liuy03@air.tsinghua.edu.cn.



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

Figure 1: Three categories of Federated Learning

- 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 exchanges model updates (i.e., parameters or gradients) with a server, which aggregates the updates and sends the aggregating result back to each party. While in VFL, each party keeps both 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 time, 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. In contrast, 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 these main differences between HFL, VFL, and FTL in Table 1. Note that Table 1 compares the conventional cases of HFL, VFL, and FTL. As this research area experiences explosive growth, some special cases may deviate from Table 1.

Table 1: 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

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 (AI) 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 two years [9, 10, 11, 12, 13], and the number of commercialized projects as well as the economic values of VFL have grown significantly. Since in VFL, data parties with different attributes of people are typically from different industrial segments, for example, a local bank and a local 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, 14, 15] or a limited perspective of VFL [16, 17]. Therefore, we provide a comprehensive overview of 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.

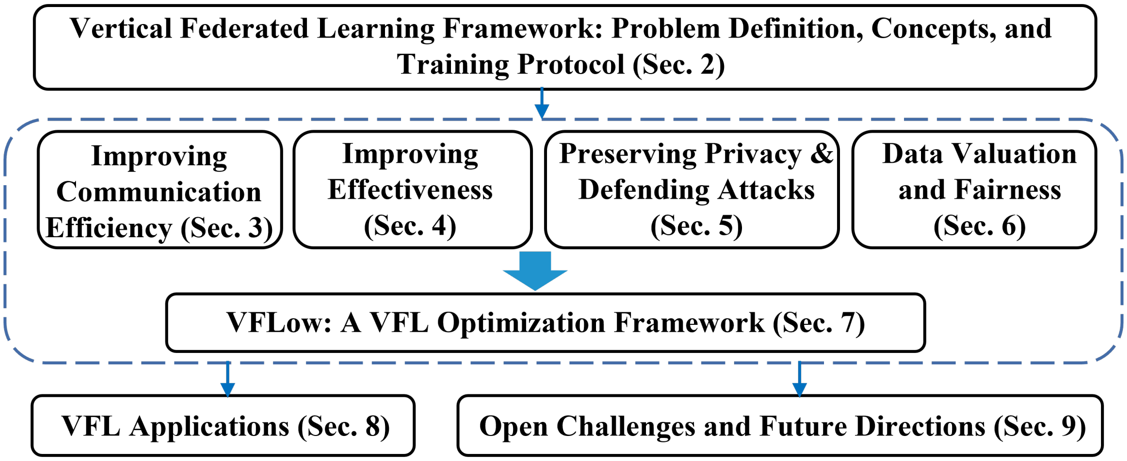

Vertical Federated Learning Framework: Problem Definition, Concepts, and Training Protocol (Sec. 2)
Improving Communication Efficiency (Sec. 3)
Improving Effectiveness (Sec. 4)
Preserving Privacy & Defending Attacks (Sec. 5)
Data Valuation and Fairness (Sec. 6)
VFLow: A VFL Optimization Framework (Sec.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

Figure 2: Relationships between sections in this work.

This paper is organized as follows: Sec. 2 overviews VFL's concepts and training procedures. Building on Sec. 2, Sec. 3, Sec. 4, and Sec. 5 discuss the efficiency, effectiveness, privacy, and security aspects of VFL algorithms. Sec. 6 discusses the challenges of data valuation, explainability, and fairness towards building a VFL ecosystem. Sec. 7 introduces **VFLow**, a VFL optimization framework guiding the design and optimization of VFL algorithms, and Sec. 8 discusses application-oriented algorithms built on VFL. Finally, Sec. 9 discusses open challenges and future directions. Figure 2 dictates the relationships between sections in this work.

2 Vertical Federated Learning framework

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

2.1 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.

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

where $\boldsymbol{\Theta}$ 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.

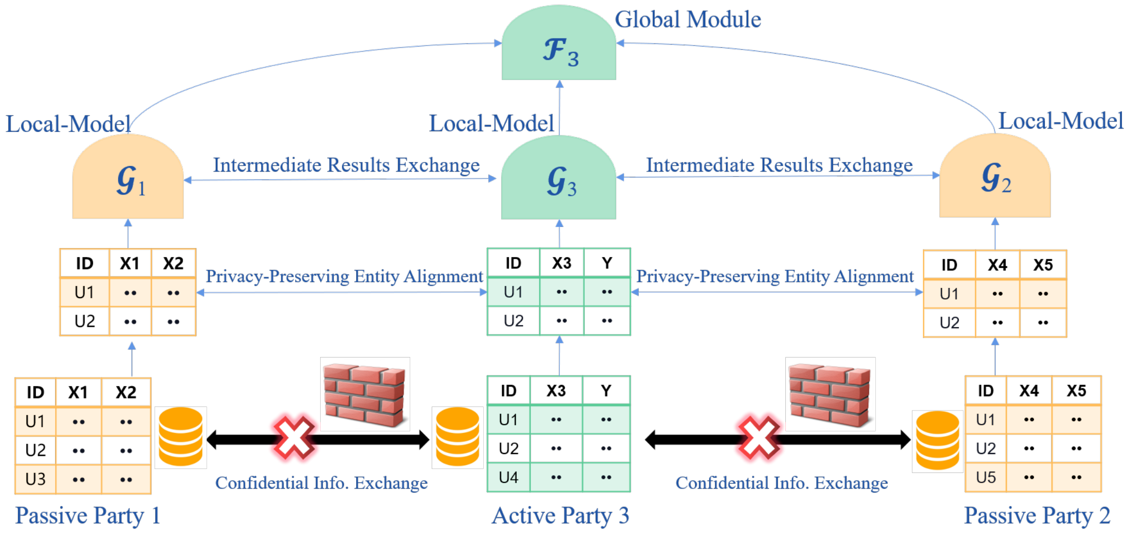


Figure 3: Illustration of the VFL system with three parties (two passive parties and one active party). $\mathcal{G}_1, \mathcal{G}_2$, and $\mathcal{G}_3$ denote the local models of the three parties, respectively, and $\mathcal{F}_3$ denotes the global module owned by the active party. The VFL training 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 2.2 for details).

VFL assumes that data are partitioned by feature space. Following [3, 18], 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 $\mathcal{D}_K \triangleq \{\mathbf{x}_{i,K}, y_{i,K}\}_{i=1}^{N}$.

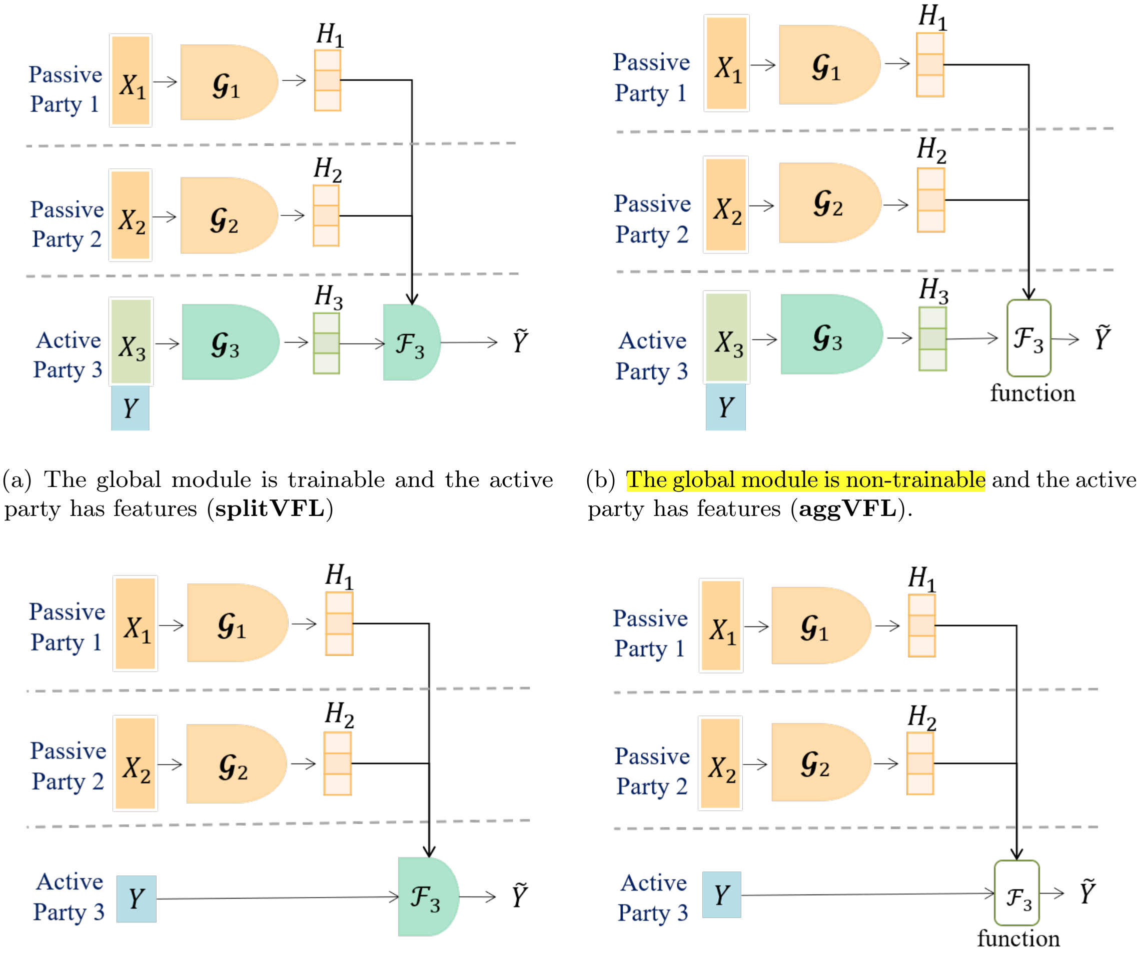
Without loss of generality, we decompose $\mathbf{\Theta}$ into local models $\mathcal{G}_k$ parameterized by θ_k, $k \in \{1, \cdots, K\}$, which operates only on local data, and a global module $\mathcal{F}_K$ parameterized by ψ_K, which is only accessible by the active party K. We rewrite the loss $f(\mathbf{\Theta}; \mathbf{x}_i, y_i)$ as:

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

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

Figure 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 [19], linear and logistic regression (LR) [3, 18, 20, 21, 22, 23], support vector machine [24, 25], neural network (NN) [26, 27, 28], as well as K-means [29] and EM algorithm [30] etc. Although most of the existing VFL works consider linearly separable local models, recent works [31] also proposed kernel methods for incorporating non-linear learning over distributed features.

The global module $\mathcal{F}_K$ can be either *trainable* [28, 32, 33] or *non-trainable*[28, 34]. If a *trainable* global module is in place, this VFL scenario is coincident with the vertical splitNN [35], where the whole model is splitted into different parties, thus we term it **splitVFL** (see Figure 4(a)). If the global module is *non-trainable*, it serves as an aggregation function, such as Sigmoid (for NN) or an optimal split finding function (for tree), that aggregates parties' intermediate results. We term this scenario **aggVFL** (see Figure 4(b)).



(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**).

(c) The global module is trainable and the active party has no feature (**splitVFL**$_c$)

(d) The global module is non-trainable and the active party has no feature (**aggVFL**$_c$)

Figure 4: Four major variants of VFL illustrated with one active party and two passive parties.

Another variant of VFL is when the active party has no features and thus it provides no local model. In this variant the active party plays the role of a central server. We refer to the active party providing no feaures in splitVFL and aggVFL, respectively, as **splitVFL**$_c$ and **aggVFL**$_c$. We illustrate these VFL variants in Figure 4 and summarize their architectural differences in Table 2.

Table 2: Comparison of splitVFL and aggVFL

	splitVFL	aggVFL
Has passive party models $\mathcal{G}_i, i = 1, ..., K$-1?	Yes	Yes
Is global module $\mathcal{F}_K$ trainable?	Yes	No
If active party K has no features	splitVFL$_c$	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 [18].

2.2 VFL Training Protocol

In this section, we describe a general training protocol for VFL, which consists of two steps: 1) Entity Alignment; 2)Privacy-preserving training. See Figure 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 Sec. 5. Whereas conventional VFL frameworks mostly consider entity alignment with exact IDs, recent studies [36] also demonstrated 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 [37], 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 Eq. (1). Then, the active party computes the gradients $\frac{\partial \ell}{\partial \psi_K}$ of its global module and updates its global module 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 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 its local model. 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 Sec. 5.

