Applied Intelligence (2023) 53:12399–12415
https://doi.org/10.1007/s10489-022-04111-0

VFL-R: a novel framework for multi-party in vertical federated learning

Jialin Li[1] · **Tongjiang Yan**[1] · **Pengcheng Ren**[1]

Accepted: 9 August 2022 / Published online: 27 September 2022

Abstract
Federated learning (FL) provides a robust distributed framework for machine learning that solves privacy leakage concerns. In the some cases, it is hard to train the FL model with limited communication sources and low computational capabilities for the coordinator. Especially, designing an efficient framework for vertical federated learning (VFL) is a concern, as each party has unique data features. Hence, this paper proposes VFL-R, a novel VFL framework combined with a ring architecture for multi-party cooperative modeling. The VFL-R framework simplifies each party's intricate communication architecture, defending against semi-honest attacks and reducing the coordinator's influence in the modeling process. Several experiments challenge our framework's communication performance against current VFL frameworks, highlighting that for similar test accuracy, VFL-R achieves $O(K)$ number of communications in one communication round and an $O(1)$ communication cost for the coordinator.

Keywords Federated learning · Vertical federated learning · VFL-R framework

1 Introduction

In traditional machine learning (ML) approaches, data are collected and stored by a single node (or centralized server) and used for training and testing. However, transmitting and centralizing data raises numerous administrative, ethical and legal issues, mainly related to privacy and data protection, according to the General Data Protection Regulation (GDPR) [1]. Federated learning (FL) empowers collaborative learning to address data issues while protecting information security [2]. Recently, the FL framework has been increasingly used in real-world applications, e.g., healthcare [3, 4], purchase recommendation [5, 6], and distributed synthetic data generation systems [7, 8].

Generally, the FL framework involves three primary steps: (i) all parties receive the latest global model W from the centralized server (also called a broker), (ii) the parties train the received model using their local data, and (iii) they upload their locally trained models W_i back to the centralized server to be aggregated and form an updated global model. These steps are repeated until a particular convergence criterion is obtained. However, such distributed framework also results in communication cost and leads to a training bottleneck. Currently, communication efficiency is still a significant concern for FL.

Recently, some researchers proposed several frameworks to improve communication efficiency in the horizontal federated learning (HFL) scenario [9–11]. The vertical federated learning (VFL) scenario is opposite to the HFL scenario, where all parties hold homogeneous data, i.e., the parties have partial overlap on the sample space, whereas they differ in the feature space. As a result, the VFL framework requires a more intricate communication architecture to ensure the other parties are unaware of the data and the characteristics of other parties. The literature has proposed several VFL frameworks. For example, in 2019, Yang et al. proposed a simple VFL framework based

This work was supported by Fundamental Research Funds for the Central Universities (20CX05012A), the Major Scientific and Technological Projects of CNPC under Grant (ZD2019-183-008), and Shandong Provincial Natural Science Foundation of China (ZR2019MF070).

✉ Tongjiang Yan
yantoji@163.com

Jialin Li
ljl19960221@163.com

Pengcheng Ren
rpc1995@163.com

[1] China University of Petroleum, Qingdao, 266580, China

on the C-S communication architecture with one parameter server (PS) and two parties [12]. Figure 1 highlights that the PS occurs as a trusted coordinator who is mainly responsible for data aggregation and information distribution. Ou et al. [13] designed a vertical federated learning system utilizing Bayesian machine learning with homomorphic encryption, while Hou et al. [14] proposed a verifiable privacy-preserving scheme (VPRF) based on a vertical federated random forest. However, the stability and reliability of the PS are pretty important, as once the PS fails to provide accurate computation results, the VFL may produce a low-quality model [15]. To eliminate the effect of PS, Chen et al. [16] proposed a secure VFL framework based on a pseudo-decentralization communication architecture. As illustrated in Fig. 2, the parties are divided into one active and many passive parties, where the active party replaces the position of PS as the coordinator. In 2021, Zhu et al. [17] introduced a secure VFL framework named PIVODL, which trained GBDTs with data labels distributed on multiple devices. Zhang et al. [18] suggested a VFL framework based on an LSTM fault classification network for the firefighting IoT platform. Chen et al. [19] proposed an efficient and interpretable inference framework for decision tree ensembles in a VFL scenario. However, the pseudo-decentralization communication architecture still needs many communications to achieve high test accuracy and privacy security. Real-word applications involving an intricate communication architecture impose high time and money costs. Although Gu et al. [20] proposed an efficient VFL framework called VFB^2 to simplify the communication architecture, VFB^2 still suffers from semi-honest attacks [21] and the coordinator's effect. Hence, it is quite challenging to design a framework that considers both communication efficiency and privacy security for the VFL scenario.

In addition, a simplified framework is urgently needed to complete VFL modeling with limited communication sources and low coordinator's effect. Hence, this paper proposes VFL-R, a novel VFL framework integrated with the ring architecture and a HE-based approach, enabling a multi-party scheme to train the model collaboratively. We summarize the contributions of this paper as follows.

