Secure Bilevel Asynchronous Vertical Federated Learning with Backward Updating

Qingsong Zhang[1,2], **Bin Gu** [3,4], **Cheng Deng**[1,*], **and Heng Huang**[3,5*]
[1]School of Electronic Engineering, Xidian University, Xi'an 710071, China; [2]JD Tech, Beijing 100176, China.
[3]JD Finance America Corporation, Mountain View, CA, USA; [4]MBZUAI, United Arab Emirates
[5]Electrical and Computer Engineering, University of Pittsburgh, PA, USA
{qszhang1995, jsgubin, chdeng.xd, henghuanghh}@gmail.com

Abstract

Vertical federated learning (VFL) attracts increasing attention due to the emerging demands of multi-party collaborative modeling and concerns of privacy leakage. In the real VFL applications, usually only one or partial parties hold labels, which makes it challenging for all parties to collaboratively learn the model without privacy leakage. Meanwhile, most existing VFL algorithms are trapped in the synchronous computations, which leads to inefficiency in their real-world applications. To address these challenging problems, we propose a novel **VF**L framework integrated with new **b**ackward updating mechanism and **b**ilevel asynchronous parallel architecture (VF$\mathbf{B}^2$), under which three new algorithms, including VF$\mathbf{B}^2$-SGD, -SVRG, and -SAGA, are proposed. We derive the theoretical results of the convergence rates of these three algorithms under both strongly convex and nonconvex conditions. We also prove the security of VF$\mathbf{B}^2$ under semi-honest threat models. Extensive experiments on benchmark datasets demonstrate that our algorithms are efficient, scalable and lossless.

Introduction

Federated learning (McMahan et al. 2016; Smith et al. 2017; Kairouz et al. 2019) has emerged as a paradigm for collaborative modeling with privacy-preserving. A line of recent works (McMahan et al. 2016; Smith et al. 2017) focus on the horizontal federated learning, where each party has a subset of samples with complete features. There are also some works (Gascón et al. 2016; Yang et al. 2019b; Dang et al. 2020) studying the vertical federated learning (VFL), where each party holds a disjoint subset of features for all samples. In this paper, we focus on VFL that has attracted much attention due to its wide applications to emerging multi-party collaborative modeling with privacy-preserving.

Currently, there are two mainstream methods for VFL, including homomorphic encryption (HE) based methods and exchanging the raw computational results (ERCR) based methods. The HE based methods (Hardy et al. 2017; Cheng et al. 2019) leverage HE techniques to encrypt the raw data and then use the encrypted data (ciphertext) for training model with privacy-preserving. However, there are two major drawbacks of HE based methods. First, the complexity of homomorphic mathematical operation on ciphertext field is very high, thus HE is extremely time consuming for modeling (Liu, Ng, and Zhang 2015; Liu et al. 2019). Second, approximation is required for HE to support operations of non-linear functions, such as Sigmoid and Logarithmic functions, which inevitably causes loss of the accuracy for various machine learning models using non-linear functions (Kim et al. 2018; Yang et al. 2019a). Thus, the inefficiency and inaccuracy of HE based methods dramatically limit their wide applications to realistic VFL tasks.

ERCR based methods (Zhang et al. 2018; Hu et al. 2019; Gu et al. 2020b) leverage labels and the raw intermediate computational results transmitted from the other parties to compute stochastic gradients, and thus use distributed stochastic gradient descent (SGD) methods to train VFL models efficiently. Although ERCR based methods circumvent aforementioned drawbacks of HE based methods, existing ERCR based methods are designed with only considering that all parties have labels, which is not usually the case in real-world VFL tasks. In realistic VFL applications, usually only one or partial parties (denoted as active parties) have the labels, and the other parties (denoted as passive parties) can only provide extra feature data but do not have labels. When these ERCR based methods are applied to the real situation with both active and passive parties, **the algorithms even cannot guarantee the convergence** because only active parties can update the gradient of loss function based on labels but the passive parties cannot, *i.e.* partial model parameters are not optimized during the training process. Thus, it comes to the crux of designing the proper algorithm for solving real-world VFL tasks with only one or partial parties holding labels.

Moreover, algorithms using synchronous computation (Gong, Fang, and Guo 2016; Zhang et al. 2018) are inefficient when applied to real-world VFL tasks, especially, when computational resources in the VFL system are unbalanced. Therefore, it is desired to design the efficient asynchronous algorithms for real-world VFL tasks. Although there have been several works studying asynchronous VFL algorithms (Hu et al. 2019; Gu et al. 2020b), it is still an open problem to design asynchronous algorithms for solving real-world VFL tasks with only one or partial parties holding labels.

In this paper, we address these challenging problems by proposing a novel framework (VF$\mathbf{B}^2$) integrated with the novel backward updating mechanism (BUM) and bilevel

*Corresponding Authors

asynchronous parallel architecture (BAPA). Specifically, the BUM enables all parties, rather than only active parties, to collaboratively update the model securely and also makes the final model lossless; the BAPA is designed for efficiently asynchronous backward updating. Considering the advantages of SGD-type algorithms in optimizing machine learning models, we thus propose three new SGD-type algorithms, *i.e.*, VF$\mathbf{B}^2$-SGD, -SVRG and -SAGA, under that framework. We summarize the contributions of this paper as follows.

- We are the first to propose the novel backward updating mechanism for ERCR based VFL algorithms, which enables all parties, rather than only parties holding labels, to collaboratively learn the model with privacy-preserving and without hampering the accuracy of final model.
- We design a bilevel asynchronous parallel architecture that enables all parties asynchronously update the model through backward updating, which is efficient and scalable.
- We propose three new algorithms for VFL, including VF$\mathbf{B}^2$-SGD, -SVRG, and -SAGA under VF$\mathbf{B}^2$. Moreover, we theoretically prove their convergence rates for both strongly convex and nonconvex problems.

Notations. $\widehat{w}$ denotes the inconsistent read of w. $\bar{w}$ denotes w to compute local stochastic gradient of loss function for collaborators, which maybe stale due to communication delay. $\psi(t)$ is the corresponding party performing the t-th global iteration. Given a finite set S, $|S|$ denotes its cardinality.