2.2.1 Tree-based VFL

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

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 [19, 38, 39, 40, 41, 42, 43]. SecureBoost [19], SecureBoost+ [38], and SecureGBM [39] exploit additive homomorphic encryption (HE) to encrypt residual errors and feature histograms transmitted between active and passive parties. SecureXGB [40] and Pivot [44] utilize secret sharing mixed with additive

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 $j = 1, 2, ...$ 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 Equation (3);
        Party $k$ updates $\theta_k^{j+1} = \theta_k^j - \eta_2 \nabla_{\theta_k} \ell$;
    end for
end for
```

HE to encrypt transmitted information. FederBoost [41] and OpBoost [42] adopt differential privacy to protect individual data trying to achieve a better balance between privacy and efficiency.

Random Forest [45] (RF) is another popular tree-based ensemble algorithm that has been integrated into VFL. RF-based VFL algorithms [46, 47, 48] typically leverage bagging and optimized parallelism to enhance the training and inference efficiency. Federated Forest [46] introduces a third party and applies RSA encryption to protect data privacy. VFRF [47] adopts randomized iterative affine cipher (RIAC) [49] to encrypt transmitted information. VPRF [48], a verifiable privacy-preserving random forest scheme, is proposed to verify data integrity and preserve data privacy.

3 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 3 and discuss them in this section.

3.1 Multiple Client Updates

One straightforward way to save the communication cost is by allowing participating parties to perform multiple local updates during each iteration. Liu et al. [18] proposed a federated stochastic block coordinate descent algorithm, called FedBCD, that allows each party to conduct multiple client updates before each communication to reduce the number of synchronizations, thereby mitigating the communication overhead. Castiglia et al. [50] proposed a flexible local update strategy for VFL, named Flex-VFL, that allows each party to conduct a different number of local updates constrained by a specified timeout for each communication round. Zhang et al. [51] proposed an adaptive local update strategy for VFL, named AdaVFL, that optimizes the number of local updates for each party in each round by minimizing the total

Table 3: Summary of existing works that aim to improve the efficiency of VFL. In the Model column, the LR denotes logistic regression, NN denotes Neural Network, XGB denotes XGBoost and GBDT denotes gradient boosting decision tree. 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 / aggVFL	LR/NN	$O(1/\sqrt{T})$	Block coordinate descent w/ multiple local updates
	Flex-VFL [50]	splitVFL$_c$	NN	$O(1/\sqrt{T})$	Customized # of local updates constrained by time
	AdaVFL [51]	aggVFL$_c$	NN	$O(1/\sqrt{T})$	Customized # of local updates through optimization
	VIMADMM [52]	splitVFL / aggVFL	NN	-	Alternative direction method of multipliers
	CELU-VFL [53]	splitVFL	NN	$O(\Delta/\sqrt{T})$	Cache-based mechanism for local updates
Asynchronous Coordination	GP-AVFL [54]	aggVFL	LR/NN	-	Asynchronous training with gradient prediction
	AVFL [55]	aggVFL	LR	-	Backup-based straggler-resilient scheme
	T-VFL [56]	splitVFL$_c$	NN	$O(1/\sqrt{T})$	Channel-aware user scheduling poicy
	VAFL [57]	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/ the same data order
	AFAP [58]	aggVFL	LR	$O(e^{-T})$	Tree-structured asynchronous communication (TSAC)
	AsySQN [59]	aggVFL	LR	$O(e^{-T})$	TSAC & quasi-Newton method
	VFB^2 [60]	aggVFL	LR	$O(e^{-T})$	TSAC & bi-level parallel update
	FDSKL [61]	aggVFL	LR	$O(1/T)$	TSAC & random features & doubly stochastic gradient
	FedGBF [62]	aggVFL	GBDT	-	Use RT as the base learner for learning GBDT
	VF^{2}Boost[63]	aggVFL	GBDT	-	Concurrent protocol & customized Paillier HE
One-Shot Communication	FedOnce [64]	splitVFL	NN	-	Unsupervised learning by predicting noise
	AE-VFL [65]	splitVFL	NN	-	Unsupervised learning using autoencoder
	CE-VFL [66]	splitVFL	NN	-	Unsupervised learning using PCA & autoencoder
Compression	AVFL [55]	aggVFL	LR	-	Principle component analysis
	CE-VFL [66]	splitVFL	NN	-	Autoencoder and principle component analysis
	SecureBoost+ [38]	aggVFL	XGB	-	Encode encrypted first-order and second-order gradients into a single message
	eHE-SecureBoost [67]	aggVFL	XGB	-	
	C-VFL [68]	splitVFL	NN	$O(1/\sqrt{T})$	Arbitrary compression scheme
	GP-AVFL+DESC [54]	aggVFL	LR/NN	-	Double-end sparse compression
Sample and Feature Selection	Coreset-VFL [69]	aggVFL$_c$	LR, K-Mean	-	Coreset to select samples
	FedSDG-FS [70]	splitVFL$_c$	NN	-	Stochastic dual-gate to select features
	SFS-VFL [71]	aggVFL$_c$	LR, KNN, SVM, GBT	-	Gini impurity to select features
	LESS-VFL[72]	splitVFL$_c$	NN	-	Group lasso regularization to select features
	FEAST [73]	aggVFL	LR, SVM, XGB, NN	-	Conditional mutual information to select features
	VFLFS [74]	splitVFL	NN	-	Use trainable transformation matrix to select features

training time. Xie et al. [52] proposed an ADMM-based optimization method to implement multiple local updates. Fu et al. [53] proposed CELU-VFL, an efficient VFL training framework that implements multiple local updates using cached statistics. These methods typically require proper choices of training parameters, e.g.learning rate, to improve convergence and exhibit trade-off between computational resources and communication efficiency.

3.2 Asynchronous Coordination

The core idea of asynchronous coordination is that each party can upload and download intermediate training results asynchronously. 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.

Li et al. [54] proposed GP-AVFL that allows parties to update local models asynchronously by leveraging a gradient prediction technique to dynamically adjust local model gradients. Cai et al. [55] proposed AVFL that accelerates VFL training by omitting the updates from the slow parties with poor network conditions. Zhang et al. [56] proposed a truncated VFL algorithm, called T-VFL, to discard parties with channel gains lower than a threshold. Chen et al. [57] proposed a vertical asynchronous federated learning algorithm called VAFL, which utilizes a query-response strategy that decouples the coordination between the server and clients. Hu et al. [27] proposed FDML, allowing each party to update its local model asynchronously but based on the same sequence of randomly sampled training data.

AsySQN [59], VFB^2 [60], and FDSKL [61] all utilize a tree-structured communication scheme [75] to enhance the communication efficiency. AsySQN [59] additionally exploits approximated Hessian information to obtain a better descent direction. VFB^2 [60] supports multiple active parties. FDSKL [61] integrates a non-linear kernel method into vertical federated learning. It leverages the random features to approximate the kernel mapping function aiming to achieve efficient computation parallelism, and adopts doubly stochastic gradients to update the kernel function for scalability. Han et al. [62] employs the random forest (RF) [45] as the base learner for learning GBDT in order to enhance parallelism and save communication rounds. To reduce the long periods of idle time and accelerate the aggregation process under cryptography, VF^{2}Boost[63] adopts a concurrent training protocol to take full advantage of computational resources and leverages a re-ordered accumulation technique and a histogram packing method to accelerate histogram construction and communication.

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.

3.3 One-shot Communication

One-shot Communication alleviates communication overhead by coordinating only once during the entire training procedure. All proposed one-shot communication approaches follow a two-step training procedure: (1) All parties extract latent representations from their original data using unsupervised learning; (2) The active party trains the global model using these latent representations.

Wu et al. [64] proposed FedOnce, in which each party leverages an unsupervised learning method, called NAT (Noise As Targets) [76], to extract latent representations from its local data. Then the active party trains the global model using its local features combined with latent representations passed from passive parties. AE-VFL [65] leverages autoencoder to extract latent representations from each party's local data, while CE-VFL [66] 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 another party. Therefore, the privacy risks for revealing these representations need to be carefully evaluated, e.g., through inversion attacks or information theory studies. Besides, one-shot methods typically involve computationally expensive unsupervised learning of effective representations. Therefore, the trade-off between communication and computation is worth investigating.

3.4 Compression

Compression is a commonly used approach in VFL to alleviate communication overhead by reducing 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 [55] leverages Principle Component Analysis (PCA) to compress transmitted data, while CE-VFL [66] utilizes both PCA and Autoencoders to learn latent representations from raw data. Two follow-up works, SecureBoost+ [38] and eHE-SecureBoost [67], of SecureBoost 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 [68] allows an arbitrary compression scheme to be applied to VFL to enhance communication efficiency and provides theoretical analysis on the impact of compressor parameters. GP-AVFL [54] employs a double-end sparse compression (DESC) technique to save communication traffic volume by squeezing the sparsity in forward outputs of local models and backward gradients transmitted from the active party to passive parties. Adaptive quantization techniques [77, 78, 79] may also be considered in future VFL research.

3.5 Sample and Feature Selection

Another approach to improve communication efficiency is to reduce the amount of data used for training and inference. For example, Coreset-VFL [69] constructs a coreset of samples to alleviate the communication burden, while FedSDG-FS [70], SFS-VFL [71], LESS-VFL[72], FEAST [73] and VFLFS [74] filter out unimportant features to save communication costs.

4 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*. We summarize these works in Table 4 and discuss them in the rest of this section. To better explain these works, we depict a general virtual dataset formed

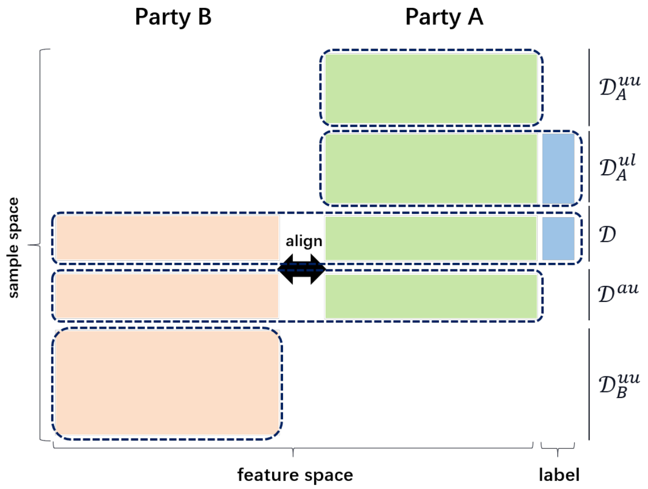


Figure 5: The virtual dataset of a two-party VFL. $\mathcal{D}$ denotes the labeled and aligned samples used by the conventional VFL formulated in Eq. (1), whereas $\mathcal{D}^{au}$ denotes aligned but unlabeled samples. $\mathcal{D}_A^{uu}$ and $\mathcal{D}_B^{uu}$ denote unaligned and unlabeled samples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

by the two parties (see Figure 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, as reported in Table 4. Specifically, $\mathcal{D}$ denotes the labeled and aligned samples, which is used by the conventional VFL formulated in Eq. (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.

Table 4: 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. $\surd$ indicates its corresponding portion of data (see Figure 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 [74]	-	√	√	√	-	Generative Models	≥ 2
		VFed-SSD [80]	√	√	-	-	-	Contrastive Learning	2
		FedHSSL [81]	√	√	√	√	-	Contrastive Learning	≥ 2
		SS-VFNAS [82]	-	√	√	√	-	Contrastive Learning	≥ 2
Semi-SL		FedCVT [83]	-	√	√	-	√	Feature & Label Estimation	2
		FedMC [84]	-	√	√	-	√	Data Collaboration	2
KD	Build a local predictor for active party A	VFedTrans [85]	√	-	-	-	√	FedSVD & Representation Distillation	≥ 2
		VFL-Infer[86]	-	√	-	-	-	Model Distillation	2
		VFed-SSD [80]	√	√	-	-	-	Model Distillation	2
		VFL-JPL [87]	-	√	-	-	√	Feature Estimation & Model Distillation	2
TL	Build a local predictor for passive party B	MMVFL [88]	-	√	-	-	-	Feature Selection & Label Transfer	≥ 2
		SFHTL [89]	-	√	√	-	√	Feature & Label Transfer	≥ 2
		SFTL [90, 91]	√	√	-	-	√	Feature Transfer	2
		PrADA [92]	√	√	-	-	-	Adversarial Domain Adaptation	3

4.1 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 Eq. (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 Eq. (4). We formulate a general Self-SL objective in VFL as follows:

$$\tilde{\psi}_A,\tilde{\theta}_A,\tilde{\theta}_B = \operatorname*{argmin}_{\psi_A,\theta_A,\theta_B} \ell_{\text{Self-SL}}(\psi_A,\theta_A,\theta_B;\mathcal{D}^{au},\mathcal{D}_A^{uu},\mathcal{D}_B^{uu}) \tag{5}$$