- We first incorporate the ring communication architecture into the VFL framework. Hence, our novel VFL framework avoids the complicated communication protocol and reduces the coordinator's effect. The performance of the VFL-R framework is evaluated based on benchmark datasets and challenged against other frameworks. The experimental results reveal that VFL-R effectively reduces the coordinator's communication cost during the modeling process while preserving a high test accuracy.
- We provide our framework's detailed theoretical analysis of the loss function and gradient. This is important as the theoretical analysis affords a better understanding

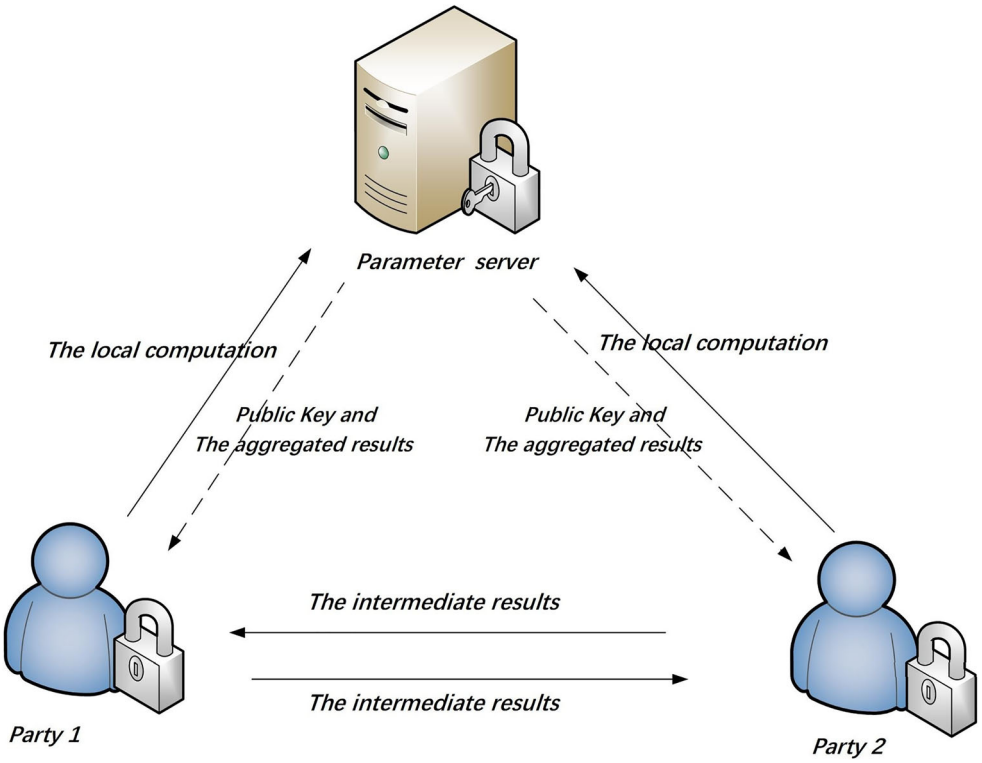

Fig. 1 The VFL framework based on the C-S communication architecture

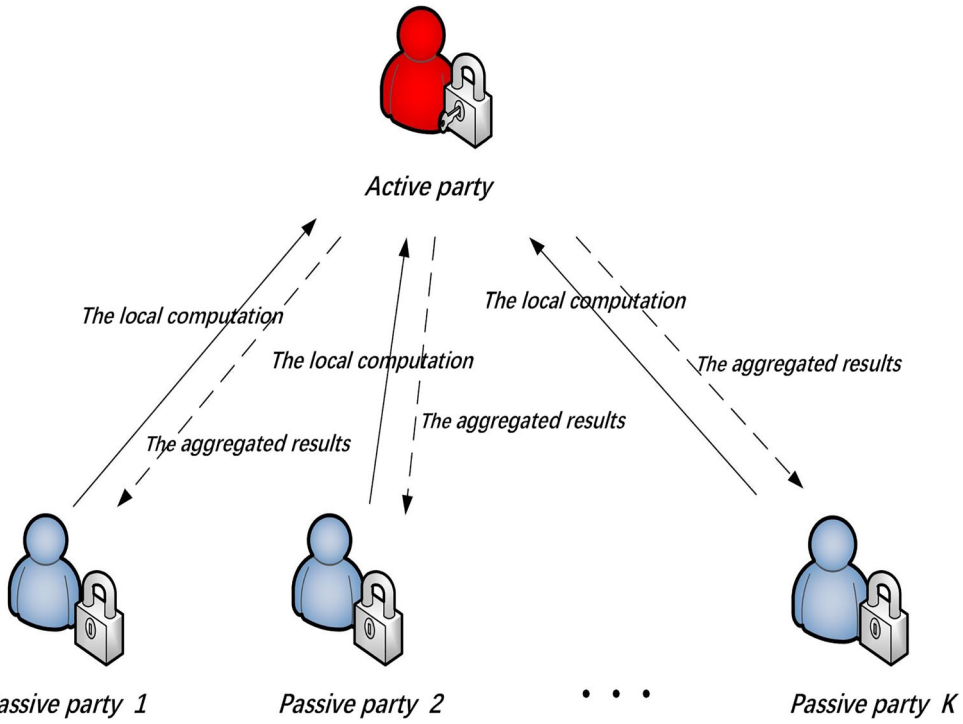

Fig. 2 The VFL framework based on the pseudo-decentralization communication architecture

of the framework's operating mechanism and optimizes the model.