Problem Formulation

Given a training set $\{x_i, y_i\}_{i=1}^n$, where $y_i \in \{-1, +1\}$ for binary classification task or $y_i \in \mathbb{R}$ for regression problem and $x_i \in \mathbb{R}^d$, we consider the model in a linear form of $w^\top x$, where $w \in \mathbb{R}^d$ corresponds to the model parameters. For VFL, x_i is vertically distributed among $q \geq 2$ parties, *i.e.*, $x_i = [(x_i)_{\mathcal{G}_1}; \cdots; (x_i)_{\mathcal{G}_q}]$, where $(x_i)_{\mathcal{G}_\ell} \in \mathbb{R}^{d_\ell}$ is stored on the ℓ-th party and $\sum_{\ell=1}^q d_\ell = d$. Similarly, there is $w = [w_{\mathcal{G}_1}; \cdots; w_{\mathcal{G}_q}]$. Particularly, we focus on the following regularized empirical risk minimization problem.

$$\min_{w\in\mathbb{R}^d} f(w) := \frac{1}{n}\sum_{i=1}^{n} \underbrace{\mathcal{L}\left(w^\top x_i, y_i\right) + \lambda \sum_{\ell=1}^{q} g(w_{\mathcal{G}_\ell})}_{f_i(w)}, \quad \text{(P)}$$

where $w^\top x_i = \sum_{\ell=1}^q w_{\mathcal{G}_\ell}^\top (x_i)_{\mathcal{G}_\ell}$, $\mathcal{L}$ denotes the loss function, $\sum_{\ell=1}^q g(w_{\mathcal{G}_\ell})$ is the regularization term, and $f_i : \mathbb{R}^d \to \mathbb{R}$ is smooth and possibly nonconvex. Examples of problem P include models for binary classification tasks (Conroy and Sajda 2012; Wang et al. 2017) and models for regression tasks (Shen et al. 2013; Wang et al. 2019).

In this paper, we introduce two types of parties: **active party** and **passive party**, where the former denotes data provider holding labels while the latter does not. Particularly, in our problem setting, there are m ($1 \leq m \leq q$) active parties. Each active party can play the role of dominator in model updating by actively launching updates. All parties, including both active and passive parties, passively launching updates play the role of collaborator. To guarantee the model security, only active parties know the form of the loss function. Moreover, we assume that the labels can be shared by all parties finally. Note that this does not obey our intention that only active parties hold the labels before training. The problem studied in this paper is stated as follows:

Given: Vertically partitioned data $\{x_{\mathcal{G}_\ell}\}_{\ell=1}^q$ stored in q parties and the labels only held by active parties.

Learn: A machine learning model **M** collaboratively learned by both active and passive parties without leaking privacy.

Lossless Constraint: The accuracy of **M** must be comparable to that of model $\mathbf{M}'$ learned under non-federated learning.

Algorithm 1 Safe algorithm of obtaining $w^T x_i$.

Input: $\{w_{\mathcal{G}_{\ell'}}\}_{\ell'=1}^q$ and $\{(x_i)_{\mathcal{G}_{\ell'}}\}_{\ell'=1}^q$ allocating at each party, index i.

Do this in parallel

1: **for** $\ell' = 1, \cdots, q$ **do**
2: Generate a ramdon number $\delta_{\ell'}$ and calculate $w_{\mathcal{G}_{\ell'}}^\top (x_i)_{\mathcal{G}_{\ell'}} + \delta_{\ell'}$,
3: **end for**
4: Obtain $\xi_1 = \sum_{\ell'=1}^q (w_{\mathcal{G}_{\ell'}}^\top (x_i)_{\mathcal{G}_{\ell'}} + \delta_{\ell'})$ through tree structure T_1.
5: Obtain $\xi_2 = \sum_{\ell'=1}^q \delta_{\ell'}$ through totally different tree structure $T_2 \neq T_1$.

Output: $w^\top x_i = \xi_1 - \xi_2$

VFB^2 Framework

In this section, we propose the novel VFB^2 framework. VFB^2 is composed of three components and its systemic structure is illustrated in Fig. 1a. The details of these components are presented in the following.

The key of designing the proper algorithm for solving real-world VFL tasks with both active and passive parties is to make the passive parties utilize the label information for model training. However, it is challenging to achieve this because direct using the labels hold by active parties leads to privacy leakage of the labels without training. To address this challenging problem, we design the BUM with painstaking.

Backward Updating Mechanism: The key idea of BUM is to make passive parties indirectly use labels to compute stochastic gradient without directly accessing the raw label data. Specifically, the BUM embeds label y_i into an intermediate value $\vartheta := \frac{\partial \mathcal{L}(w^\top x_i, y_i)}{\partial (w^\top x_i)}$. Then ϑ and i are distributed backward to the other parties. Consequently, the passive parties can also compute the stochastic gradient and update the model by using the received ϑ and i (please refer to Algorithms 2 and 3 for details). Fig. 1b depicts the case where ϑ is distributed from party 1 to the rest parties. In this case, all parties, rather than only active parties, can collaboratively learn the model without privacy leakage.

For VFL algorithms with BUM, dominated updates in different active parties are performed in distributed-memory parallel, while collaborative updates within a party are performed in shared-memory parallel. The difference of parallelism fashion leads to the challenge of developing a new

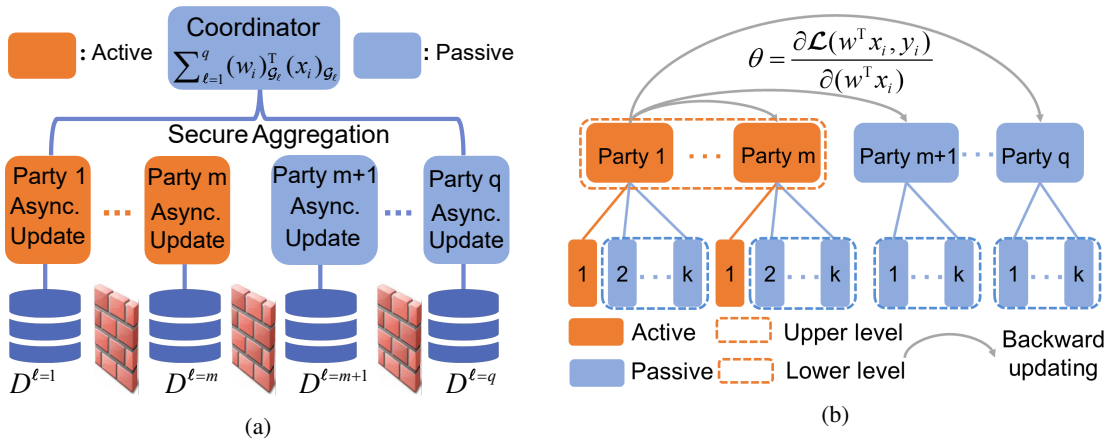


Figure 1: (a): System structure of VF$\mathbf{B}^2$ framework. (b): Illustration of the BUM and BAPA, where k is the number of threads.

parallel architecture instead of just directly adopting the existing asynchronous parallel architecture for VFL. To tackle this challenge, we elaborately design a novel BAPA.

Bilevel Asynchronous Parallel Architecture: The BAPA includes two levels of parallel architectures, where the upper level denotes the inner-party parallel and the lower one is the intra-party parallel. More specifically, the inner-party parallel denotes distributed-memory parallel between active parties, which enables all active parties to asynchronously launch dominated updates; while the intra-party one denotes the shared-memory parallel of collaborative updates within each party, which enables multiple threads within a specific party to asynchronously perform the collaborative updates. Fig. 1b illustrates the BAPA with m active parties.