where $\ell_{\text{Self-SL}}$ is the self-supervised learning loss that optimizes ψ_A, θ_A and θ_B for learning good representations using unlabeled data. Li et al. [80] proposed VFed-SSD that pretrains local models ψ_A, θ_A and θ_B through Eq. (5) based on positive and negative sample pairs, which are formed from aligned data $\mathcal{D}^{au}$ leveraging matched pair detection (MPD) technique. Then, VFed-SSD finetunes pretrained models $\tilde{\psi}_A$, $\tilde{\theta}_A$ and $\tilde{\theta}_B$ through Eq. (4) based on labeled and aligned samples $\mathcal{D}$. He et al. [81] proposed FedHSSL, a federated hybrid self-supervised learning framework, that pretrains θ_A and θ_B through Eq. (5) based on cross-party views of aligned samples $\mathcal{D}^{au}$ and local views (via data augmentations) 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 Eq. (4) based on $\mathcal{D}$. Feng [74] proposed a VFLFS algorithm that optimizes Eq. (4) and Eq. (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 $\mathcal{D}$.

4.2 Semi-supervised Approaches

Rather than boosting representation learning capability leveraging self-supervised learning, Kang et al. [83] and Yitao et al. [84] proposed 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:

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

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.

Kang et al. [83] proposed a Semi-SL algorithm named FedCVT to implement Eq. (6). 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 trains the VFL model based on $\tilde{\mathcal{D}}$. Yitao et al. [84] proposed FedMC that integrates data collaboration [93] into VFL to implement Eq. (6). FedMC first forms a latent feature space using $\mathcal{D}$. In this latent feature space, it measures the distance between each pair of unaligned samples from the active party and passive party, respectively. 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 expanded training set $\tilde{\mathcal{D}}$. Next, FedMC trains the VFL model based on $\tilde{\mathcal{D}}$.

4.3 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 [85, 86, 80, 87] proposed 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.

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

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 Eq. (4) or Eq. (5).

Wang et al. [85] proposed a vertical federated knowledge transfer approach (VFedTrans) via representation distillation that enables the active party A to make inferences on unaligned local data. To this end, VFedTrans first learns federated representations through FedSVD [94] based on aligned samples $\mathcal{D}^{au}$, and then it utilizes autoencoders as teacher models to transfer the knowledge encoded in the federated representations to the active party A's local models ψ_A^s and θ_A^s as students. Ren et al. [86] proposed VFL-Infer, a VFL framework that pretrains teacher models ψ_A^t, θ_A^t and θ_B^t through Eq. (4), and then leverages these teacher models to help party A train its local models ψ_A^s and θ_A^s through Eq. (7). Li et al. [80] proposed VFed-SSD that trains teacher models through Eq. (5) using cross-party contrastive learning based on aligned data $\mathcal{D}^{au}$ and distills knowledge from teacher models to help the active party A to train its local models ψ_A^s and θ_A^s. In another work along this line of research, Li et al. [87] proposed a joint privileged learning in the VFL setting (VFL-JPL) to train local models for the active party A. By employing the feature imitation and ranking consistency restriction, VFL-JPL can effectively train the active party A's local models through Eq. (7) based on both aligned and unaligned samples as well as knowledge distilled from teacher models pretrained through Eq. (4).

4.4 Transfer Learning-based Approaches

Transfer-learning (TL) based VFL approaches [88, 90, 91, 92, 89] 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:

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

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 Eq. (8).

Liu et al. [90] proposed a secure federated transfer learning framework (SFTL), the pioneering work exploring transfer learning in VFL. SFTL 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 $\mathcal{D}_B$. As a follow-up work, Sharma et al. [91] leverage a more efficient secure computation framework named SPDZ [95] to further enhance the efficiency of SFTL.

SFTL can only transfer knowledge from one source party to one target party. To support multi-party knowledge transfer, Feng et al. [88] proposed a Multi-Participant Multi-Class VFL (MMVFL) that leverages consistency regularization to transfer label information from the active party to all passive parties such that each passive party can learn a local predictor with its pseudo-labeled samples. Feng et al. [89] further proposed a semi-supervised federated heterogeneous transfer learning (SFHTL) that utilizes unaligned samples of all parties and aligned samples to build a local predictor for each party. Specifically, SFHTL 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.

Kang et al. [92] proposed PrADA to address the label deficiency of VFL through domain adaptation (DA). PrADA involves a label-rich source party A, a label-deficient target party B, and a third party that provides rich features for both parties A and B. PrADA treats the third party as a bridge to transfer the knowledge from the source party to the target party and leverages the adversarial domain adaptation to minimize the domain discrepancy between the source and target domains.

5 Preserving Data Privacy and Defending Against Attacks

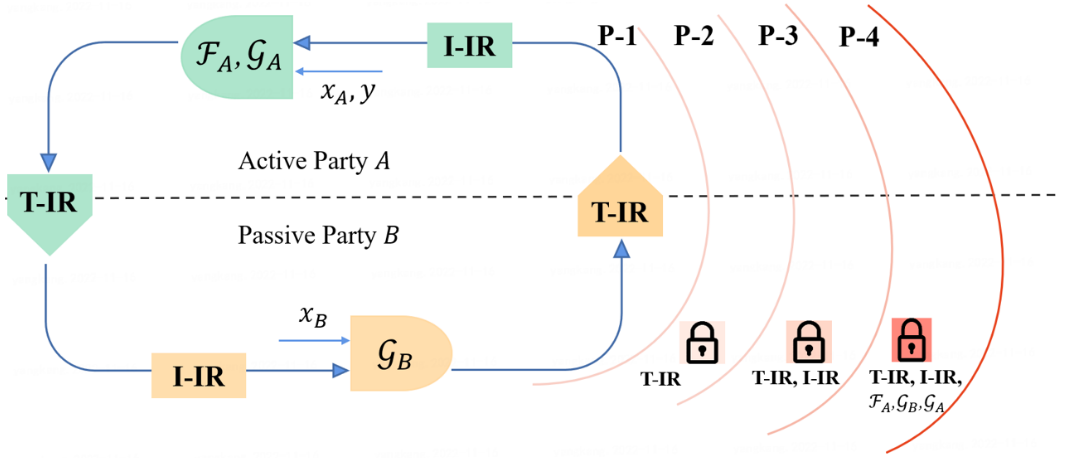
In a VFL system, privacy threats may emerge from the inside or the outside of the system, or both. If the attacker attempts to learn information about the private data of other parties without deviating from the VFL protocol, it is regarded as *honest-but-curious*. The attacker is regarded as *malicious* if it fails to adhere to the VFL protocol. In this section, we first review privacy-preserving protocols involved in the typical VFL framework (Sec. 5.1 and Sec. 5.2), followed by discussions on emerging research on attacks and defense strategies (Sec. 5.3 and Sec. 5.4).

5.1 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 [96] and oblivious transfer [97, 98] etc. The standard PSI protocol is typically applied to a two-party VFL system. [99, 100] 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. [101] proposed an adapted PSI protocol for asymmetrical ID alignment using Pohlig-Hellman encryption scheme and a obfuscate set to help protect the entity information of a weaker party with far less samples than the other party from being exposed. [102] proposed a method called FLORIST that 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.

5.2 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. Figure 6 illustrates these protocols.



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)

Figure 6: A 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.

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 Algo 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. [18] 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.

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:

- Nonprivate Labels. These are cases where labels can be accessed by all parties for training and the security model is to protect features only [27, 68, 50].

- Nonprivate global module or local models. These are cases where the global module [103] or local models [104, 105, 106, 32] are considered *white-boxed* to adversaries.

Since these variants relax the basic security 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 nonprivate label and nonprivate 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.

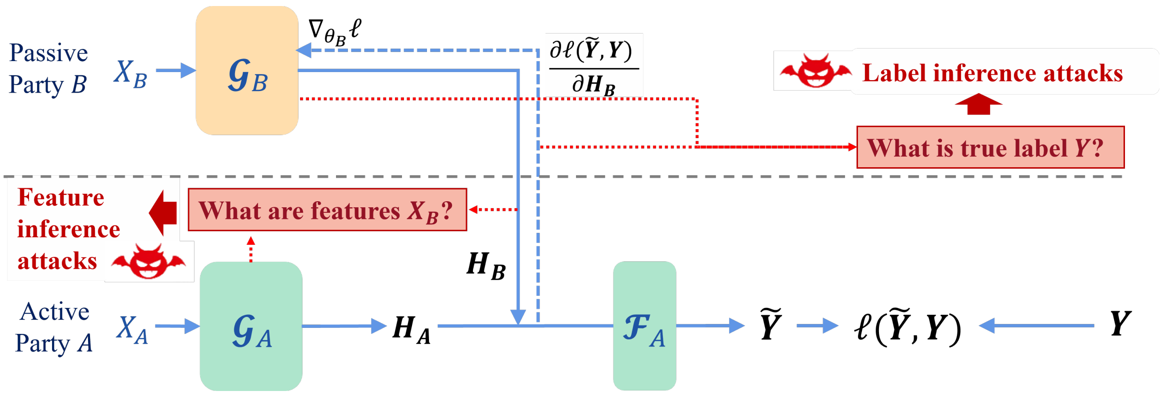


Figure 7: Illustration of data inference attacks in VFL system. The active party A typically infers features or attributes of the passive party B, while the passive party B typically infers labels of the active party A.

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, 107] 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 [19] 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) [23]. 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 privacy-preserving techniques such as SPDZ [91] have also been integrated with VFL [91, 108].

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 [44] and hybrid schemes that combine HE and SS [109, 110]. 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 [28, 19, 44]. However, it requires complex computations which limits its efficiency and scalability.

5.3 Defending against Data Inference Attacks

In a typical VFL system, both features and labels are considered private, whereas most data attacks to HFL scenarios consider features as the target. Therefore, both feature and label protections are critical research subjects for VFL. Figure 7 illustrates data inference attacks in VFL.

5.3.1 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

Table 5: Summary of existing data inference attacks in VFL. A.P. represents the Attacking Phase. In the A.P. column, TRG denotes Training Phase and INF denotes Inference Phase.

	Attacking Method	VFL Setting	Model	Against Protocol	A.P.	Auxiliary Requirement
Label Inference Attack	Direct Label Inference (DLI) [28, 34]	aggVFL	NN	P-1	TRG	–
	Norm Scoring (NS) [111]	splitVFL$_c$	NN	P-1	TRG	–
	Direction Scoring (DS) [111]	splitVFL$_c$	NN	P-1	TRG	–
	Residual Reconstruction (RR) [112]	aggVFL	LR	P-2	TRG	–
	Gradient Inversion (GI) [34]	aggVFL	NN	P-2	TRG	–
	Gradient Inversion with a Label Prior [113]	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) [114]	splitVFL$_c$	NN	P-3	INF	-
	Label-related Relation Inference (LRI) [103]	splitVFL$_c$	GNN	P-0(g)	INF	-
Feature Inference Attack	Binary Feature Inference (BFI) [115]	splitVFL	NN	P-1	TRG	Binary features
	Reverse Multiplication Attack (RMA) [116]	aggVFL	LR	P-2	TRG	Corrupted coordinator
	Protocol-aware Active Attack (PAA) [117]	aggVFL	LR	P-2	TRG	Victim has 1 feature
	Reverse Sum Attack (RSA) [116]	aggVFL	GBDT	P-2	TRG	–
	Equality Solving Attack (ESA) [104]	aggVFL	LR	P-0(g)	INF	–
	Path Restriction Attack (PRA) [104]	aggVFL	Tree	P-0(g)	INF	–
	Generative Regression Network (GRN) [104]	aggVFL	NN	P-0(g)	INF	–
	White-Box Model Inversion (MI) [105, 106]	aggVFL / splitVFL$_c$	LR/NN	P-0(g)	INF	–
	Black-Box Model Inversion (MI) [105, 106]	aggVFL / splitVFL$_c$	LR/NN	P-1	INF	Labeled data
	Catastrophic Data Leakage in VFL (CAFE) [32]	aggVFL$_c$	NN	P-0(g)	TRA	–
	Infer Attribute from Representation (IAR) [118]	aggVFL$_c$ & splitVFL$_c$	NN	P-0(g)	INF	Attribute data

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 5.

Label inference attacks using sample-level gradient. 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) [111, 28]. DLI can achieve accuracy up to 100% if the active party adopts a *nontrainable* 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 [111]. 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 [111] 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 [107]). 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 through the gradient inversion attack (GI) [34, 113] or the residue reconstruction attack (RR) [112] using only the local model gradient. Following the same philosophy of the deep leakage from gradient method [119], passive party B leverages GI to reconstruct the active party's labels by minimizing the distance between the predicted local model gradients $\nabla_{\theta_B}\hat{\ell}$ and the ground truth ones $\nabla_{\theta_B}\ell$. We formulate a general GI attack for inferring labels as follows:

$$y^* = \arg\min_{\hat{y},\hat{\psi}_A,\hat{H}_A} \ell_{\text{GI}}(\nabla_{\theta_B}\hat{\ell}, \nabla_{\theta_B}\ell) + \lambda R_{\text{GI}}(\hat{y}) \tag{9}$$

where $\nabla_{\theta_B}\hat{\ell} = \nabla_{\theta_B}\mathcal{L}(\mathcal{F}_A(\hat{\psi}_A; \hat{H}_A, H_B), \hat{y})$, in which $\hat{y}$ is the label variable needs to be optimized, while $\hat{\psi}_A$ and $\hat{H}_A$ are active party A's global module parameter and local model output, respectively, that are estimated by party B in order to mount GI attack because party B has no access to them; $\nabla_{\theta_B}\ell = \nabla_{\theta_B}\mathcal{L}(\mathcal{F}_A(\psi_A; H_A, H_B), y)$ denotes the ground truth local model gradients; R_{GI} regularizes the label variable $\hat{y}$ based on the label prior, aiming to enhance the quality of $\hat{y}$ [113].