- To protect the privacy security of each party, we integrate a HE-based approach into our framework. Meanwhile, we analyze the classic semi-honest threat models and demonstrate VFL-R's robustness to semi-honest attacks.

The remainder of this paper is organized as follows. Section 2 introduces some necessary methods and concepts for the VFL-R framework. Section 3 defines our framework's new model formula, while Section 4 describes the proposed framework in detail. Section 5 presents the security analysis, and Sections 6, 7 and 8 present the experimental setup, varying experiment settings, and challenge VFL-R against different VFL frameworks. Finally, Section 9 concludes this work and provides some future research direction.

2 Preliminary

This section introduces some methods and concepts for the VFL-R framework.

2.1 Paillier homomorphic encryption

Our framework employs the Paillier Homomorphic Encryption (PHE) [22, 23] scheme to protect data privacy, an additive homomorphic encryption method that performs addition and multiplication on the encrypted values. This paper defines a new encryption operation $'\odot'$.

Definition 1 (Encryption operation $'\odot'$) For any $a, b \in \mathbb{R}^n$, the $'\odot'$ performs the following calculation:

- $a \oplus b = [[a]] + [[b]] = [[a + b]]$ (addition)
- $a \otimes b = a^T[[b]] = [[a^T b]]$ (scalar product),

where the additive homomorphic encryption of a vector u ($u \in \mathbb{R}^n$) is represented as $[[u]]$.

2.2 Vertical federated learning

Let a training sample be $\{(\mathbf{x}_i, y_i) : i = 1, 2, \ldots, n\}$, where $\mathbf{x}_i$ ($\mathbf{x}_i \in \mathbb{R}^d$) and y_i denote the input vector and output label, respectively, d is the feature's dimension of the training sample. For the VFL setting, $\mathbf{x}_i$ is vertically distributed among K parties and each party owns a disjoint subset of features vector $\mathbf{x}_{[i,k]}$ ($\mathbf{x}_{[i,k]} \in \mathbb{R}^{d_k}, k = 1, 2, \ldots, K$), where d_k is the features dimension of the k-th party and $\sum_{k=1}^{K} d_k = d$. Similarly, we define $\Theta = [\theta_1; \theta_2; \ldots; \theta_K]$, where θ_k ($\theta_k \in \mathbb{R}^{d_k}$) denotes the parameter of the k-th party. Suppose that the K-th party holds the label information $y_{[i,K]}$ $\left(y_{[i,K]} \in \mathbb{R}\right)$, we focus on the following empirical risk minimization function:

$$\min_{\theta} \mathcal{L}(\Theta) \triangleq \frac{1}{n} \sum_{i=1}^{n} f\left(\sum_{k=1}^{K} \mathbf{x}_{[i,k]}\theta_i, y_{[i,K]}\right) + \lambda R(\Theta), \quad (1)$$

where $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ is smooth and convex, λ is a tuning parameter, and $f(\cdot)$ and $R(\cdot)$ denote the loss function and regularizer, respectively.

2.3 Gradient descent

The stochastic gradient descent (SGD) [24, 25] is the most commonly used algorithm for solving convex optimization problems. In this paper, we employ the gradient-based method to optimize (1) . Let $\mathcal{L}(\Theta)$ be the derivative, the local parameters θ_k from the k-th party are updated according to:

$$\theta_k^* = \theta_k - \alpha \nabla \mathcal{L}(\theta_k), \tag{2}$$

where α is the learning rate and $\nabla \mathcal{L}(\theta_k)$ denotes the gradient of $\mathcal{L}(\Theta)$ with respect to θ_k. The empirical risk function reaches the step-wise minimum according to the gradient.

2.4 Loss function

The model's loss function depends on the model's purpose and can be regarded as a true function of one variable t ($t \in \mathbb{R}$) [26]. We rewrite the loss function in (1) as $\frac{1}{n}\sum_{i=1}^{n} f(t)$ with $t = w - y$ for regression or $t = wy$ for classification, where $w = \sum_{k=1}^{K} \mathbf{x}_{[i,k]}\theta_i$ and $y = y_{[i,K]}$. Some common loss functions are reported in Table 1.

3 Preparations for VFL-R framework

A natural question arising is which loss function of Table 1 should be adopted by our framework. To answer this question, this section introduces the necessary theoretical analysis and derives a new model formula applicable to our framework.