To utilize feature data provided by other parties, a party need obtain $w^T x_i = \sum_{\ell=1}^{q} w_{\mathcal{G}_\ell}^{\top}(x_i)_{\mathcal{G}_\ell}$. Many recent works achieved this by aggregating the local intermediate computational results securely (Hu et al. 2019; Gu et al. 2020a). In this paper, we use the efficient tree-structured communication scheme (Zhang et al. 2018) for secure aggregation, whose security was proved in (Gu et al. 2020a).

Secure Aggregation Strategy: The details are summarized in Algorithm 1. Specifically, at step 2, $w_{\mathcal{G}_\ell}^{\top}(x_i)_{\mathcal{G}_\ell}$ is computed locally on the ℓ-th party to prevent the direct leakage of $w_{\mathcal{G}_\ell}$ and $(x_i)_{\mathcal{G}_\ell}$. Especially, a random number δ_ℓ is added to $w_{\mathcal{G}_\ell}^{\top}(x_i)_{\mathcal{G}_\ell}$ to mask the value of $w_{\mathcal{G}_\ell}^{\top}(x_i)_{\mathcal{G}_\ell}$, which can enhance the security during aggregation process. At steps 4 and 5, ξ_1 and ξ_2 are aggregated through tree structures T_1 and T_2, respectively. Note that T_2 is totally different from T_1 that can prevent the random value being removed under threat model 1 (defined in section). Finally, value of $w^{\top}x_i = \sum_{\ell=1}^{q}(w_{\mathcal{G}_\ell}^{\top}(x_i)_{\mathcal{G}_\ell}$ is recovered by removing term $\sum_{\ell=1}^{q}\delta_\ell$ from $\sum_{\ell=1}^{q}(w_{\mathcal{G}_\ell}^{\top}(x_i)_{\mathcal{G}_\ell} + \delta_\ell)$ at the output step. Using such aggregation strategy, $(x_i)_{\mathcal{G}_\ell}$ and $w_{\mathcal{G}_\ell}$ are prevented from leaking during the aggregation.

Secure Bilevel Asynchronous VFL Algorithms with Backward Updating

SGD (Bottou 2010) is a popular method for learning machine learning (ML) models. However, it has a poor convergence rate due to the intrinsic variance of stochastic gradient. Thus, many popular variance reduction techniques have been proposed, including the SVRG, SAGA, SPIDER (Johnson and Zhang 2013; Defazio, Bach, and Lacoste-Julien 2014; Wang et al. 2019) and their applications to other problems (Huang, Chen, and Huang 2019; Huang et al. 2020; Zhang et al. 2020; Dang et al. 2020; Yang et al. 2020a,b; Li et al. 2020; Wei et al. 2019). In this section we raise three SGD-type algorithms, *i.e.* the SGD, SVRG and SAGA, which are the most popular ones among SGD-type methods for the appealing performance in practice. We summarize the detailed steps of VF$\mathbf{B}^2$-SGD in Algorithms 2 and 3. For VF$\mathbf{B}^2$-SVRG and -SAGA, one just needs to replace the update rule with corresponding one.

As shown in Algorithm 2, at each dominated update, the dominator (an active party) calculates ϑ and then distributes ϑ together with i to the collaborators (the rest $q-1$ parties). As shown in Algorithm 3, for party ℓ, once it has received the ϑ and i, it will launch a new collaborative update asynchronously. As for the dominator, it computes the local

Algorithm 2 VF$\mathbf{B}^2$-SGD for active party ℓ to actively launch dominated updates.

Input: Local data $\{(x_i)_{\mathcal{G}_\ell}, y_i\}_{i=1}^{n}$ stored on the ℓ-th party, learning rate γ.

1: Initialize the necessary parameters.
Keep doing in parallel (distributed-memory parallel for multiple active parties)
2: Pick up an index i randomly from $\{1, ..., n\}$.
3: Compute $\widehat{w}^{\top}x_i = \sum_{\ell'=1}^{q}\widehat{w}_{\mathcal{G}_{\ell'}}^{\top}(x_i)_{\mathcal{G}_{\ell'}}$ based on Algorithm 1.
4: Compute $\vartheta = \frac{\partial \mathcal{L}(\widehat{w}^{\top}x_i, y_i)}{\partial(\widehat{w}^{\top}x_i)}$.
5: Send ϑ and index i to collaborators.
6: Compute $\widetilde{v}^{\ell} = \nabla_{\mathcal{G}_\ell} f_i(\widehat{w})$.
7: Update $w_{\mathcal{G}_\ell} \leftarrow w_{\mathcal{G}_\ell} - \gamma\widetilde{v}^{\ell}$.
End parallel

Algorithm 3 VF$\mathbf{B}^2$-SGD for the ℓ-th party to passively launch collaborative updates.

Input: Local data $\{(x_i)_{\mathcal{G}_\ell}, y_i\}_{i=1}^n$ stored on the ℓ-th party, learning rate γ.

1: Initialize the necessary parameters (for passive parties).
Keep doing in parallel (shared-memory parallel for multiple threads)
2: Receive ϑ and the index i from the dominator.
3: Compute $\widetilde{v}^\ell = \nabla_{\mathcal{G}_\ell}\mathcal{L}(\bar{w}) + \lambda\nabla_{\mathcal{G}_\ell} g(\widehat{w}) = \vartheta \cdot (x_i)_{\mathcal{G}_\ell} + \lambda\nabla g(\widehat{w}_{\mathcal{G}_\ell})$.
4: Update $w_{\mathcal{G}_\ell} \leftarrow w_{\mathcal{G}_\ell} - \gamma\widetilde{v}^\ell$.
5: **End parallel**

stochastic gradient as $\nabla_{\mathcal{G}_\ell} f_i(\widehat{w}) = \nabla_{\mathcal{G}_\ell}\mathcal{L}(\widehat{w}) + \lambda\nabla g(\widehat{w}_{\mathcal{G}_\ell})$. While, for the collaborator, it uses the received ϑ to compute $\nabla_{\mathcal{G}_\ell}\mathcal{L}$ and local $\widehat{w}$ to compute $\nabla_{\mathcal{G}_\ell} g$ as shown at step 3 in Algorithm 3. Note that active parties also need perform Algorithm 3 to collaborate with other dominators to ensure that the model parameters of all parties are updated.

Theoretical Analysis

In this section, we provide the convergence analyses. Please see the arXiv version for more details. We first present preliminaries for strongly convex and nonconvex problems.