The RR attack is tailored to linear models and aims to infer the plaintext value of encrypted gradient $[[\frac{\partial \ell}{\partial H_B}]]$ by solving an optimization problem as follows [120]:

$$\xi^* = \underset{\hat{\xi}}{\text{argmin}} ||x_B^{\text{T}} \cdot \hat{\xi} - \nabla_{\theta_B}\ell||_2^2 \tag{10}$$

where $\hat{\xi}$ is the variable representing the plaintext value of $[[\frac{\partial \ell}{\partial H_B}]]$ and ξ^* is the reconstructed values of $\frac{\partial \ell}{\partial H_B}$, based on which DLI, NS or DS can be applied to infer labels.

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 [23, 110]. 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) [28], in which the passive party is semi-honest. An *active* version of model completion (AMC) is also proposed in [28]. It leverages a malicious local optimizer instead of normal ones (e.g., Adam) to trick the trained federated model into relying more on the local model of the attacker than other parties such that the attacker can obtain a local model with better performance. MC relies heavily on the adequateness of the auxiliary data owned by the passive party as an attacker. Sun et al. [114] 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. [103] 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. LRI first recovers the active party's local outputs using an optimization-based method. It then recovers relations by forming an adjacency matrix based on outputs from the attacker's and the active party's local models and prediction results.

5.3.2 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 share. Various attacking methods has been proposed to infer features from shallow models (e.g., logistic regression and decision trees) [116, 104, 115] and complex models (e.g., neural networks and random forests) [104, 105, 106, 32]. We summarize existing feature inference attacks in Table 5. 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 attackers in proposed feature inference algorithms 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 [105, 106] during the inference phase and gradient inversion during the training phase [32].

The core idea of model inversion (MI) is to optimize variable $\hat{x}_B$ to approximate the passive party's real input data x_B such that the predicted output $\hat{v}$ of the VLF model is close enough to the real output v computed based on x_B. We formulate a general MI attack as follows:

$$x_B^* = \arg\min_{\hat{x}_B} \ell_{\mathrm{MI}}(\hat{v}, v) + \alpha R_{\mathrm{MI}}(\hat{x}_B) \tag{11}$$

where $\mathcal{L}_{\mathrm{MI}}$ is the loss function that minimizes the distance between v and $\hat{v}$ to optimize $\hat{x}_B$, and R_{MI} regularizes the variable $\hat{x}_B$ based on a prior knowledge. $\hat{v}$ is computed by:

$$\hat{v} = \mathcal{F}_A\left(\mathcal{G}_B(\hat{x}_B, \theta_B), \mathcal{G}_A(x_A, \theta_A), y_A\right) \tag{12}$$

where x_A and y_A are features and labels belonging to the active party; local models $\mathcal{G}_A$ and $\mathcal{G}_B$ can be linear models, tree models or neural network models, and their model parameters θ_A and θ_B are fixed during the optimization; $\mathcal{F}_A$ is the global module that aggregates the outputs of local models and generates $\hat{v}$.

He et al. [105] and Jiang et al. [106] proposed similar white-box model inversion attacks under the SplitNN and aggVFL settings, respectively. Luo et al.[104] proposed three white-box feature inference attacks to learn $\hat{x}_B$ for three different models. These attacks can be seen as specialized MI. More specifically, they designed an Equality Solving Attack (ESA) for the logistic regression, a Path Restriction Attack (PRA) for the decision tree, and a Generative Regression Network (GRN) for attacking the neural network and random forest. The three attacks generally follow the optimization problem defined in Eq. (11).

The gradient inversion (GI) attack was initially proposed in [119] under the HFL setting. The CAFE [32] extended GI to a white-box VFL setting, where the attacker has access to the passive party's model parameters and gradients as well as the aligned data indices. With this knowledge, CAFE can achieve 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. [115] 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 [126] to boost the attack efficiency. Weng et al. [116] and Hu et al. [117] proposed a Reverse Multiplication Attack (RMA) and a Protocol-aware Active Attack (PAA), respectively, to infer the private features x_B of the passive party B in the vertical logistic regression setting that applies P-2 protocol. In RMA, the attacker infers features x_B of the passive party B by solving linear equations in which x_B is the only unknown variable, assuming the coordinator helps decrypt ciphertexts. In PAA, the attacker first obtains the passive party B's outputs through solving a linear system and then utilizes these outputs to infer features of the passive party B. Weng et al. [116] also proposed a Reserve Sum Attack (RSA) targeting SecureBoost. RSA aims to infer the partial order of the passive party's input features by encoding magic numbers into the least significant bits of the encrypted first and second-order

Table 6: Summary of existing cryptographic defense strategies in VFL. In the Defense Scheme column, GC denotes Garbled Circuits, SS denotes Secret Sharing, HE denotes Homomorphic Encryption, FE denotes Functional Encryption and TEE denotes Trusted Execution Environment. In the Adversarial Assumption column, SH denotes Semi-Honest and MA denotes Malicious. In the Protocol column, we assign each defense with Protocols (see Sec. 5.2) it satisfies; "a" and "p" denote active and passive parties, respectively.

Defense Work	VFL Setting	Model	Defense Scheme	Protocol	Party	Require Coordinator	Adversarial Assumption
GasconLR [21]	aggVFL	LR	GC+SS	P-3	≥ 2	✓	SH
HardyLR [107]	aggVFL	LR	HE	P-2	≥ 2	✓	SH
BaiduLR [121]	aggVFL	LR	HE	P-2	≥ 2	✗	SH
SecureLR [122]	aggVFL	LR	HE+SS	P-2	≥ 2	✗	SH
CAESAR [23]	aggVFL	LR	HE+SS	P-3	$= 2$	✗	SH
HeteroLR [110]	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+ [38]	aggVFL	XGB	HE	P-2	≥ 2	✗	SH
SecureXGB [40]	aggVFL	XGB	HE+SS	P-3	$= 2$	✗	SH
MP-FedXGB [43]	aggVFL	XGB	SS	P-3	≥ 2	✓	SH
SecureGBM [39]	aggVFL	LGBM	HE	P-2	$= 2$	✗	SH
Pivot [44]	aggVFL	RF / GBDT	HE+SS	P-3	≥ 2	✗	SH, $\leq K$-1 colluded parties
Enhanced Pivot [44]	aggVFL	DT	HE+SS	P-4	≥ 2	✗	SH, $\leq K$-1 colluded parties
FedSGC [123]	aggVFL$_c$	GNN	HE	P-2	$= 2$	✗	SH
ACML [124]	splitVFL$_c$	NN	HE	P-1	$= 2$	✗	SH
PrADA [92]	splitVFL	NN	HE	P-1	≥ 2	✗	SH
BlindFL [109]	splitVFL	NN	HE+SS	a :P-2, p :P-4	$= 2$	✗	SH
SFTL [90]	aggVFL	NN	HE	P-2	$= 2$	✗	SH
SFTL [90]	aggVFL	NN	SS	P-3	$= 2$	✗	SH
SEFTL [91]	aggVFL	NN	HE+SPDZ	P-3	$= 2$	✗	MA,dishonest majority
N-TEE [125]	aggVFL	XGB	TEE	P-3	≥ 2	✗	SH

gradients. He et al. [105] proposed a black-box model inversion (MI) attack to learn x_B^* under the splitNN setting. More specifically, the attacker first trains a shadow model $\hat{\mathcal{G}}_B$ that mimics the behavior of the local model $\mathcal{G}_B$ using some auxiliary data, and then the attacker learns x_B^* according to Eq. (11) and Eq. (12) with $\hat{\mathcal{G}}_B$ in place of $\mathcal{G}_B$. Jiang et al. [106] proposed a similar 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 [118].

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

5.3.3 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 Sec. 5.2 as a vehicle to compare representative CDS, as listed in Table 6. We consider a defense follow a particular protocol only when it satisfies all requirements of that protocol.

A line of research works [107, 21, 121, 23, 3, 122, 110] focuses on designing CDS to protect the data privacy of vertical linear and logistic regressions. Gascon et al. [21] proposed a hybrid MPC protocol that combines Yao's garbled circuits with tailored protocols for securely solving vertical linear regression (GasconLR). Hardy et al. [107] proposed a HE-based scheme for training the vertical logistic regression (HardyLR). Follow-up works BaiduLR [121] and SecureLR [122] remove the coordinator from the training and inference procedure by relaxing either efficiency or privacy constraint. HardyLR, BaiduLR and SecureLR are vulnerable to privacy attacks targeting batch-level gradients (Sec. 5.3.1). To address this limitation, Chen et al. [23] 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 [110] extends CAESAR further to 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 [90, 91, 109] or are tailored to protect specific intermediate results exposed to the adversary [92, 124] for balancing privacy and efficiency. SFTL [90] designed 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 [91] leverages SPDZ [95] to enhance the efficiency of SFTL further. BlindFL [109] is proposed to build 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 [124] is proposed to build 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 [92] 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 [123] utilizes HE to protect transmitted graph structural information.

For tree-based VFL, SecureBoost [19], SecureBoost+ [38], SecureXGB [40], and MP-FedXGB [43] 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 a SS scheme with distributed optimization to support more-than-two-party scenarios. SecureGBM [39] is a LightGBM-based VFL using additive HE to protect transmitted information. Pivot [44] 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, Chamani et al. [125] introduced a feature inference attack leveraging approximate distribution of feature values and proposed two countermeasures based on Trusted Execution Environment (TEE) to mitigate feature leakage risks.

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 are not feasible, non-cryptographic defense strategies are preferred.

5.3.4 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 [131], gradient sparsification [132, 133] and their hybrid [134]. These methods typically exhibit a trade-off between utility and privacy.

Adding Noise (DP)[119, 28, 111, 135] is a basic defense method for reducing leakage

Table 7: Summary of emerging specialized defense strategies for defending against data leakage attacks (see Table 5).

	Defense Work	VFL Setting	Model	Defense Scheme	Against Attack	Defending Party
Defenses against Label Inference Attack	MARVELL[111]	splitVFL$_c$	NN	Add Noise	NS, DS	Active party
	Max-Norm[111]	splitVFL$_c$	NN	Add Noise	NS, DS	Active party
	CAE [34]	aggVFL	NN	HE+Disguise Label	DLI, MC	Active party
	DCAE [34]	aggVFL	NN	HE+Disguise Label+DG	DLI, MC	Active party
	PELoss [127]	splitVFL$_c$	NN	Potential Energy Loss	MC	Active party
	dCorr [114]	splitVFL$_c$	NN	Minimize Correlation	SA	Active party
	RM [128]	aggVFL	LR	HE+Random Mask	RR	Active party
Defenses against Feature Inference Attack	FG [32]	splitVFL	NN	Random Fake Gradients	CAFE	Passive party
	DRAVL [129]	splitVFL$_c$	NN	Adversarial Training	MI	Passive party
	MD [115]	splitVFL	NN	Masquerade	BFIA	Passive party
	DP-Paillier-MGD [117]	aggVFL	LR	HE+DP	PAA	Passive party
	FedPass [130]	splitVFL	NN	Passport	CAFE, MI	Passive party

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 feature or label leakages [28, 82]. [136] introduced a hybrid differentially private VFL method that adds Gaussian noise to all parties' intermediate results to achieve both local and joint differential privacy. [41, 42] applied differentially private noise to federated gradient-based decision trees in customized ways to achieve a good privacy-utility trade-off. Chen et al. [137] integrate GNN into splitVFL setting and leverage DP-enhanced additive secret sharing to protect data privacy. **Gradient Discretization** (GD)[131] encodes originally continuous gradients into discrete ones, aiming to reduce the private information disclosed to the attacker so that the attacker cannot precisely infer private data through discrete gradients. [34, 28] leveraged a specialized version of GD, named DiscreteSGD, to defend against label inference attacks in VFL. **Gradient Sparsification** (GS)[132] 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. GS are readily applied to distributed learning and HFL scenarios [132, 133]. It is also effective in defending against various label inference attacks for VFL. [34, 28].

A feasible direction to achieve better trade-offs between privacy and utility is designing hybrid defense schemes combining multiple defense strategies [134]. Another direction is to design specialized defense strategies tailored to specific data inference attacks.