3.1 Theoretical analysis

For the existing VFL frameworks, Wan et al. [27] assumed that the loss function is implicitly linearly separable in the form of $f(t) = g \circ h(t)$, where g is any differentiable function and $h(t)$ is a linearly separable function in the form of $\sum_{k=1}^{K} h\left(\theta_k, \mathbf{x}_{[i,k]}\right)$. In this paper, we give a new property for the loss function involving the encryption operation $'\odot'$.

Table 1 Typical loss functions used in machine learning

Loss Function	f(t)
the hinge loss	$[1 - wy]_+$
the logistic loss	$\log_2\left(1 + e^{-wy}\right)$
the quadratic loss	$\|w - y\|_2^2$
the absolute value loss	$\|w - y\|$
the huber loss	$\begin{cases} \frac{1}{2}(w-y)^2 & \|w-y\| \le \delta \\ \delta\|w-y\| - \frac{1}{2}\delta^2 & otherwise \end{cases}$

Property 1 (*Encryption composed property*) For $\forall t \in \mathbb{R}$, $[[f(t)]] \in \mathcal{M}\left(\ell;'\odot'\right)$. The encrypted set $\mathcal{M}\left(\ell;'\odot'\right) = \{m \mid m = (\ell;'\odot')\}$, where m comprises the elements from $\ell = \{\ell_1, \ell_2, \ldots, \ell_K\}$ is as follows : $m = \ell_1' \odot' \ell_2' \odot' \ldots' \odot' \ell_K$.

3.2 New model formula in $\mathcal{L}(\Theta)$

In our framework, the K-th party computes the encrypted loss function $[[f(t)]]$ in the form of $\mathcal{M}_{[K]}\left(\ell;'\odot'\right)$, where $[K]$ is the index of the K-th party. We assume that the regularizer satisfies the encryption decomposition property. Then, the K-th party will compute the encrypted regularizer $[[R(\Theta)]]$ in the form of $\mathcal{N}_{[K]}\left(\theta,'\odot'\right)$ with the set $\theta = \{\theta_1, \theta_2, \ldots, \theta_K\}$. The new model formula in (1) can be rewritten as:

$$\frac{1}{n}\sum_{i=1}^{n} \mathcal{M}_{[K]}\left(\ell;'\odot'\right) + \lambda \mathcal{N}_{[K]}\left(\theta;'\odot'\right). \tag{3}$$

3.3 Aggregation of the encrypted gradient $[[\nabla \mathcal{L}(\theta_k)]]$

The encrypted gradient aggregation is important for our framework to update the local parameters. Thus, this subsection introduces the assumption for the gradient.

Assumption 1 *The gradients ∇f and ∇R satisfy Property 1, namely $[[\nabla f]]$ comprises elements from set $\mathcal{A}$ and $[[\nabla R]]$ is composed of elements from set $\mathcal{B}$.*

Theorem 1 *Under Assumption 1, the encrypted gradient $[[\nabla \mathcal{L}(\theta_k)]]$ can be composed of the elements from set $\mathcal{L}$: for $t = w - y$, $\mathcal{L} = \mathcal{A} \cup \mathcal{B} \cup \left\{\mathbf{x}_{[i,k]}\right\}$, and for $t = wy$, $\mathcal{L} = \mathcal{A} \cup \mathcal{B} \cup \left\{\mathbf{x}_{[i,k]}, y_{[i,K]}\right\}$.*

Proof For $t = w - y$, we derive the explicit form of $\nabla \mathcal{L}(\theta_k)$ according to (1) as:

$$\nabla \mathcal{L}(\theta_k) = \frac{1}{n}\sum_{i=1}^{n}\left(\nabla f \times \mathbf{x}_{[i,k]}\right) + \nabla R. \tag{4}$$

Considering the encrypted form

$$\begin{aligned}[[\nabla \mathcal{L}(\theta_k)]] &= \frac{1}{n}\sum_{i=1}^{n}\left(\nabla f \otimes \mathbf{x}_{[i,k]}\right) \oplus \nabla R \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{x}_{[i,k]}^T [[\nabla f]]\right) + [[\nabla R]],\end{aligned} \tag{5}$$

$\exists\ \mathcal{L} = \mathcal{A} \cup \mathcal{B} \cup \left\{\mathbf{x}_{[i,k]}\right\}$, namely the $[[\nabla \mathcal{L}(\theta_k)]]$, is comprises elements from set $\mathcal{L}$.

For $t = wy$, we derive the explicit form of $\nabla \mathcal{L}(\theta_k)$ according to (1) as:

$$\nabla \mathcal{L}\left(\theta_k\right) = \frac{1}{n} \sum_{i=1}^{n} \left(y_{[i,K]} \nabla f \times \mathbf{x}_{[i,k]}\right) + \nabla R. \tag{6}$$

Considering the encrypted form

$$\begin{aligned}[[\nabla \mathcal{L}\left(\theta_k\right)]] &= \frac{1}{n} \sum_{i=1}^{n} \left(y_{[i,K]} \otimes \nabla f \otimes \mathbf{x}_{[i,k]}\right) \oplus \nabla R \\ &= \frac{1}{n} \sum_{i=1}^{n} \left(y_{[i,K]}\, \mathbf{x}_{[i,k]}^{T}\, [[\nabla f]]\right) + [[\nabla R]], \end{aligned} \tag{7}$$