Assumption 1. *For $f_i(w)$ in problem P, we assume the following conditions hold:*

1. **Lipschitz Gradient:** *Each function f_i, $i = 1, \ldots, n$, there exists $L > 0$ such that for $\forall\ w, w' \in \mathbb{R}^d$, there is*

$$\|\nabla f_i(w) - \nabla f_i(w')\| \leq L\|w - w'\|. \tag{1}$$

2. **Block-Coordinate Lipschitz Gradient:** *For $i = 1, \ldots, n$, there exists an $L_\ell > 0$ for the ℓ-th block $\mathcal{G}_\ell$, where $\ell = 1, \cdots, q$ such that*

$$\|\nabla_{\mathcal{G}_\ell} f_i(w + U_\ell\Delta_\ell) - \nabla_{\mathcal{G}_\ell} f_i(w)\| \leq L_\ell\|\Delta_\ell\|, \tag{2}$$

where $\Delta_\ell \in \mathbb{R}^{d_\ell}$, $U_\ell \in \mathbb{R}^{d\times d_\ell}$ and $[U_1, \cdots, U_q] = I_d$.

3. **Bounded Block-Coordinate Gradient:** *There exists a constant G such that for $f_i,\ i = 1, \cdots, n$ and block $\mathcal{G}_\ell$, $\ell = 1, \cdots, q$, it holds that $\|\nabla_{\mathcal{G}_\ell} f_i(w)\|^2 \leq G$.*

Assumption 2. *The regularization term g is L_g-smooth, which means that there exists an $L_g > 0$ for $\ell = 1, \ldots, q$ such that $\forall w_{\mathcal{G}_\ell}, w'_{\mathcal{G}_\ell} \in \mathbb{R}^{d_\ell}$ there is*

$$\|\nabla g(w_{\mathcal{G}_\ell}) - \nabla g(w'_{\mathcal{G}_\ell})\| \leq L_g\|w_{\mathcal{G}_\ell} - w'_{\mathcal{G}_\ell}\|. \tag{3}$$

Assumption 2 imposes the smoothness on g, which is necessary for the convergence analyses. Because, as for a specific collaborator, it uses the received $\widehat{w}$ (denoted as $\bar{w}$) to compute $\nabla_{\mathcal{G}_\ell}\mathcal{L}$ and local $\widehat{w}$ to compute $\nabla_{\mathcal{G}_\ell} g = \nabla g(w_{\mathcal{G}_\ell})$, which makes it necessary to track the behavior of g individually. Similar to previous research works (Lian et al. 2015; Huo and Huang 2017; Leblond, Pedregosa, and Lacoste-Julien 2017), we introduce the bounded delay as follows.

Assumption 3. Bounded Delay: *Time delays of inconsistent reading and communication between dominator and its collaborators are upper bounded by τ_1 and τ_2, respectively.*

Given $\widehat{w}$ as the inconsistent read of w, which is used to compute the stochastic gradient in dominated updates, following the analysis in (Gu et al. 2020b), we have

$$\widehat{w}_t - w_t = \gamma \sum_{u\in D(t)} U_{\psi(u)}\widetilde{v}_u^{\psi(u)}, \tag{4}$$

where $D(t) = \{t-1, \cdots, t-\tau_0\}$ is a subset of non-overlapped previous iterations with $\tau_0 \leq \tau_1$. Given $\bar{w}$ as the parameter used to compute the $\nabla_{\mathcal{G}_\ell}\mathcal{L}$ in collaborative updates, which is the steal state of $\widehat{w}$ due to the communication delay between the specific dominator and its corresponding collaborators. Then, following the analyses in (Huo and Huang 2017), there is

$$\bar{w}_t = \widehat{w}_{t-\tau_0} = \widehat{w}_t + \gamma \sum_{t'\in D'(t)} U_{\psi(t')}\widetilde{v}_{t'}^{\psi(t')}, \tag{5}$$

where $D'(t) = \{t-1, \cdots, t-\tau_0\}$ is a subset of previous iterations performed during the communication and $\tau_0 \leq \tau_2$.

Convergence Analysis for Strongly Convex Problem

Assumption 4. *Each function f_i, $i = 1, \ldots, n$, is μ-strongly convex, i.e., $\forall\ w,\ w' \in \mathbb{R}^d$ there exists a $\mu > 0$ such that*

$$f_i(w) \geq f_i(w') + \langle\nabla f_i(w'), w - w'\rangle + \frac{\mu}{2}\|w - w'\|^2. \tag{6}$$

For strongly convex problem, we introduce notation $K(t)$ that denotes a minimum set of successive iterations fully visiting all coordinates from global iteration number t. Note that this is necessary for the asynchronous convergence analyses of the global model. Moreover, we assume that the size of $K(t)$ is upper bounded by η_1, *i.e.*, $|K(t)| \leq \eta_1$. Based on $K(t)$, we introduce the epoch number $v(t)$ as follow.

Definition 1. *Let $P(t)$ be a partition of $\{0, 1, \cdots, t-\sigma'\}$, where $\sigma' \geq 0$. For any $\kappa \subseteq P(t)$ we have that there exists $t' \leq t$ such that $K(t') = \kappa$, and $\kappa_1 \subseteq P(t)$ such that $K(0) = \kappa_1$. The epoch number for the t-th global iteration, i.e., $v(t)$ is defined as the maximum cardinality of $P(t)$.*

Given the definition of epoch number $v(t)$, we have the following theoretical results for μ-strongly convex problem.

Theorem 1. *Under Assumptions 1-3 and 4, to achieve the accuracy ϵ of problem P for VF$\mathbf{B}^2$-SGD, i.e., $\mathbb{E}(f(w_t) - f(w^*)) \leq \epsilon$, let $\gamma \leq \frac{\epsilon\mu^{1/3}}{(G96L_*^2)^{1/3}}$, if $\tau \leq min\{\epsilon^{-4/3}, \frac{(GL_*^2)^{2/3}}{\epsilon^2\mu^{2/3}}\}$, the epoch number $v(t)$ should satisfy $v(t) \geq \frac{44(GL_*^2)^{1/3}}{\mu^{4/3}\epsilon} log(\frac{2(f(w_0)-f(w^*))}{\epsilon})$, where $L_* = max\{L, \{L_\ell\}_{\ell=1}^q, L_g\}$, $\tau = max\{\tau_1^2, \tau_2^2, \eta_1^2\}$, w^0 and w^* denote the initial point and optimal point, respectively.*

Theorem 2. *Under Assumptions 1-3 and 4, to achieve the accuracy ϵ of problem P for VF$\mathbf{B}^2$-SVRG, let $C = (L_*^2\gamma + L_*)\frac{\gamma^2}{2}$ and $\rho = \frac{\gamma\mu}{2} - \frac{16L_*^2\eta_1 C}{\mu}$, we can carefully choose γ such that*

$$1)\ 1 - 2L_*^2\gamma^2\tau > 0;\ \ 2)\ \rho > 0;\ \ 3)\ \frac{8L_*^2\tau^{1/2}C}{\rho\mu} \leq 0.05;$$
$$4)\ L_*^2\gamma^2\tau^{3/2}(28C + 5\gamma)\frac{36G}{\rho(1 - 2L_*^2\gamma^2\tau)} \leq \frac{\epsilon}{8}, \tag{7}$$

where $v(t)$ should satisfy $v(t) \geq \frac{log0.25}{log(1-\rho)}$ and the outer loop number S should satisfy $S \geq \frac{1}{log\frac{4}{3}} log\frac{2f(w_0)-f(w^*)}{\epsilon}$.