5.3.5 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 7.

Defenses against label inference attacks. Li et al. proposed MARVELL [111], which is tailored to defend against Norm Scoring (NS) and Direction Scoring (DS) attacks by adding optimized noise to the sample-level gradients. They also proposed a heuristic Max-Norm defense against the two attacks. Liu et al. [34] proposed label disguising methods, called Confusional AutoEncoder (CAE) and DiscreteSGD-enhanced Confusional AutoEncoder (DCAE), which directly protects label information by encoding the original real label to soft fake labels with maximum confusion. PEloss [127] and dCorr [114] 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. Tan et al. [128] proposed a Random Masking (RM) defense against the Residue Reconstruction attack (RR) by injecting zeros into randomly selected positions of the HE-encrypted sample-level gradients to prevent the RR from reconstructing these gradients correctly. FedPass [130] leverages passport techniques to thwart both label and feature inference attacks.

Defenses against feature inference attacks. Fake Gradients (FG)[32] 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. Sun et al. [129] proposed DRAVL to defend against Model Inversion (MI) through adversarial training. In [115], 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, thereby protecting the true binary features. Hu et al. proposed DP-Paillier-MGD [117] 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 [138, 118] and mutual information regularization [118] were proposed to safeguard sensitive attributes of training samples.

5.4 Defending against Backdoor Attacks

Different from data leakage 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.

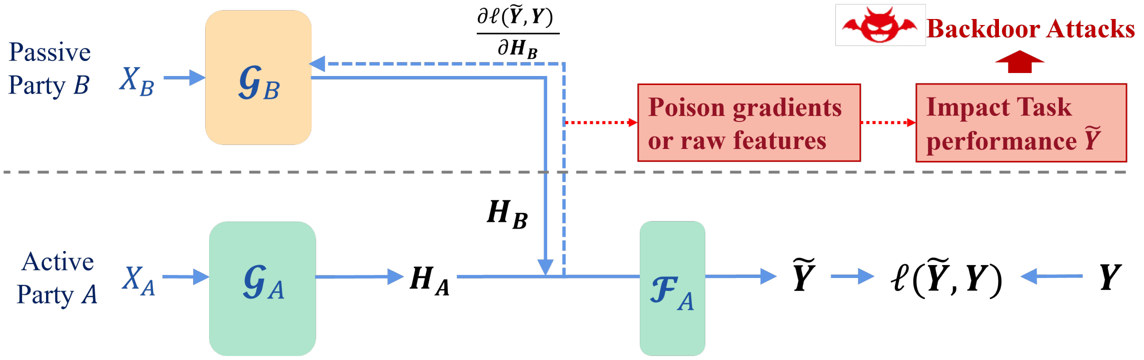


Figure 8: Illustration of backdoor attacks in VFL. Passive parties are the backdoor attackers who aim to impact the task performance of the active party.

In the rest of this section, we summarize existing backdoor attacks and defenses.

5.4.1 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. Figure 8 illustrates backdoor attacks, and Table 8 summarizes the settings and methods for existing backdoor attacks.

Targeted backdoor attacks secretly train a model that achieves high performance on both the original and the targeted backdoor tasks. The objective function of targeted backdoor attacks can be written as follows:

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

Table 8: Summary of existing backdoor attacks in literature.

	Attacking Method	VFL Setting	Against Protocol	Attacking Phase	Auxiliary Requirement
Targeted Backdor Attack	Label Replacement Backdoor (LRB)[139]	aggVFL	P-2	Training	$\geq$ 1 label of clean samples
	Adversarial Dominant Input (ADI)[140]	VLR/splitVFL$_c$	P-0(g)/P-1	Inference	a few samples of other party
Non-targeted Backdoor Attack	Adversarial attack[141, 33]	splitVFL/aggVFL	P-1	Training	–
	Missing attack[33]	splitVFL/aggVFL	P-3	Training	–
	Graph-Fraudster[142]	splitVFL	P-2	Inference	–

Table 9: Summary of defense strategies for defending against backdoor attacks.

Defense Work	VFL Setting	Defense Scheme	Against Attack
DP[34]	aggVFL	Add Noise	Targeted
GS [34]	aggVFL	Sparsify Gradient	Targeted
CAE [34]	aggVFL	HE+Disguise Label	Targeted
DCAE [34]	aggVFL	HE+Disguise Label+DG	Targeted
RVFR [33]	splitVFL	Robust Feature Sub-space Recovery	Targeted/Non-targeted

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. [139] 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 to achieve a high backdoor accuracy while keeping the main task accuracy at a high level. Pang et al. [140] introduced the Adversarial Dominating Input (ADI), which is an input sample with features that override all other features and lead to a certain model output, and proposed gradient-based methods in both white-boxed and black-boxed settings.

Non-targeted backdoor attacks, similar to Byzantine attacks [143] that are typically studied in HFL, aim to hurt the convergence or the performance of the original task by using adversarial samples [141, 33], noisy samples or missing features[33]. An adversarial sample is generated using the Fast Gradient Sign Method (FGSM), 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 [141]. Multiple research works [141, 33] demonstrate the effectiveness of this kind of attack in its misleading performance. If Δx_i is simply a randomly generated perturbation, then the attack is referred to as the noisy-sample attack.

The missing-feature attack simulates real-world VFL scenarios with unstable network[33] in which, for example, the local model output of a passive party may failed to reach the active party for collaboration.

5.4.2 Defense Strategies

Traditional defense strategies such as adding noise and GS are effective in defending against targeted and non-targeted backdoor attacks[33, 34]. 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 [139], 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 attack [34]. RVFR [33] 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 9.

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

6 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.

6.1 Data Valuation

Currently, most research works on data valuations for FL framework still focus on HFL scenarios [144, 145, 146, 147], while data valuations on VFL are much less studied. [148, 149] 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. [150] proposed a model-free approach that uses conditional mutual information for Shapley to evaluate the feature importance and data values in VFL. [151] proposed an embedding-based Shapley evaluation method for VFL and applied this method to both asynchronous and synchronous settings. [152] 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 future research direction.

6.2 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 explainability of VFL. For example, [153] proposed an explainable VFL framework using credibility assessment and counterfactual analysis to control data quality and explain counterfactual instances. [154] designed a VFL scheme based on logistic regression with bounded constraints for interpretable scorecards in credit scoring. [92] 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.

6.3 Fairness

Machine learning models trained in a collaborative setting may inherit bias towards certain user groups. Addressing fairness problem in VFL is an emerging research topic. FairVFL [155] is a framework to use adversarial learning to remove bias for the fairness-sensitive features in a privacy-preserving VFL setting. [156] 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.

6.4 Datasets

We list datasets commonly used in current VFL works in Table 10. Most of the datasets used in VFL research are tabular datasets from Finance, Healthcare, and Advertising. This manifests that, on the one hand, VFL has a broad range of applications in the three fields. On the other hand, tabular datasets dominate VFL research for their convenience in forming multi-party scenarios in VFL, indicating that we are short of research datasets of diverse types (e.g., image, text, or video). In addition, only NUSWIDE and Vehicle datasets consist of multi-modal features that can naturally simulate the two-party VFL scenario. Other datasets listed in Table 10 are adopted from existing machine learning research works, and there is no established way for VFL researchers to partition these datasets for VFL research. Therefore, facilitating industrial applications and academic research in the VFL area calls for practical datasets and high-quality benchmarks.

Table 10: Commonly used datasets in VFL. In the Size column, the number represents the total amount of samples of each dataset. For the three graph datasets, the number on the left of / represents the number of nodes, while the number on the right represents the number of edges.

Dataset	Data Type	Size	Description
Income [157]	Tabular	48842	Demographics and income features
Bank [158]	Tabular	41188	Demographics and economic features
Credit Card [159]	Tabular	30000	Demographics and payments
Give Me Some Credit [160]	Tabular	250000	Debt features
MIMIC III [161]	Tabular	42276	Medical records
Breast Cancer [162]	Tabular	569	Breast tumor features
Diabetes [163]	Tabular	400	Patient records
Avazu [164]	Tabular	4M	Click-through data
Criteo [165]	Tabular	4.5M	Click-through data
Vehicle [166]	Tabular	98528	Acoustic and seismic signals
Drive [167]	Tabular	58509	Electric current drive signals
Cover type [168]	Tabular	581012	Digital spatial data
NUSWIDE [169]	Tabular	269648	Image and the associated tags from Flickr
Handwritten [170]	Tabular	2000	Handwritten digit features
Epsilon [171]	Tabular	500000	Synthetic data
BHI [172]	Image	277524	Medical images
CheXpert [173]	Image	65240	Medical images
Modelnet [174]	Image	20000	Multi-views of 3D objects
Cora [175]	Graph	2708/5429	Citation network
Citeseer [175]	Graph	3327/4732	Citation network
PubMed [175]	Graph	19717/44338	Citation network
Yahoo Answers [176]	Text	1.46M	Corpus of questions and answers
News20 [177]	Text	19928	Newsgroup documents

7 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 Figure 9. We termed this framework **VFLow**.

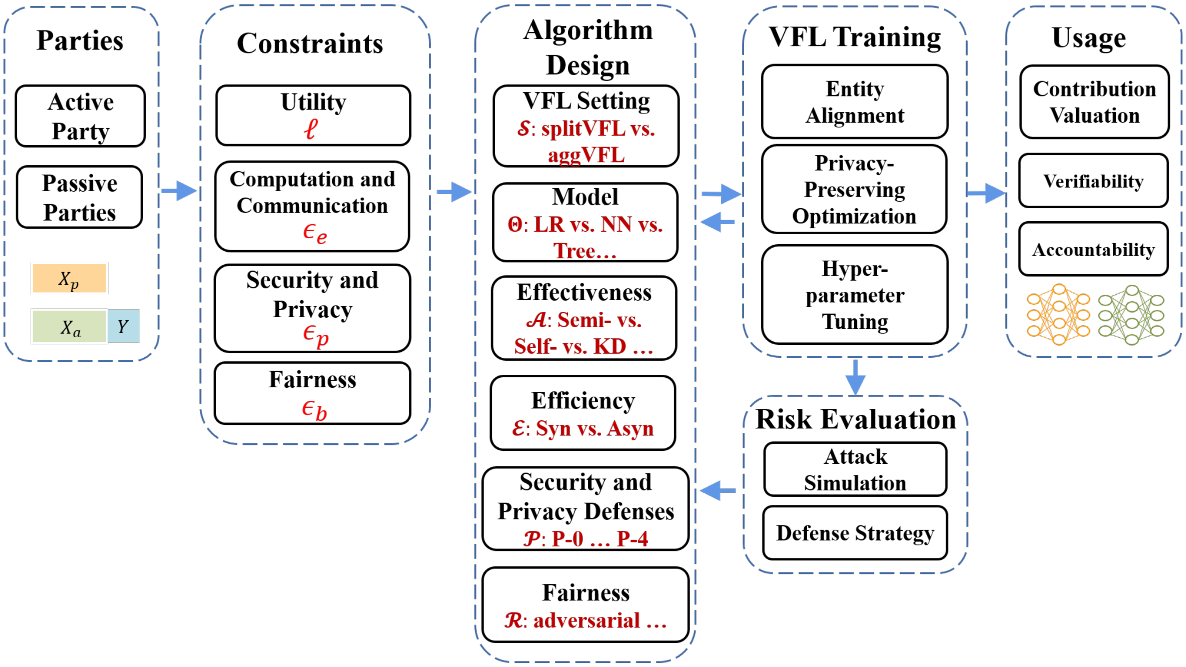


Figure 9: VFLow: A Framework for setting up, designing and optimizing VFL algorithms.

In VFLow, we take into account major constraints, including 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 Sec. 9). We further extend the objective function formulated in Eq. 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:

$$\begin{aligned}&\min_{\Theta} \ell(\Theta; \mathcal{S}, \mathcal{A}, \mathcal{E}, \mathcal{P}, \mathcal{R}, \mathcal{D})\\ &s.t. \quad M_p(\Theta; \mathcal{K}, \mathcal{P}) \leq \epsilon_p, M_e(\Theta; \mathcal{E}, \mathcal{P})) \leq \epsilon_e, M_b(\mathcal{R}, \mathcal{D}) \leq \epsilon_b\end{aligned} \tag{13}$$

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 [178]. 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.

8 Applications

Due to its practical merits for enabling data collaboration between multiple institutions across industries, VFL has attracted increasing attention from both academia and industry. In this section, we provide an overview of VFL applications.

Recommendation systems are typically adopted in VFL to support advertising applications. Federated bandit can be used as a promising technique [179, 180, 181] for FL. Shmueli et al. [182] proposed a privacy-preserving collaborative filtering protocol. Atarashi et al. [183] 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. [184] proposed a secure cross-platform recommendation based on secure computation protocols. Zhang et al.[185] 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.[186] proposed a hybrid federated learning recommendation algorithm named HyFL, which exploits the advantages of both HFL and VFL. Cai et al. [187] proposed a DP-based VFL recommendation framework between a social recommender system and a user social graph.

Many internet companies have adopted VFL to support advertising. For example, ByteDance developed a tree-based VFL algorithm based on the Fedlearner framework, which significantly improves its advertising efficiency [188]. Based on the VFL module in its 9N-FL framework, JD has established a joint model for advertising, which has promoted the cumulative increase of all participating parties' income [189]. Tencent applied its Angel PowerFL platform to establish a VFL federation between advertisers and advertising platforms to boost model accuracy[190]. Based on the trusted intelligent computing service framework (TICS), Huawei applied VFL to advertising[191] 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 [154]. In [23], a secure large-scale sparse logistic regression algorithm is designed and applied to financial risk control. Kang et al. [92] developed a fine-grained adversarial domain adaptation algorithm to address the label deficiency issue in the financial field. Long et al. [192] discussed the applications and open challenges for FL in open banking. Wang et al. [149] 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 [117] and applied to clinical diagnosis. Chen et al.[57] proposed an asynchronous VFL framework and verified the effectiveness of this framework on the public health care dataset MIMIC-III. In [193], 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. [65] proposed an efficient VFL method using autoencoders to predict hearing impairment after surgery based on a vestibular schwannoma dataset. Song et al.[194] 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.[195] proposed a VFL algorithm to locate charging stations for electric vehicles while protecting user privacy. [196] discussed the opportunities for VFL to be utilized in 5G wireless networks. [197] proposed a VFL-based cooperative sensing scheme for cognitive radio networks. [198] developed a VFL framework for optical network disaggregation. [199] applied VFL to collaborative power consumption predictions in smart grid applications. [200] 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. [201] proposed an aimNet that helps the FL model learn better representations from textual and visual features through multi-task learning. Liang et al. [82] 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 [137] and FedVGCN [202] 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 [123] performs node classification on another scenario where one party has only graph structural information while other parties have only node features.