$\exists\ \mathcal{L} = \mathcal{A} \cup \mathcal{B} \cup \left\{\mathbf{x}_{[i,k]}, y_{[i,K]}\right\}$, namely the $[[\nabla \mathcal{L}(\theta_k)]]$ can be composed of the elements from set $\mathcal{L}$. □

According to Theorem 1, we note that the local data $\left\{\mathbf{x}_{[i,k]}, y_{[i,K]}\right\}$ is necessary to compute the $[[\nabla \mathcal{L}(\theta_k)]]$. However, in this paper the K-th party only computes the $[[\nabla f]]$ and $[[\nabla R]]$ encrypted results during the aggregation process, written as $\mathcal{D}_{[K]}\left(D, ' \odot'\right)$, where $D = \mathcal{A} \cup \mathcal{B}$, and as $\{d_1, d_2, \ldots, d_K\}$. Then, each party will compute the encrypted gradient $[[\nabla \mathcal{L}\left(\theta_k\right)]]$ during the local updating process. The purpose is to avoid gradient information leakage and reduce the calculation pressure during the aggregation process.

4 The VFL-R architecture

This section introduces the novel VFL framework based on the ring architecture illustrated in Fig. 3. The design framework has the following characteristics:

- It includes two party types, one coordinator and some workers. The coordinator does not participate in the model training.
- A one-way channel exists among each worker and a two-way channel between the coordinator and the K-th worker.
- During the modeling process, each worker only needs one public key from the coordinator. Changing the encryption pairs in our framework is unnecessary.

4.1 The VFL-R framework

We divide our framework into three phases. In Phase One, the primary task is to aggregate the model function and the encrypted results, while Phase Two aims to perform local updating among each worker. Finally, Phase Three focuses on the decryption of the encrypted local parameters.

a. Phase One

The K-th worker needs to compute the $\mathcal{M}_{[K]}(\ell; ' \odot')$, $\mathcal{N}_{[K]}\left(\theta; ' \odot'\right)$ and $\mathcal{D}_{[K]}\left(D, ' \odot'\right)$. The aggregation ideas are summarized as:

$$\begin{aligned} \mathcal{M}_{[1]} &= \mathcal{M}\left(\ell_1; ' \odot'\right) \\ \mathcal{M}_{[2]} &= \mathcal{M}\left(\mathcal{M}_{[1]} \cup \ell_2; ' \odot'\right) \\ &\cdots\cdots \\ \mathcal{M}_{[K-1]} &= \mathcal{M}\left(\mathcal{M}_{[K-2]} \cup \ell_{K-1}; ' \odot'\right) \\ \mathcal{M}_{[K]}\left(\ell; ' \odot'\right) &\in \mathcal{M}\left(\mathcal{M}_{[K-1]} \cup \ell_K; ' \odot'\right). \end{aligned} \tag{8}$$

Denote that the encrypted set $\mathcal{M}_{[i]}\ (i = 1, 2, \ldots, K-1)$. Each element in $\mathcal{M}_{[i]}$ can be used to compute $\mathcal{M}_{[K]}\left(\ell; ' \odot'\right)$. The *1*-st worker computes the encrypted ℓ_1 to $\mathcal{M}_{[1]}$, while the *2*-nd worker computes new elements based on the elements from $\mathcal{M}_{[1]} \cup \ell_2$. With the transfer of $\mathcal{M}_{[i]}$ suggested in our proposed framework, we increase the element availability when computing the target model. Hence, the K-th worker will compute the $\mathcal{M}_{[K]}\left(\ell; ' \odot'\right)$ and similarly, the $\mathcal{N}_{[K]}\left(\theta; ' \odot'\right)$ can be aggregated as:

$$\begin{aligned} \mathcal{N}_{[1]} &= \mathcal{N}\left(\theta_1; ' \odot'\right) \\ \mathcal{N}_{[2]} &= \mathcal{N}\left(\mathcal{N}_{[1]} \cup \theta_2; ' \odot'\right) \\ &\cdots\cdots \\ \mathcal{N}_{[K-1]} &= \mathcal{N}\left(\mathcal{N}_{[K-2]} \cup \theta_{K-1}; ' \odot'\right) \\ \mathcal{N}_{[K]}\left(\theta; ' \odot'\right) &\in \mathcal{N}\left(\mathcal{N}_{[K-1]} \cup \theta_K; ' \odot'\right). \end{aligned} \tag{9}$$

The $\mathcal{D}_{[K]}\left(D, ' \odot'\right)$ can be aggregated as:

$$\begin{aligned} \mathcal{D}_{[1]} &= \mathcal{D}\left(d_1; ' \odot'\right) \\ \mathcal{D}_{[2]} &= \mathcal{D}\left(\mathcal{D}_{[1]} \cup d_2; ' \odot'\right) \\ &\cdots\cdots \\ \mathcal{D}_{[K-1]} &= \mathcal{D}\left(\mathcal{D}_{[K-2]} \cup d_{K-1}; ' \odot'\right) \\ \mathcal{D}_{[K]}\left(D; ' \odot'\right) &\in \mathcal{D}\left(\mathcal{D}_{[K-1]} \cup d_K; ' \odot'\right). \end{aligned} \tag{10}$$

This phase includes the following three steps.