Theorem 3. *Under Assumptions 1-3 and 4, to achieve the accuracy ϵ of problem P for VF$\boldsymbol{B}^2$-SAGA, let $c_0 = \left(2\gamma^3\tau^{3/2} + (L_*^2\gamma^3\tau + L_*\gamma^2)180\gamma^2\tau^{3/2} + 8\gamma^2\tau\right)\frac{18GL_*^2}{1-72L_*^2\gamma^2\tau}$, $c_1 = 2L_*^2\tau(L_*^2\gamma^3\tau + L_*\gamma^2)$, $c_2 = 4(L_*^2\gamma^3\tau + L_*\gamma^2)\frac{L_*^2\tau}{n}$, and $\rho \in (1-\frac{1}{n}, 1)$, we can choose γ such that*

$$1)\ 1 - 72L_*^2\gamma^2\tau > 0;\ 2)\ 0 < 1 - \frac{\gamma\mu}{4} < 1;$$

$$3)\ \frac{4c_0}{\gamma\mu(1-\rho)\left(\frac{\gamma\mu^2}{4} - 2c_1 - c_2\right)} \leq \frac{\epsilon}{2};$$

$$4)\ -\frac{\gamma\mu^2}{4} + 2c_1 + c_2\left(1 + (1 - \frac{1-\frac{1}{n}}{\rho})^{-1}\right) \leq 0;$$

$$5)\ -\frac{\gamma\mu^2}{4} + c_2 + c_1\left(2 + (1 - \frac{1-\frac{1}{n}}{\rho})^{-1}\right) \leq 0, \quad (8)$$

the epoch number $v(t)$ should satisfy $v(t) \geq \frac{1}{\log\frac{1}{\rho}} \log\frac{2\left(2\rho-1+\frac{\gamma\mu}{4}\right)(f(w_0)-f(w^*))}{\epsilon\left(\rho-1+\frac{\gamma\mu}{4}\right)\left(\frac{\gamma\mu^2}{4}-2c_1-c_2\right)}$.

Remark 1. *For strongly convex problems, given the assumptions and parameters in corresponding theorems, the convergence rate of VF$\boldsymbol{B}^2$-SGD is $\mathcal{O}(\frac{1}{\epsilon}log(\frac{1}{\epsilon}))$, and those of VF$\boldsymbol{B}^2$-SVRG and VF$\boldsymbol{B}^2$-SAGA are $\mathcal{O}(log(\frac{1}{\epsilon}))$.*

Convergence Analysis for Nonconvex Problem

Assumption 5. *Nonconvex function $f(w)$ is bounded below,*

$$f^* := \inf_{w\in\mathbb{R}^d} f(w) > -\infty. \quad (9)$$

Assumption 5 guarantees the feasibility of nonconvex problem (P). For nonconvex problem, we introduce the notation $K'(t)$ that denotes a set of q iterations fully visiting all coordinates, *i.e.*, $K'(t) = \{\{t, t+\bar{t}_1, \cdots, t+\bar{t}_{q-1}\} : \psi(\{t, t+\bar{t}_1, \cdots, t+\bar{t}_{q-1}\}) = \{1, \cdots, q\}\}$, where the t-th global iteration denotes a dominated update. Moreover, these iterations are performed respectively on a dominator and $q-1$ different collaborators receiving ϑ calculated at the t-th global iteration. Moreover, we assume that $K'(t)$ can be completed in η_2 global iterations, *i.e.*, for $\forall t' \in \mathcal{A}(t)$, there is $\eta_2 \geq \max\{u | u \in K'(t')\} - t'$. Note that, different from $K(t)$, there is $|K'(t)| = q$ and the definition of $K'(t)$ does not emphasize on "successive iterations" due to the difference of analysis techniques between strongly convex and nonconvex problems. Based on $K'(t)$, we introduce the epoch number $v'(t)$ as follow.

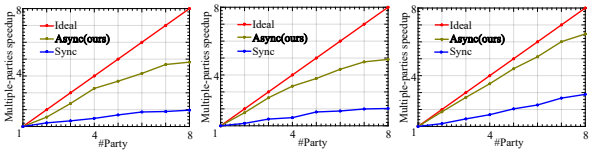
Definition 2. *$\mathcal{A}(t)$ denotes a set of global iterations, where for $\forall\, t' \in \mathcal{A}(t)$ there is the t'-th global iteration denoting a dominated update and $\cup_{\forall t' \in \mathcal{A}(t)} K'(t') = \{0, 1, \cdots, t\}$. The epoch number $v'(t)$ is defined as $|\mathcal{A}(t)|$.*

Give the definition of epoch number $v'(t)$, we have the following theoretical results for nonconvex problem.

Theorem 4. *Under Assumptions 1-3 and 5, to achieve the ϵ-first-order stationary point of problem P, i.e. $\mathbb{E}\|\nabla f(w)\| \leq \epsilon$ for stochastic variable w, for VF$\boldsymbol{B}^2$-SGD, let $\gamma = \frac{\epsilon}{L_* qG}$, if $\tau \leq \frac{512qG}{\epsilon^2}$, the total epoch number T should satisfy*

$$T \geq \frac{\mathbb{E}\left[f(w^0) - f^*\right] L_* qG}{\epsilon^2}, \quad (10)$$

where $L_* = max\{L, \{L_\ell\}_{\ell=1}^q, L_g\}$, $\tau = max\{\tau_1^2, \tau_2^2, \eta_2^2\}$, $f(w^0)$ is the initial function value and f^* is defined in Eq. 9.



(a) SGD-based (b) SVRG-based (c) SAGA-based

Figure 2: q-parties speedup scalability with $m = 2$ on D_4.

Theorem 5. *Under Assumptions 1-3 and 5, to solve problem P with VF$\boldsymbol{B}^2$-SVRG, let $\gamma = \frac{m_0}{L_* n^\alpha}$, where $0 < m_0 < \frac{1}{8}$, $0 < \alpha \leq 1$, if epoch number N in an outer loop satisfies $N \leq \lfloor \frac{n^\alpha}{2m_0} \rfloor$, and $\tau < min\{\frac{n^{2\alpha}}{20m_0^2}, \frac{1-8m_0}{40m_0^2}\}$, there is*

$$\frac{1}{T}\sum_{s=1}^{S}\sum_{t=0}^{N-1}\mathbb{E}||\nabla f(w_{t_0}^s)||^2 \leq \frac{L_* n^\alpha \mathbb{E}\left[f(w_0) - f(w^*)\right]}{T\sigma}, \quad (11)$$

where T is the total number of epoches, t_0 is the start iteration of epoch t, σ is a small value independent of n.