9 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. Thanks to the rapid development of efficient privacy-preserving technologies in recent years, more and more VFL projects and open-sourced platforms have been developed and applied in real-world scenarios, connecting data silos in various industries. However, 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 on this subject, there is still a lack of universally effective defense strategies that are lossless and highly efficient. The trade-off between utility-privacy-efficiency [203] is still the focus of future studies. Applying multi-objective optimization techniques [178] in VFLow is a promising research direction towards trustworthy VFL [204]. 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 [205] in an efficient manner to prevent malicious IP attacks while fulfilling privacy requirements. 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 and Blockchained VFL. Automated machine learning (AutoML) is of great interest in alleviating human effort and achieving satisfactory model performance [206]. Liang et al. [82] proposed a Vertical Federated Neural Architecture Search that learns individual model architecture for each client. [207] discussed challenges in applying NAS to VFL under encryption. For VFL, participants without labels can not perform individual training or evaluation locally. Thus, their hyperparameters are nested in the collaborative training. This unique setting makes AutoML in VFL more challenging. Blockchain is leveraged to address the issue that a vanilla FL framework heavily relies on a central server, which may lead to a single point of failure or privacy vulnerabilities. By utilizing Blockchain, participating parties can exchange their model updates in a decentralized and verifiable manner. How to integrate Blockchain into VFL frameworks to improve the overall security and robustness is an interesting future direction.

10 Concluding Remarks

Vertical federated learning enables collaborative learning of feature-partitioned data distributed across multiple institutions. It has become an attractive solution for solving industrial data silo problems caused by the enforcement of strict data regulations. Despite its practical usefulness, as 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 efforts and new 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 these challenges in this area.

References

[1] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 13, no. 3, pp. 1–207, 2019.

[2] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *CoRR*, vol. abs/1602.05629, 2016. [Online]. Available: http://arxiv.org/abs/1602.05629

[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, jan 2019.

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

[5] C. Ju, D. Gao, R. Mane, B. Tan, Y. Liu, and C. Guan, "Federated transfer learning for eeg signal classification," *IEEE Engineering in Medicine and Biology Society (EMBC)*, 2020.

[6] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konecný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao, "Advances and open problems in federated learning," *CoRR*, vol. abs/1912.04977, 2019. [Online]. Available: http://arxiv.org/abs/1912.04977

[7] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: vision, hype and reality for data privacy and protection," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[8] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106854, 2020.

[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, A. J. Hall, P. Papadopoulos, T. Titcombe, A. Ismail, T. Cebere, R. Sandmann, R. Roehm, and M. A. Hoeh, "Pyvertical: A vertical federated learning framework for multi-headed splitnn," *arXiv preprint arXiv:2104.00489*, 2021.

[11] Bytedance, "Fedlearner:vertical federated gbdt model," https://github.com/bytedance/fedlearner/tree/master/example/tree_model#readme.

[12] C. He, S. Li, J. So, X. Zeng, M. Zhang, H. Wang, X. Wang, P. Vepakomma, A. Singh, H. Qiu *et al.*, "Fedml: A research library and benchmark for federated machine learning," *arXiv preprint arXiv:2007.13518*, 2020.

[13] Q. Li, Y. Cai, Y. Han, C. M. Yung, T. Fu, and B. He, "Fedtree: A fast, effective, and secure tree-based federated learning system," 2022.

[14] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[15] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[16] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[17] 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 *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2022, pp. 1059–1064.

[18] Y. Liu, X. Zhang, Y. Kang, L. Li, T. Chen, M. Hong, and Q. Yang, "Fedbcd: A communication-efficient collaborative learning framework for distributed features," *IEEE Transactions on Signal Processing*, pp. 1–12, 2022.

[19] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, and Q. Yang, "Secureboost: A lossless federated learning framework," *CoRR*, vol. abs/1901.08755, 2019. [Online]. Available: http://arxiv.org/abs/1901.08755

[20] C. Gratton, N. K. Venkategowda, R. Arablouei, and S. Werner, "Distributed ridge regression with feature partitioning," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2018, pp. 1423–1427.

[21] A. Gascón, P. Schoppmann, B. Balle, M. Raykova, J. Doerner, S. Zahur, and D. Evans, "Secure linear regression on vertically partitioned datasets," *IACR Cryptology ePrint Archive*, vol. 2016, p. 892, 2016.

[22] A. P. Sanil, A. F. Karr, X. Lin, and J. P. Reiter, “Privacy preserving regression modelling via distributed computation,” in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’04. New York, NY, USA: Association for Computing Machinery, 2004, p. 677–682.

[23] C. Chen, J. Zhou, L. Wang, X. Wu, W. Fang, J. Tan, L. Wang, A. X. Liu, H. Wang, and C. Hong, “When homomorphic encryption marries secret sharing: Secure large-scale sparse logistic regression and applications in risk control,” in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2652–2662.

[24] H. Yu, J. Vaidya, and X. Jiang, “Privacy-preserving svm classification on vertically partitioned data,” in *Pacific-asia conference on knowledge discovery and data mining*. Springer, 2006, pp. 647–656.

[25] R. Xu, N. Baracaldo, Y. Zhou, A. Anwar, J. Joshi, and H. Ludwig, “Fedv: Privacy-preserving federated learning over vertically partitioned data,” in *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security*, 2021, pp. 181–192.

[26] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, “A secure federated transfer learning framework,” *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 70–82, 2020.

[27] Y. Hu, D. Niu, J. Yang, and S. Zhou, “Fdml: A collaborative machine learning framework for distributed features,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery &amp; Data Mining*, ser. KDD ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2232–2240.

[28] C. Fu, X. Zhang, S. Ji, J. Chen, J. Wu, S. Guo, J. Zhou, A. X. Liu, and T. Wang, “Label inference attacks against vertical federated learning,” in *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA: USENIX Association, Aug. 2022.

[29] Z. Li, T. Wang, and N. Li, “Differentially private vertical federated clustering,” *arXiv preprint arXiv:2208.01700*, 2022.

[30] W. Ou, J. Zeng, Z. Guo, W. Yan, D. Liu, and S. Fuentes, “A homomorphic-encryption-based vertical federated learning scheme for rick management,” *Computer Science and Information Systems*, vol. 17, no. 3, pp. 819–834, 2020.

[31] Z. Dang, B. Gu, and H. Huang, *Large-Scale Kernel Method for Vertical Federated Learning*. Cham: Springer International Publishing, 2020, pp. 66–80.

[32] X. Jin, P.-Y. Chen, C.-Y. Hsu, C.-M. Yu, and T. Chen, “Cafe: Catastrophic data leakage in vertical federated learning,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 994–1006, 2021.

[33] J. Liu, C. Xie, K. Kenthapadi, O. O. Koyejo, and B. Li, “Rvfr: Robust vertical federated learning via feature subspace recovery,” 2021.

[34] T. Zou, Y. Liu, Y. Kang, W. Liu, Y. He, Z. Yi, Q. Yang, and Y.-Q. Zhang, “Defending batch-level label inference and replacement attacks in vertical federated learning,” *IEEE Transactions on Big Data*, 2022.

[35] I. Ceballos, V. Sharma, E. Mugica, A. Singh, A. Roman, P. Vepakomma, and R. Raskar, “Splitnn-driven vertical partitioning,” *CoRR*, vol. abs/2008.04137, 2020. [Online]. Available: https://arxiv.org/abs/2008.04137

[36] Z. Wu, Q. Li, and B. He, "A coupled design of exploiting record similarity for practical vertical federated learning," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 21 087–21 100. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/84b744165a0597360caad96b06e69313-Paper-Conference.pdf

[37] L. Wan, W. K. Ng, S. Han, and V. C. S. Lee, "Privacy-preservation for gradient descent methods," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 775–783.

[38] 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," *arXiv preprint arXiv:2110.10927*, 2021.

[39] Z. Feng, H. Xiong, C. Song, S. Yang, B. Zhao, L. Wang, Z. Chen, S. Yang, L. Liu, and J. Huan, "Securegbm: Secure multi-party gradient boosting," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 1312–1321.

[40] W. Fang, D. Zhao, J. Tan, C. Chen, C. Yu, L. Wang, L. Wang, J. Zhou, and B. Zhang, "Large-scale secure xgb for vertical federated learning," ser. CIKM '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 443–452.

[41] Z. Tian, R. Zhang, X. Hou, J. Liu, and K. Ren, "Federboost: Private federated learning for gbdt," *arXiv preprint arXiv:2011.02796*, 2020.

[42] X. Li, Y. Hu, W. Liu, H. Feng, L. Peng, Y. Hong, K. Ren, and Z. Qin, "Opboost: A vertical federated tree boosting framework based on order-preserving desensitization," *In Proceedings of the 49th International Conference on Very Large Data Bases*, 2022.

[43] 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, sep 2022.

[44] Y. Wu, S. Cai, X. Xiao, G. Chen, and B. C. Ooi, "Privacy preserving vertical federated learning for tree-based models," *Proc. VLDB Endow.*, vol. 13, no. 12, p. 2090–2103, jul 2020.

[45] T. K. Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, 1995, pp. 278–282 vol.1.

[46] Y. Liu, Y. Liu, Z. Liu, Y. Liang, C. Meng, J. Zhang, and Y. Zheng, "Federated forest," *IEEE Transactions on Big Data*, 2020.

[47] H. Yao, J. Wang, P. Dai, L. Bo, and Y. Chen, "An efficient and robust system for vertically federated random forest," *arXiv preprint arXiv:2201.10761*, 2022.

[48] J. Hou, M. Su, A. Fu, and Y. Yu, "Verifiable privacy-preserving scheme based on vertical federated random forest," *IEEE Internet of Things Journal*, 2021.

[49] K. Loyka, H. Zhou, and S. P. Khatri, "A homomorphic encryption scheme based on affine transforms," in *Proceedings of the 2018 on Great Lakes Symposium on VLSI*, ser. GLSVLSI '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 51–56.

[50] T. Castiglia, S. Wang, and S. Patterson, “Flexible vertical federated learning with heterogeneous parties,” *ArXiv*, vol. abs/2208.12672, 2022.

[51] J. Zhang, S. Guo, Z. Qu, D. Zeng, H. Wang, Q. Liu, and A. Y. Zomaya, “Adaptive vertical federated learning on unbalanced features,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 12, pp. 4006–4018, 2022.

[52] C. Xie, P.-Y. Chen, C. Zhang, and B. Li, “Improving Privacy-Preserving Vertical Federated Learning by Efficient Communication with ADMM,” Jul. 2022.

[53] F. Fu, X. Miao, J. Jiang, H. Xue, and B. Cui, “Towards Communication-efficient Vertical Federated Learning Training via Cache-enabled Local Updates,” *arXiv preprint arXiv:2207.14628*, 2022.