- **Step 1:** The coordinator creates encryption pairs and sends the public key to the K-th worker. Then, the K-th worker sends the public key to the *1*-st worker.
- **Step 2:** The *1*-st worker receives the public key and computes the encrypted sets $\mathcal{M}_{[1]}, \mathcal{N}_{[1]}, \mathcal{D}_{[1]}$. The *2*-nd worker does the same operations as the *1*-st worker after receiving the public key and $\mathcal{M}_{[1]}, \mathcal{N}_{[1]}, \mathcal{D}_{[1]}$. This process is repeated until the encrypted sets $\mathcal{M}_{[K-1]}, \mathcal{N}_{[K-1]}, \mathcal{D}_{[K-1]}$ are sent to the K-th worker.
- **Step 3:** The K-th worker completes the aggregation of $\mathcal{M}_{[K]}\left(\ell; ' \odot'\right), \mathcal{N}_{[K]}\left(\theta; ' \odot'\right)$ and $\mathcal{D}_{[K]}\left(D, ' \odot'\right)$.

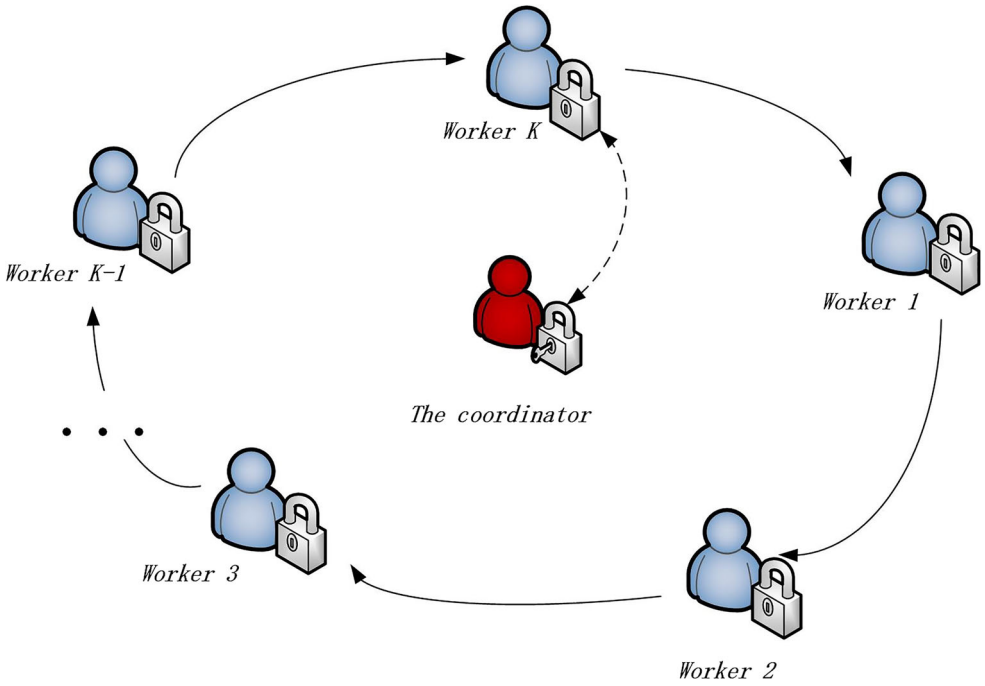

Fig. 3 The pipeline of VFL-R framework

b. Phase Two

In this phase, each worker computes the encrypted gradient and updates the local parameters. The steps are as follows.

- **Step 4:** The K-th worker uses the $\mathcal{D}_{[K]}\left(D,'\odot'\right)$ to compute the $[[\nabla\mathcal{L}(\theta_K)]]$ and updates the local parameters in the form of $[[\theta_K^*]] = [[\theta_K]] - \alpha[[\nabla\mathcal{L}(\theta_K)]]$ under the ciphertext environment. Next, the $\mathcal{D}_{[K]}\left(D,'\odot'\right)$ is sent to the *1*-st worker and the *1*-st worker does the same things as the last worker. This procedure repeats until all workers complete the local updating.
- **Step 5:** As illustrated in Figs. 4 and 5, all workers perform Steps 2-4 during the t-th $(1 < t < T)$ iteration. The coordinator does not play any role during the modeling process and rarely has access to the intermediate results concerning the target model.

c. Phase three

Since the local parameter updating is in the ciphertext environment, it is necessary to decrypt the local parameters in the T-th iteration.