Theorem 6. *Under Assumptions 1-3 and 5, to solve problem P with VF$\boldsymbol{B}^2$-SAGA, let $\gamma = \frac{m_0}{L_* n^\alpha}$, where $0 < m_0 < \frac{1}{20}$, $0 < \alpha \leq 1$, if total epoch number T satisfies $T \leq \lfloor \frac{n^\alpha}{4m_0} \rfloor$ and $\tau < min\{\frac{n^{2\alpha}}{180m_0^2}, \frac{1-20m_0}{40m_0^2}\}$, there is*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}||\nabla f(w_{t_0})||^2 \leq \frac{L_* n^\alpha \mathbb{E}\left[f(w_0) - f(w^*)\right]}{T\sigma}. \quad (12)$$

Remark 2. *For nonconvex problems, given conditions in the theorems, the convergence rate of VF$\boldsymbol{B}^2$-SGD is $\mathcal{O}(1/\sqrt{T})$, and those of VF$\boldsymbol{B}^2$-SVRG and VF$\boldsymbol{B}^2$-SAGA are $\mathcal{O}(1/T)$.*

Security Analysis

We discuss the data security and model security of VF$\boldsymbol{B}^2$ under two semi-honest threat models commonly used in security analysis (Cheng et al. 2019; Xu et al. 2019; Gu et al. 2020a). Specially, these two threat models have different threat abilities, where threat model 2 allows collusion between parties while threat model 1 does not.

- **Honest-but-curious** (threat model 1): All workers will follow the algorithm to perform the correct computations. However, they may use their own retained records of the intermediate computation result to infer other worker's data and model.
- **Honest-but-colluding** (threat model 2): All workers will follow the algorithm to perform the correct computations. However, some workers may collude to infer other worker's data and model by sharing their retained records of the intermediate computation result.

Similar to (Gu et al. 2020a), we prove the security of VF$\mathbf{B}^2$ by analyzing and proving its ability to prevent inference attack defined as follows.

Definition 3 (Inference attack)**.** *An inference attack on the ℓ-th party is to infer $(x_i)_{\mathcal{G}_\ell}$ (or $w_{\mathcal{G}_\ell}$) belonging to other parties or y_i hold by active parties without directly accessing them.*

Lemma 1. *Given an equation $o_i = w_{\mathcal{G}_\ell}^\top (x_i)_{\mathcal{G}_\ell}$ or $o_i = \frac{\partial \mathcal{L}(\widehat{w}^\top x_i, y_i)}{\partial (\widehat{w}^\top x_i)}$ with only o_i being known, there are infinite different solutions to this equation.*

The proof of lemma 1 is shown in the arXiv version. Based on lemma 1, we obtain the following theorem.

Theorem 7. *Under two semi-honest threat models, VF$\mathbf{B}^2$ can prevent the inference attack.*

Feature and model security: During the aggregation, the value of $o_i = w_{\mathcal{G}_\ell}^\top (x_i)_{\mathcal{G}_\ell}$ is masked by δ_ℓ and just the value of $w_{\mathcal{G}_\ell}^\top (x_i)_{\mathcal{G}_\ell} + \delta_\ell$ is transmitted. Under threat model 1, one even can not access the true value of o_i, let alone using relation $o_i = w_{\mathcal{G}_\ell}^\top (x_i)_{\mathcal{G}_\ell}$ to refer $w_{\mathcal{G}_\ell}^\top$ and $(x_i)_{\mathcal{G}_\ell}$. Under threat model 2, the random value δ_ℓ has risk of being removed from term $w_{\mathcal{G}_\ell}^\top (x_i)_{\mathcal{G}_\ell} + \delta_\ell$ by colluding with other parties. Applying lemma 1 to this circumstance, and we have that even if the random value is removed it is still impossible to exactly refer $w_{\mathcal{G}_\ell}^\top$ and $(x_i)_{\mathcal{G}_\ell}$. Thus, the aggregation process can prevent inference attack under two semi-honest threat models.

Label security: When analyze the security of label, we do not consider the collusion between active parties and passive parties, which will make preventing labels from leaking meaningless. In the backward updating process, if a passive party ℓ wants to infer y_i through the received ϑ, it must solve the equation $\vartheta = \frac{\partial \mathcal{L}(\widehat{w}^\top x_i, y_i)}{\partial (\widehat{w}^\top x_i)}$. However, only ϑ is known to party ℓ. Thus, following from lemma 1, we have that it is impossible to exactly infer the labels. Moreover, the collusion between passive parties has no threats to the security of labels. Therefore, the backward updating can prevent inference attack under two semi-honest threat models.

From above analyses, we have that the feature security, label security and model security are guaranteed in VFB^2.

Experiments

In this section, extensive experiments are conducted to demonstrate the efficiency, scalability and losslessness of our algorithms. More experiments are presented in the arXiv version.

Experiment Settings: All experiments are implemented on a machine with four sockets, and each sockets has 12 cores. To simulate the environment with multiple machines (or parties), we arrange an extra thread for each party to schedule its k threads and support communication with (threads of) the other parties. We use MPI to implement the communication scheme. The data are partitioned vertically and randomly into q non-overlapped parts with nearly equal number of features. The number of threads within each parties, *i.e.* k, is set as m. We use the training dataset or randomly select 80% samples as the training data, and the testing dataset or the rest as the testing data. An optimal learning rate γ is chosen from $\{5e^{-1}, 1e^{-1}, 5e^{-2}, 1e^{-2}, \cdots\}$ with regularization coefficient $\lambda = 1e^{-4}$ for all experiments.

	Financial		Large-Scale	
	D_1	D_2	D_3	D_4
#Samples	24,000	96,257	17,996	175,000
#Features	90	92	1,355,191	16,609,143

Table 1: Dataset Descriptions.

Datasets: We use four classification datasets summarized in Table 1 for evaluation. Especially, D_1 (UCICreditCard) and D_2 (GiveMeSomeCredit) are the real financial datsets from the Kaggle website*, which can be used to demonstrate the ability to address real-world tasks; D_3 (news20) and D_4 (webspam) are the large-scale ones from the LIBSVM (Chang and Lin 2011) website†. Note that we apply one-hot encoding to categorical features of D_1 and D_2 , thus the number of features become 90 and 92, respectively.