[54] M. Li, Y. Chen, Y. Wang, and Y. Pan, “Efficient asynchronous vertical federated learning via gradient prediction and double-end sparse compression,” in *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2020, pp. 291–296.

[55] D. Cai, T. Fan, Y. Kang, L. Fan, M. Xu, S. Wang, and Q. Yang, “Accelerating vertical federated learning,” *IEEE Transactions on Big Data*, no. 01, pp. 1–10, jul 2022.

[56] Z. Zhang, G. Zhu, and S. Cui, “Low-Latency Cooperative Spectrum Sensing via Truncated Vertical Federated Learning,” *arXiv preprint arXiv:2208.03694*, 2022.

[57] T. Chen, X. Jin, Y. Sun, and W. Yin, “Vafl: A method of vertical asynchronous federated learning,” *arXiv preprint arXiv:2007.06081*, 2020.

[58] B. Gu, A. Xu, Z. Huo, C. Deng, and H. Huang, “Privacy-preserving asynchronous vertical federated learning algorithms for multiparty collaborative learning,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2021.

[59] Q. Zhang, B. Gu, C. Deng, S. Gu, L. Bo, J. Pei, and H. Huang, “Asysqn: Faster vertical federated learning algorithms with better computation resource utilization,” in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery &amp; Data Mining*, ser. KDD ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 3917–3927.

[60] Q. Zhang, B. Gu, C. Deng, and H. Huang, “Secure bilevel asynchronous vertical federated learning with backward updating,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, 2021, pp. 10 896–10 904.

[61] B. Gu, Z. Dang, X. Li, and H. Huang, “Federated doubly stochastic kernel learning for vertically partitioned data,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 2483–2493.

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

[63] F. Fu, Y. Shao, L. Yu, J. Jiang, H. Xue, Y. Tao, and B. Cui, “Vf2boost: Very fast vertical federated gradient boosting for cross-enterprise learning,” in *Proceedings of the 2021 International Conference on Management of Data*, 2021, pp. 563–576.

[64] Z. Wu, Q. Li, and B. He, “Practical Vertical Federated Learning with Unsupervised Representation Learning,” *IEEE Transactions on Big Data*, 2022.

[65] D. Cha, M. Sung, Y.-R. Park *et al.*, “Implementing vertical federated learning using autoencoders: Practical application, generalizability, and utility study,” *JMIR medical informatics*, vol. 9, no. 6, 2021.

[66] A. Khan, M. ten Thij, and A. Wilbik, “Communication-Efficient Vertical Federated Learning,” *Algorithms*, vol. 15, no. 8, p. 273, 2022.

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

[68] T. J. Castiglia, A. Das, S. Wang, and S. Patterson, “Compressed-vfl: Communication-efficient learning with vertically partitioned data,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 2738–2766.

[69] L. Huang, Z. Li, J. Sun, and H. Zhao, “Coresets for vertical federated learning: Regularized linear regression and k-means clustering,” in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 29 566–29 581. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/be7b70477c8fca697f14b1dbb1c086d1-Paper-Conference.pdf

[70] A. Li, H. Peng, L. Zhang, J. Huang, Q. Guo, H. Yu, and Y. Liu, “Fedsdg-fs: Efficient and secure feature selection for vertical federated learning,” *IEEE International Conference on Computer Communications*, 2023.

[71] R. Zhang, H. Li, M. Hao, H. Chen, and Y. Zhang, “Secure feature selection for vertical federated learning in ehealth systems,” in *IEEE International Conference on Communications*, 2022.

[72] T. Castiglia, Y. Zhou, S. Wang, S. Kadhe, N. Baracaldo, and S. Patterson, “Less-vfl: Communication-efficient feature selection for vertical federated learning,” *International Conference on Machine Learning*, 2023.

[73] R. Fu, Y. Wu, Q. Xu, and M. Zhang, “Feast: A communication-efficient federated feature selection framework for relational data,” *Proc. ACM Manag. Data*, 2023.

[74] S. Feng, “Vertical federated learning-based feature selection with non-overlapping sample utilization,” *Expert Systems with Applications*, vol. 208, p. 118097, 2022.

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

[76] P. Bojanowski and A. Joulin, “Unsupervised learning by predicting noise,” ser. ICML’17. JMLR.org, 2017, p. 517–526.

[77] Y. Mao, Z. Zhao, G. Yan, Y. Liu, T. Lan, L. Song, and W. Ding, “Communication-efficient federated learning with adaptive quantization,” *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, aug 2022.

[78] D. Jhunjhunwala, A. Gadhikar, G. Joshi, and Y. C. Eldar, “Adaptive quantization of model updates for communication-efficient federated learning,” in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 3110–3114.

[79] G. Yan, T. Li, S.-L. Huang, T. Lan, and L. Song, "Ac-sgd: Adaptively compressed sgd for communication-efficient distributed learning," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 9, pp. 2678–2693, 2022.

[80] W. Li, Q. Xia, J. Deng, H. Cheng, J. Liu, K. Xue, Y. Cheng, and S.-T. Xia, "Semi-supervised cross-silo advertising with partial knowledge transfer," *arXiv preprint arXiv:2205.15987*, 2022.

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

[82] X. Liang, Y. Liu, J. Luo, Y. He, T. Chen, and Q. Yang, "Self-supervised Cross-silo Federated Neural Architecture Search," Feb. 2021.

[83] Y. Kang, Y. Liu, and X. Liang, "FedCVT: Semi-supervised Vertical Federated Learning with Cross-view Training," *ACM Transactions on Intelligent Systems and Technology (TIST)*, May 2022.

[84] Y. Yang, X. Ye, and T. Sakurai, "Multi-view federated learning with data collaboration," ser. ICMLC 2022. New York, NY, USA: Association for Computing Machinery, 2022, p. 178–183.

[85] C. Huang, L. Wang, and X. Han, "Vertical federated knowledge transfer via representation distillation for healthcare collaboration networks," in *Proceedings of the ACM Web Conference 2023*, New York, NY, USA, 2023, p. 4188–4199.

[86] Z. Ren, L. Yang, and K. Chen, "Improving availability of vertical federated learning: Relaxing inference on non-overlapping data," *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2022.

[87] W. Li, Q. Xia, H. Cheng, K. Xue, and S.-T. Xia, "Vertical semi-federated learning for efficient online advertising," *arXiv preprint arXiv:2209.15635*, 2022.

[88] S. Feng and H. Yu, "Multi-participant multi-class vertical federated learning," *arXiv preprint arXiv:2001.11154*, 2020.

[89] S. Feng, B. Li, H. Yu, Y. Liu, and Q. Yang, "Semi-Supervised Federated Heterogeneous Transfer Learning," *Knowledge-Based Systems*, vol. 252, p. 109384, 2022.

[90] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, "A secure federated transfer learning framework," *IEEE Intelligent Systems*, vol. 35, pp. 70–82, 2020.

[91] S. Sharma, C. Xing, Y. Liu, and Y. Kang, "Secure and efficient federated transfer learning," *2019 IEEE International Conference on Big Data (Big Data)*, pp. 2569–2576, 2019.

[92] 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 Transactions on Big Data*, no. 01, pp. 1–12, jul 2022.

[93] A. Imakura and T. Sakurai, "Data collaboration analysis framework using centralization of individual intermediate representations for distributed data sets," *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, vol. 6, no. 2, p. 04020018, 2020.

[94] D. Chai, L. Wang, L. Fu, J. Zhang, K. Chen, and Q. Yang, "Practical lossless federated singular vector decomposition over billion-scale data," 2022.

[95] I. Damgård, V. Pastro, N. Smart, and S. Zakarias, “Multiparty computation from somewhat homomorphic encryption,” in *Annual Cryptology Conference*. Springer, 2012, pp. 643–662.

[96] G. Liang and S. S. Chawathe, “Privacy-preserving inter-database operations,” in *International Conference on Intelligence and Security Informatics*. Springer, 2004, pp. 66–82.

[97] B. Pinkas, T. Schneider, and M. Zohner, “Faster private set intersection based on OT extension,” in *23rd USENIX Security Symposium (USENIX Security 14)*. San Diego, CA: USENIX Association, Aug. 2014, pp. 797–812.

[98] ——, “Scalable private set intersection based on ot extension,” *ACM Transactions on Privacy and Security (TOPS)*, vol. 21, no. 2, pp. 1–35, 2018.

[99] Z. Zhou, Y. Tian, and C. Peng, “Privacy-preserving federated learning framework with general aggregation and multiparty entity matching,” *Wireless Communications and Mobile Computing*, vol. 2021, 2021.

[100] L. Lu and N. Ding, “Multi-party private set intersection in vertical federated learning,” in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2020, pp. 707–714.

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

[102] J. Sun, X. Yang, Y. Yao, A. Zhang, W. Gao, J. Xie, and C. Wang, “Vertical federated learning without revealing intersection membership,” *arXiv preprint arXiv:2106.05508*, 2021.

[103] P. Qiu, X. Zhang, S. Ji, T. Du, Y. Pu, J. Zhou, and T. Wang, “Your labels are selling you out: Relation leaks in vertical federated learning,” *IEEE Transactions on Dependable and Secure Computing*, pp. 1–16, 2022.

[104] X. Luo, Y. Wu, X. Xiao, and B. C. Ooi, “Feature Inference Attack on Model Predictions in Vertical Federated Learning,” in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, Apr. 2021, pp. 181–192.

[105] Z. He, T. Zhang, and R. B. Lee, “Model inversion attacks against collaborative inference,” in *Proceedings of the 35th Annual Computer Security Applications Conference*, 2019, pp. 148–162.

[106] X. Jiang, X. Zhou, and J. Grossklags, “Comprehensive Analysis of Privacy Leakage in Vertical Federated Learning During Prediction.” *Proc. Priv. Enhancing Technol.*, vol. 2022, no. 2, pp. 263–281, 2022.

[107] S. Hardy, W. Henecka, H. Ivey-Law, R. Nock, G. Patrini, G. Smith, and B. Thorne, “Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption,” *CoRR*, vol. abs/1711.10677, 2017.

[108] S. Sharma, C. Xing, and Y. Liu, “Privacy-preserving deep learning with spdz,” in *The AAAI Workshop on Privacy-Preserving Artificial Intelligence*, vol. 4, 2019.

[109] F. Fu, H. Xue, Y. Cheng, Y. Tao, and B. Cui, “Blindfl: Vertical federated machine learning without peeking into your data,” in *Proceedings of the 2022 International Conference on Management of Data*, ser. SIGMOD ’22. New York, NY, USA: Association for Computing Machinery, 2022, p. 1316–1330.

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

[111] O. Li, J. Sun, X. Yang, W. Gao, H. Zhang, J. Xie, V. Smith, and C. Wang, “Label leakage and protection in two-party split learning,” *arXiv preprint arXiv:2102.08504*, 2021.

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

[113] S. Kariyappa and M. K. Qureshi, “Exploit: Extracting private labels in split learning,” 2021. [Online]. Available: https://arxiv.org/abs/2112.01299

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

[115] P. Ye, Z. Jiang, W. Wang, B. Li, and B. Li, “Feature reconstruction attacks and countermeasures of dnn training in vertical federated learning,” 2022. [Online]. Available: https://arxiv.org/abs/2210.06771

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

[117] Y. Hu, T. Cai, J. Shan, S. Tang, C. Cai, E. Song, B. Li, and D. Song, “Is vertical logistic regression privacy-preserving? a comprehensive privacy analysis and beyond,” *arXiv preprint arXiv:2207.09087*, 2022.

[118] C. Song and V. Shmatikov, “Overlearning reveals sensitive attributes,” in *International Conference on Learning Representations*, 2020.

[119] L. Zhu, Z. Liu, and S. Han, “Deep leakage from gradients,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. dÁlché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 14 774–14 784.

[120] Y. Kang, J. Luo, Y. He, X. Zhang, L. Fan, and Q. Yang, “A framework for evaluating privacy-utility trade-off in vertical federated learning,” 2022. [Online]. Available: https://arxiv.org/abs/2209.03885

[121] S. Yang, B. Ren, X. Zhou, and L. Liu, “Parallel distributed logistic regression for vertical federated learning without third-party coordinator,” *IJCAI-19 Workshop on Federated Machine Learning for User Privacy and Data Confidentiality*, 2019.

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

[123] W. D. Tsz-Him Cheung and S. Li, “Fedsgc: Federated simple graph convolution for node classification,” *In IJCAI Workshops,*, 2021.

[124] Y. Zhang and H. Zhu, “Additively homomorphical encryption based deep neural network for asymmetrically collaborative machine learning,” *CoRR*, vol. abs/2007.06849, 2020. [Online]. Available: https://arxiv.org/abs/2007.06849