- **Step 6:** As illustrated in Fig. 6, the K-th worker sends $[[\theta_K]]$ to the *1*-th worker after updating the local parameters. Then the *1*-th worker sends the $\{[[\theta_1]], [[\theta_K]]\}$ to the *2*-th worker after updating the local parameters. This process repeats until the encrypted set $\Theta = \{[[\theta_1]], [[\theta_2]], \ldots, [[\theta_K]]\}$ is sent to the coordinator. The coordinator will decrypt he encrypted set using its private key.

5 Security analysis

This section discusses our framework's privacy security. Given that the semi-honest threat models have been widely used in FL security analysis [28–30], we introduce two assumptions for semi-honest threat models and analyze the privacy security from two aspects: the coordinator and the workers.

Assumption 2 (Honest-but-curious) *Each party follows the designing protocol to perform the correct computations. However, some parties may infer the other party's raw data and model by retaining the intermediate computation result records.*

Assumption 3 (Honest-but-colluding) *Each party follows the designing protocol to perform the correct computations. Unlike Assumption 2, some parties may collude to infer the other party's raw data and model by sharing their own retained records.*

For workers In our framework, each worker passes the intermediate results and updates local parameters in the ciphertext environment. Workers usually receive the encrypted values from other workers, while under the encryption protection, it is challenging to perform inference attacks for other workers under Assumptions 2–3.

For the coordinator In our framework, the coordinator's task is to distribute the public key in the *1*-th iteration and

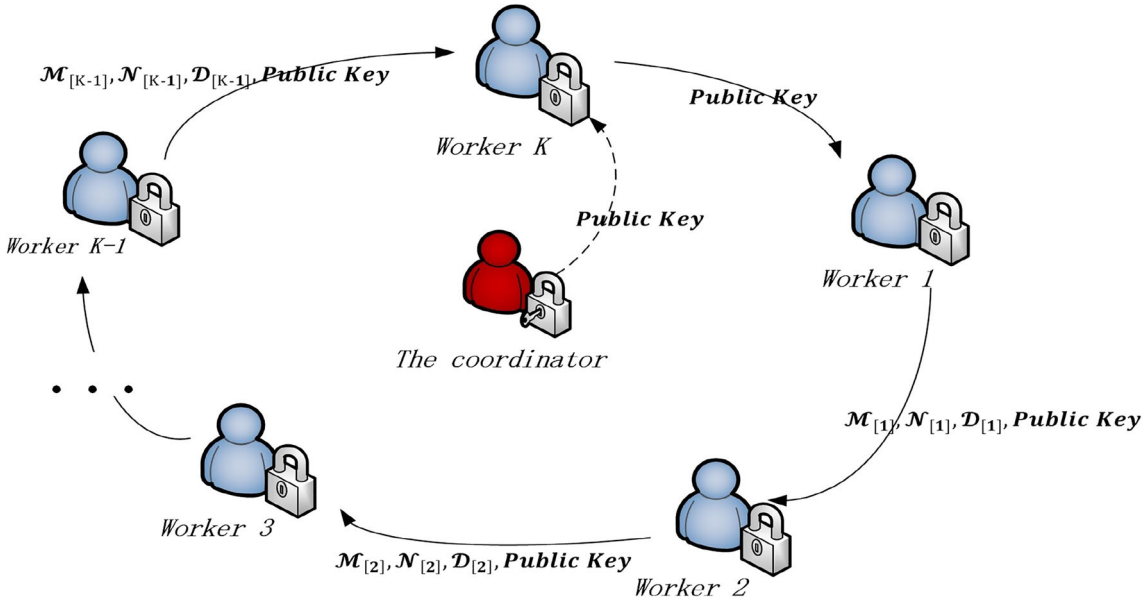

Fig. 4 The aggregation process for the VFL-R framework in Steps 1-3

decrypt the local parameters in the T-th iteration. Even if the coordinator obtains the actual values of the local parameters, it is different from inferring the raw data under Assumption 2.

6 Experiment setting

All experiments simulate the VFL scenario utilizing Python 3.8.5 on an Intel Core E5-2640 CPU 2.40GHz. The data were partitioned vertically into four non-overlapping parties with a nearly equal number of features. We randomly selected 70% of the samples as the training data, and the remaining were employed as testing data.

6.1 Problem

The following experiment focuses on the binary classifications problem and utilizes the logistic regression [31, 32] scheme written as:

$$f(w, y) \triangleq \frac{1}{n} \sum_{i=1}^{n} \log \left[1 + \exp(-wy)\right], \tag{11}$$