Problems: We consider ℓ_2-norm regularized logistic regression problem for μ-strong convex case

$$\min_{w\in\mathbb{R}^d} f(w) := \frac{1}{n}\sum_{i=1}^{n} \log(1 + e^{-y_i w^\top x_i}) + \frac{\lambda}{2}\|w\|^2, \quad (13)$$

and the nonconvex logistic regression problem

$$\min_{w\in\mathbb{R}^d} f(w) := \frac{1}{n}\sum_{i=1}^{n} \log(1 + e^{-y_i w^\top x_i}) + \frac{\lambda}{2}\sum_{i=1}^{d} \frac{w_i^2}{1 + w_i^2}.$$

Evaluations of Asynchronous Efficiency and Scalability

To demonstrate the asynchronous efficiency, we introduce the synchronous counterparts of our algorithms (*i.e.*, synchronous VFL algorithms with BUM, denoted as VF**B**) for comparison. When implementing the synchronous algorithms, there is a synthetic straggler party which may be 30% to 50% slower than the faster party to simulate the real application scenario with unbalanced computational resource.

Asynchronous Efficiency: In these experiments, we set $q = 8$, $m = 3$ and fix the γ for algorithms with a same SGD-type but in different parallel fashions. As shown in Figs. 3 and 4, the loss v.s. run time curves demonstrate that our algorithms consistently outperform their synchronous counterparts regarding the efficiency.

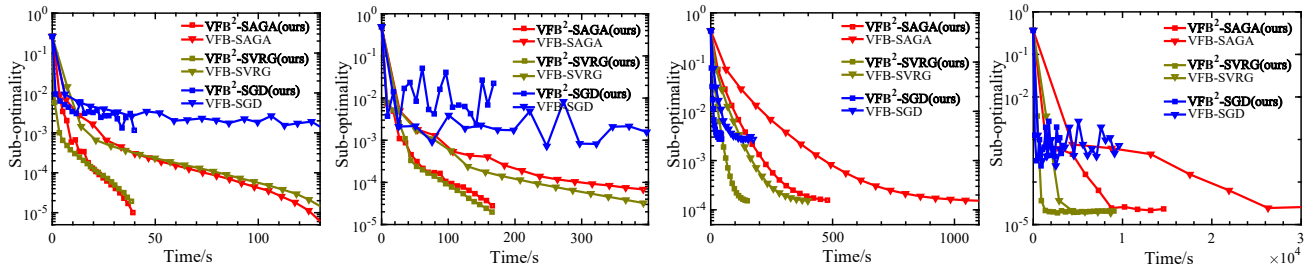
Moreover, from the perspective of loss v.s. epoch number, we have that algorithms based on SVRG and SAGA have the better convergence rate than that of SGD-based algorithms which is consistent to the theoretical results.

Asynchronous Scalability: We also consider the asynchronous speedup scalability in terms of the number of total parties q. Given a fixed m, q-parties speedup is defined as

$$q\text{-parties speedup} = \frac{\text{Run time of using 1 party}}{\text{Run time of using } q \text{ parties}}, \quad (14)$$

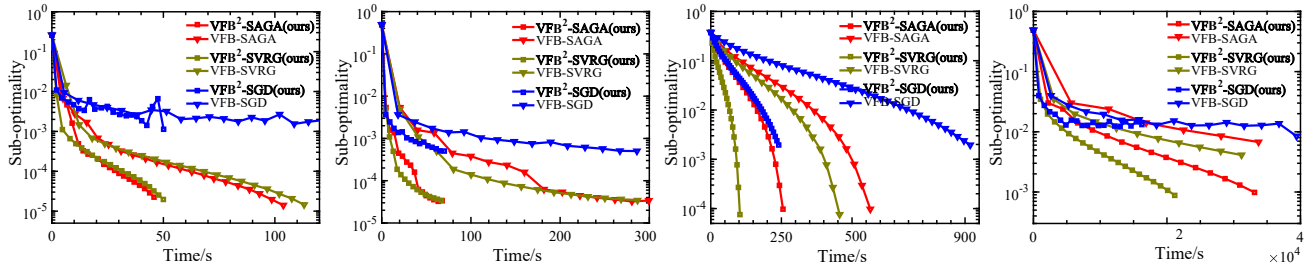
*https://www.kaggle.com/datasets

†https://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets/



(a) Data: D_1 (b) Data: D_2 (c) Data: D_3 (d) Data: D_4

Figure 3: Results for solving μ-strongly convex VFL models (Problem 13), where the number of epoches (points) denotes how many passes over the dataset the algorithm makes.



(a) Data: D_1 (b) Data: D_2 (c) Data: D_3 (d) Data: D_4

Figure 4: Results for solving nonconvex VFL models (Problem 14), where the number of epoches (points) denotes how many passes over the dataset the algorithm makes.

	Algorithm	D_1	D_2	D_3	D_4
Problem (13)	NonF	81.96%±0.25%	93.56%±0.19%	98.29%±0.21%	92.17%±0.12%
	AFSVRG-VP	79.35%±0.19%	93.35%±0.18%	97.24%±0.11%	89.17%±0.10%
	Ours	81.96%±0.22%	93.56%±0.20%	98.29%±0.20%	92.17%±0.13%
Problem (14)	NonF	82.03%±0.32%	93.56%±0.25%	98.45%±0.29%	92.71%±0.24%
	AFSVRG-VP	79.36%±0.24%	93.35%±0.22%	97.59%±0.13%	89.98%±0.14%
	Ours	82.03%±0.34%	93.56%±0.24%	98.45%±0.33%	92.71%±0.27%

Table 2: Accuracy of different algorithms to evaluate the losslessness of our algorithms (10 trials).

where run time is defined as time spending on reaching a certain precision of sub-optimality, *i.e.*, $1e^{-3}$ for D_4. We implement experiment for Problem (14), results of which are shown in Fig. 2. As depicted in Fig. 2, our asynchronous algorithms has much better q-parties speedup scalability than synchronous ones and can achieve near linear speedup.

Evaluation of Losslessness

To demonstrate the losslessness of our algorithms, we compare VF**B**^{2}-SVRG with its non-federated (NonF) counterpart (all data are integrated together for modeling) and ERCR based algorithm but without BUM, *i.e.*, AFSVRG-VP proposed in (Gu et al. 2020b). Especially, AFSVRG-VP also uses distributed SGD method but can not optimize the parameters corresponding to passive parties due to lacking labels. When implementing AFSVRG-VP, we assume that only half parties have labels, *i.e.*, parameters corresponding to the features held by the other parties are not optimized. Each comparison is repeated 10 times with $m = 3$, $q = 8$, and a same stop criterion, *e.g.*, $1e^{-5}$ for D_1. As shown in Table 2, the accuracy of our algorithms are the same with those of NonF algorithms and are much better than those of AFSVRG-VP, which are consistent to our claims.

Conclusion

In this paper, we proposed a novel backward updating mechanism for the real VFL system where only one or partial parties have labels for training models. Our new algorithms enable all parties, rather than only active parties, to collaboratively update the model and also guarantee the algorithm convergence, which was not held in other recently proposed ERCR based VFL methods under the real-world setting. Moreover, we proposed a bilevel asynchronous parallel architecture to make ERCR based algorithms with backward updating more efficient in real-world tasks. Three practical SGD-type of algorithms were also proposed with theoretical guarantee.