[125] J. G. Chamani and D. Papadopoulos, "Mitigating leakage in federated learning with trusted hardware," *arXiv preprint arXiv:2011.04948*, 2020.

[126] M. W. Mahoney, "Randomized algorithms for matrices and data," *Found. Trends Mach. Learn.*, vol. 3, no. 2, p. 123–224, feb 2011.

[127] F. Zheng, C. Chen, B. Yao, and X. Zheng, "Making split learning resilient to label leakage by potential energy loss," 2022. [Online]. Available: https://arxiv.org/abs/2210.09617

[128] 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

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

[130] H. Gu, J. Luo, Y. Kang, L. Fan, and Q. Yang, "Fedpass: Privacy-preserving vertical federated deep learning with adaptive obfuscation," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, 2023, pp. 3759–3767.

[131] N. Dryden, T. Moon, S. A. Jacobs, and B. Van Essen, "Communication quantization for data-parallel training of deep neural networks," in *2016 2nd Workshop on Machine Learning in HPC Environments (MLHPC)*. IEEE, 2016, pp. 1–8.

[132] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," *arXiv preprint arXiv:1704.05021*, 2017.

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

[134] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1310–1321.

[135] C. Dwork, "Differential privacy: A survey of results," in *International conference on theory and applications of models of computation*. Springer, 2008, pp. 1–19.

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

[137] C. Chen, J. Zhou, L. Zheng, H. Wu, L. Lyu, J. Wu, B. Wu, Z. Liu, L. Wang, and X. Zheng, "Vertically federated graph neural network for privacy-preserving node classification," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, 7 2022, pp. 1959–1965.

[138] J. Jia and N. Z. Gong, "AttriGuard: A practical defense against attribute inference attacks via adversarial machine learning," in *27th USENIX Security Symposium*, 2018, pp. 513–529.

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

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

[141] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *stat*, vol. 1050, p. 20, 2015.

[142] 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," 2021. [Online]. Available: https://arxiv.org/abs/2110.06468

[143] D. Alistarh, Z. Allen-Zhu, and J. Li, "Byzantine stochastic gradient descent," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[144] H. Yu, Z. Liu, Y. Liu, T. Chen, M. Cong, X. Weng, D. Niyato, and Q. Yang, "A fairness-aware incentive scheme for federated learning," in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 2020, pp. 393–399.

[145] T. Song, Y. Tong, and S. Wei, "Profit allocation for federated learning," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 2577–2586.

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

[147] Z. Liu, Y. Chen, H. Yu, Y. Liu, and L. Cui, "Gtg-shapley: Efficient and accurate participant contribution evaluation in federated learning," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 4, pp. 1–21, 2022.

[148] G. Wang, C. X. Dang, and Z. Zhou, "Measure contribution of participants in federated learning," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 2597–2604.

[149] G. Wang, "Interpret federated learning with shapley values," *CoRR*, vol. abs/1905.04519, 2019. [Online]. Available: http://arxiv.org/abs/1905.04519

[150] X. Han, L. Wang, and J. Wu, "Data valuation for vertical federated learning: An information-theoretic approach," *CoRR*, vol. abs/2112.08364, 2021. [Online]. Available: https://arxiv.org/abs/2112.08364

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

[152] J. Jiang, L. Burkhalter, F. Fu, B. Ding, B. Du, A. Hithnawi, B. Li, and C. Zhang, "Vf-ps: How to select important participants in vertical federated learning, efficiently and securely?" in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 2088–2101. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/0e1a2388cd2f78069f4d048d935cb218-Paper-Conference.pdf

[153] P. Chen, X. Du, Z. Lu, J. Wu, and P. C. Hung, "Evfl: An explainable vertical federated learning for data-oriented artificial intelligence systems," *Journal of Systems Architecture*, vol. 126, p. 102474, 2022.

[154] F. Zheng, K. Li, J. Tian, X. Xiang *et al.*, "A vertical federated learning method for interpretable scorecard and its application in credit scoring," *arXiv preprint arXiv:2009.06218*, 2020.

[155] T. Qi, F. Wu, C. Wu, L. Lyu, T. Xu, Z. Yang, Y. Huang, and X. Xie, “Fairvfl: A fair vertical federated learning framework with contrastive adversarial learning,” *arXiv preprint arXiv:2206.03200*, 2022.

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

[157] R. Kohavi, “Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 202–207.

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

[159] 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 systems with applications*, vol. 36, no. 2, pp. 2473–2480, 2009.

[160] Kaggle, “Give me some credit dataset,” https://www.kaggle.com/c/GiveMeSomeCredit.

[161] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, “Mimic-iii, a freely accessible critical care database,” *Scientific data*, vol. 3, p. 160035, 2016.

[162] W. N. Street, W. H. Wolberg, and O. L. Mangasarian, “Nuclear feature extraction for breast tumor diagnosis,” in *Biomedical image processing and biomedical visualization*, vol. 1905. SPIE, 1993, pp. 861–870.

[163] 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 *Proceedings of the annual symposium on computer application in medical care.* American Medical Informatics Association, 1988, p. 261.

[164] Kaggle, “Avazu dataset,” https://www.kaggle.com/c/avazu-ctr-prediction.

[165] Criteo-Labs., “Criteo dataset,” https://labs.criteo.com/2014/02/download-kaggle-display-advertising-challenge-dataset/.

[166] M. F. Duarte and Y. H. Hu, “Vehicle classification in distributed sensor networks,” *Journal of Parallel and Distributed Computing*, vol. 64, no. 7, pp. 826–838, 2004.

[167] D. Dua and C. Graff, “UCI machine learning repository,” "http://archive.ics.uci.edu/ml", 2017.

[168] J. A. Blackard and D. J. Dean, “Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables,” *Computers and electronics in agriculture*, vol. 24, no. 3, pp. 131–151, 1999.

[169] 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. of ACM Conf. on Image and Video Retrieval (CIVR’09)*, Santorini, Greece., Jul. 2009.

[170] 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.

[171] PASCAL-Challenge-2008, “epsilon dataset,” https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

[172] P. Mooney, “Breast histopathology images,” https://www.kaggle.com/datasets/paultimothymooney/breast-histopathology-images.

[173] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya *et al.*, “Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 590–597.

[174] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, “3D ShapeNets: A deep representation for volumetric shapes,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1912–1920.

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

[176] S. Rakshit, “Yahoo answers dataset,” https://www.kaggle.com/soumikrakshit/yahoo-answers-dataset.

[177] S. S. Keerthi, D. DeCoste, and T. Joachims, “A modified finite newton method for fast solution of large scale linear svms.” *Journal of Machine Learning Research*, vol. 6, no. 3, 2005.

[178] Y. Kang, H. Gu, X. Tang, Y. He, Y. Zhang, J. He, Y. Han, L. Fan, K. Chen, and Q. Yang, “Optimizing privacy, utility and efficiency in constrained multi-objective federated learning,” *CoRR*, vol. abs/2305.00312, 2023.

[179] Z. Liu, L. Song, and C. Fragouli, “Federated multi-armed bandits with vector rewards for aspect-based recommendations,” in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022, pp. 1079–1084.

[180] T. Li and L. Song, “Privacy-preserving communication-efficient federated multi-armed bandits,” *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 3, pp. 773–787, 2022.

[181] Z. Zhu, J. Zhu, J. Liu, and Y. Liu, “Federated bandit: A gossiping approach,” *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 5, no. 1, feb 2021.

[182] E. Shmueli and T. Tassa, “Secure multi-party protocols for item-based collaborative filtering,” in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, ser. RecSys ’17. New York, NY, USA: Association for Computing Machinery, 2017, p. 89–97.

[183] K. Atarashi and M. Ishihata, “Vertical federated learning for higher-order factorization machines,” in *Advances in Knowledge Discovery and Data Mining*, K. Karlapalem, H. Cheng, N. Ramakrishnan, R. K. Agrawal, P. K. Reddy, J. Srivastava, and T. Chakraborty, Eds. Cham: Springer International Publishing, 2021, pp. 346–357.

[184] J. Cui, C. Chen, L. Lyu, C. Yang, and W. Li, “Exploiting data sparsity in secure cross-platform social recommendation,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 10 524–10 534, 2021.

[185] J. Zhang and Y. Jiang, "A vertical federation recommendation method based on clustering and latent factor model," in *2021 International Conference on Electronic Information Engineering and Computer Science (EIECS)*. IEEE, 2021, pp. 362–366.

[186] 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 *2022 IEEE International Conference on Web Services (ICWS)*. IEEE, 2022, pp. 322–330.

[187] J. Cai, Y. Liu, X. Liu, J. Li, and H. Zhuang, "Privacy-preserving federated cross-domain social recommendation," *arXiv preprint arXiv:2206.03200*, 2022.

[188] F. Cai, "Bytedance breaks federated learning: Open source fedlearner framework, 209% increase in advertising efficiency." *Available online: https://www.jiqizhixin.com/articles/2020-11-03-9 (accessed on 15 March 2021)*, 2020.

[189] Y. Hou, "Jd's exploration and practice of large-scale federated learning." *Available online: https://zhuanlan.zhihu.com/p/376697402 (accessed on 31 May 2021)*, 2021.

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

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

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

[193] T. Rooijakkers, "Convinced—enabling privacy-preserving survival analyses using multi-party computation," 2020.

[194] Y. Song, Y. Xie, H. Zhang, Y. Liang, X. Ye, A. Yang, and Y. Ouyang, "Federated learning application on telecommunication-joint healthcare recommendation," in *2021 IEEE 21st International Conference on Communication Technology (ICCT)*. IEEE, 2021, pp. 1443–1448.

[195] Z. Teimoori, A. Yassine, and M. S. Hossain, "A secure cloudlet-based charging station recommendation for electric vehicles empowered by federated learning," *IEEE Transactions on Industrial Informatics*, 2022.

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

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

[198] N. Hashemi, P. Safari, B. Shariati, and J. K. Fischer, "Vertical federated learning for privacy-preserving ml model development in partially disaggregated networks," in *2021 European Conference on Optical Communication (ECOC)*. IEEE, 2021, pp. 1–4.

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

[200] N. Ge, G. Li, L. Zhang, and Y. Liu, "Failure prediction in production line based on federated learning: an empirical study," *Journal of Intelligent Manufacturing*, pp. 1–18, 2021.

[201] F. Liu, X. Wu, S. Ge, W. Fan, and Y. Zou, "Federated learning for vision-and-language grounding problems," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 11 572–11 579.

[202] X. Ni, X. Xu, L. Lyu, C. Meng, and W. Wang, "A vertical federated learning framework for graph convolutional network," *CoRR*, vol. abs/2106.11593, 2021. [Online]. Available: https://arxiv.org/abs/2106.11593

[203] X. Zhang, Y. Kang, K. Chen, L. Fan, and Q. Yang, "Trading off privacy, utility and efficiency in federated learning," *ACM Transactions on Intelligent Systems and Technology*, 2022.

[204] Z. Ren, Y. Kang, L. Fan, L. Yang, T. Fan, Y. Tong, and Q. Yang, "Secureboost hyperparameter tuning via multi-objective federated learning," *International Workshop on Trustworthy Federated Learning in Conjunction with IJCAI*, 2023.

[205] B. Li, L. Fan, H. Gu, J. Li, and Q. Yang, "Fedipr: Ownership verification for federated deep neural network models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[206] Q. Yao, M. Wang, Y. Chen, W. Dai, Y.-F. Li, W.-W. Tu, Q. Yang, and Y. Yu, "Taking Human out of Learning Applications: A Survey on Automated Machine Learning," Oct. 2018.

[207] H. Zhu, H. Zhang, and Y. Jin, "From federated learning to federated neural architecture search: a survey," *Complex & Intelligent Systems*, vol. 7, no. 2, pp. 639–657, 2021.