Acknowledgments

Q.S. Zhang and C. Deng were supported in part by the National Natural Science Foundation of China under Grant 62071361, the National Key R&D Program of China under Grant 2017YFE0104100, and the China Research Project under Grant 6141B07270429.

References

Bottou, L. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, 177–186. Springer.

Chang, C.-C.; and Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)* 2(3): 27.

Cheng, K.; Fan, T.; Jin, Y.; Liu, Y.; Chen, T.; and Yang, Q. 2019. SecureBoost: A Lossless Federated Learning Framework. *arXiv preprint arXiv:1901.08755* .

Conroy, B.; and Sajda, P. 2012. Fast, exact model selection and permutation testing for l2-regularized logistic regression. In *Artificial Intelligence and Statistics*, 246–254.

Dang, Z.; Li, X.; Gu, B.; Deng, C.; and Huang, H. 2020. Large-Scale Nonlinear AUC Maximization via Triply Stochastic Gradients. *IEEE Transactions on Pattern Analysis and Machine Intelligence* .

Defazio, A.; Bach, F.; and Lacoste-Julien, S. 2014. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in NIPS*, 1646–1654.

Gascón, A.; Schoppmann, P.; Balle, B.; Raykova, M.; Doerner, J.; Zahur, S.; and Evans, D. 2016. Secure Linear Regression on Vertically Partitioned Datasets. *IACR Cryptology ePrint Archive* 2016: 892.

Gong, Y.; Fang, Y.; and Guo, Y. 2016. Private data analytics on biomedical sensing data via distributed computation. *IEEE/ACM transactions on computational biology and bioinformatics* 13(3): 431–444.

Gu, B.; Dang, Z.; Li, X.; and Huang, H. 2020a. Federated Doubly Stochastic Kernel Learning for Vertically Partitioned Data. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2483–2493.

Gu, B.; Xu, A.; Deng, C.; and Huang, h. 2020b. Privacy-Preserving Asynchronous Federated Learning Algorithms for Multi-Party Vertically Collaborative Learning. *arXiv preprint arXiv:2008.06233* .

Hardy, S.; Henecka, W.; Ivey-Law, H.; Nock, R.; Patrini, G.; Smith, G.; and Thorne, B. 2017. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *arXiv preprint arXiv:1711.10677* .

Hu, Y.; Niu, D.; Yang, J.; and Zhou, S. 2019. FDML: A collaborative machine learning framework for distributed features. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2232–2240.

Huang, F.; Chen, S.; and Huang, H. 2019. Faster Stochastic Alternating Direction Method of Multipliers for Nonconvex Optimization. In *ICML*, 2839–2848.

Huang, F.; Gao, S.; Pei, J.; and Huang, H. 2020. Accelerated zeroth-order momentum methods from mini to minimax optimization. *arXiv preprint arXiv:2008.08170* .

Huo, Z.; and Huang, H. 2017. Asynchronous mini-batch gradient descent with variance reduction for non-convex optimization. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Johnson, R.; and Zhang, T. 2013. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in NIPS*, 315–323.

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. 2019. Advances and Open Problems in Federated Learning. *arXiv preprint arXiv:1912.04977* .

Kim, M.; Song, Y.; Wang, S.; Xia, Y.; and Jiang, X. 2018. Secure logistic regression based on homomorphic encryption: Design and evaluation. *JMIR medical informatics* 6(2): e19.

Leblond, R.; Pedregosa, F.; and Lacoste-Julien, S. 2017. Asaga: Asynchronous Parallel Saga. In *20th International Conference on Artificial Intelligence and Statistics (AISTATS) 2017*.

Li, M.; Deng, C.; Li, T.; Yan, J.; Gao, X.; and Huang, H. 2020. Towards Transferable Targeted Attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 641–649.

Lian, X.; Huang, Y.; Li, Y.; and Liu, J. 2015. Asynchronous parallel stochastic gradient for nonconvex optimization. In *Advances in Neural Information Processing Systems*, 2737–2745.

Liu, F.; Ng, W. K.; and Zhang, W. 2015. Encrypted gradient descent protocol for outsourced data mining. In *2015 IEEE 29th International Conference on Advanced Information Networking and Applications*, 339–346. IEEE.

Liu, Y.; Liu, Y.; Liu, Z.; Zhang, J.; Meng, C.; and Zheng, Y. 2019. Federated Forest. *arXiv preprint arXiv:1905.10053* .

McMahan, H. B.; Moore, E.; Ramage, D.; Hampson, S.; et al. 2016. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629* .

Shen, X.; Alam, M.; Fikse, F.; and Rönnegård, L. 2013. A novel generalized ridge regression method for quantitative genetics. *Genetics* 193(4): 1255–1268.

Smith, V.; Chiang, C.-K.; Sanjabi, M.; and Talwalkar, A. S. 2017. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, 4424–4434.

Wang, X.; Ma, S.; Goldfarb, D.; and Liu, W. 2017. Stochastic quasi-Newton methods for nonconvex stochastic optimization. *SIAM Journal on Optimization* 27(2): 927–956.

Wang, Z.; Ji, K.; Zhou, Y.; Liang, Y.; and Tarokh, V. 2019. SpiderBoost and Momentum: Faster Variance Reduction Algorithms. In *Advances in NIPS*, 2403–2413.

Wei, K.; Yang, M.; Wang, H.; Deng, C.; and Liu, X. 2019. Adversarial Fine-Grained Composition Learning for Unseen Attribute-Object Recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 3741–3749.

Xu, R.; Baracaldo, N.; Zhou, Y.; Anwar, A.; and Ludwig, H. 2019. Hybridalpha: An efficient approach for privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, 13–23.

Yang, K.; Fan, T.; Chen, T.; Shi, Y.; and Yang, Q. 2019a. A Quasi-Newton Method Based Vertical Federated Learning Framework for Logistic Regression. *arXiv preprint arXiv:1912.00513* .

Yang, M.; Deng, C.; Yan, J.; Liu, X.; and Tao, D. 2020a. Learning Unseen Concepts via Hierarchical Decomposition and Composition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10248–10256.

Yang, Q.; Liu, Y.; Chen, T.; and Tong, Y. 2019b. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10(2): 12.

Yang, X.; Deng, C.; Wei, K.; Yan, J.; and Liu, W. 2020b. Adversarial Learning for Robust Deep Clustering. *Advances in Neural Information Processing Systems* 33.

Zhang, G.-D.; Zhao, S.-Y.; Gao, H.; and Li, W.-J. 2018. Feature-Distributed SVRG for High-Dimensional Linear Classification. *arXiv preprint arXiv:1802.03604* .

Zhang, Q.; Huang, F.; Deng, C.; and Huang, H. 2020. Faster Stochastic Quasi-Newton Methods. *arXiv preprint arXiv:2004.06479* .