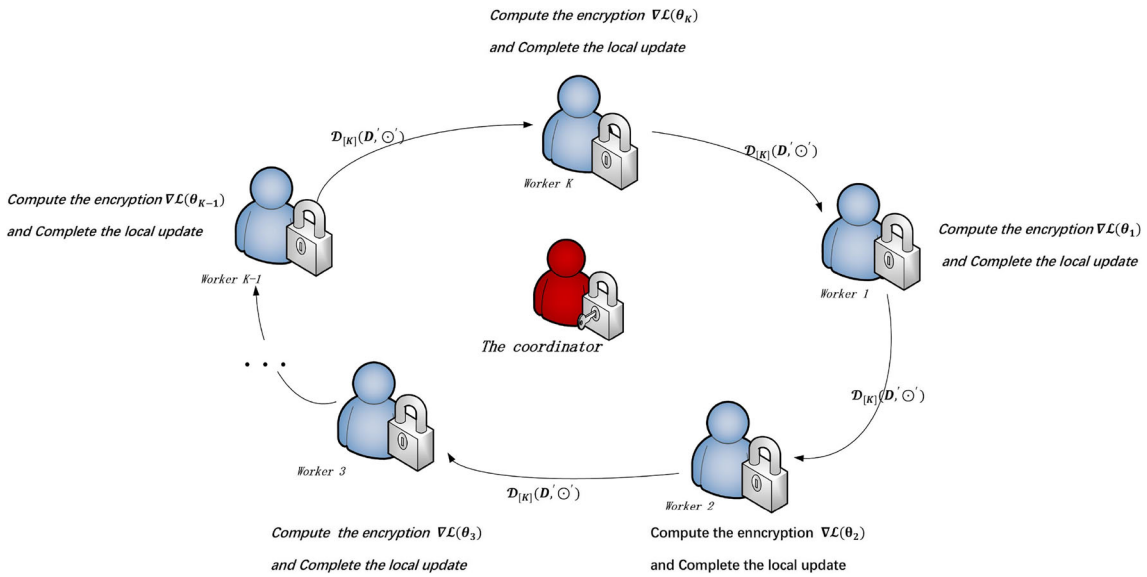

Fig. 5 The local updating process for the VFL-R framework in Step 4

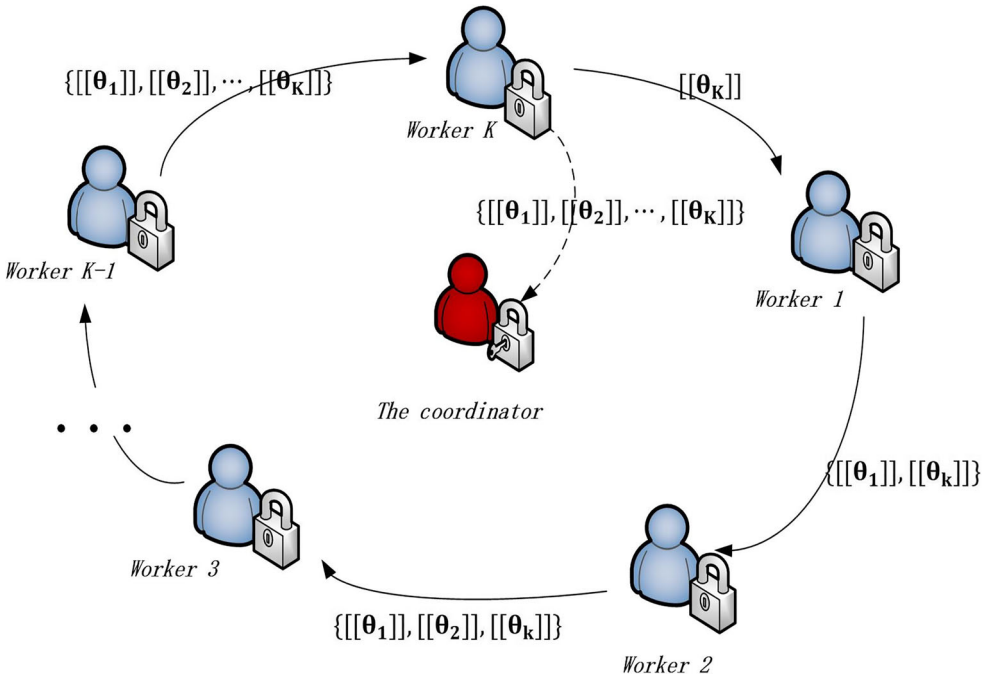

Fig. 6 The decryption of local parameters for the VFL-R framework in Step 6

where $w = \mathbf{x}_{[i,k]}\theta_i$ and $y = \{-1, 1\}$. We add the ℓ_2-norm regularized risk written as $R(\Theta) = \frac{1}{2}\|\Theta\|_2^2$ to avoid overfitting. Meanwhile, we use the second-order Taylor approximations for the logistic loss function to solve the non-linear problem [33]. The model function can be written as:

$$f(w, y) \approx \frac{1}{n}\sum_{i=1}^{n}\left(\log 2 - \frac{1}{2}wy + \frac{1}{8}w^2\right) + \frac{\lambda}{2}\|\Theta\|_2^2. \quad (12)$$

The gradient with respect to θ_k is:

$$\nabla_{[k]f(w,y)} \approx \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{4}w - \frac{1}{2}y\right)\mathbf{x}_{[i,k]} + \lambda\theta_k. \quad (13)$$

6.2 Benchmark datasets

We evaluate our framework's performance on benchmark datasets with various numbers of samples and features. Specifically, we select four datasets from the UCI datasets [34]: the lonosphere, statlog (Heart), sonar, and breast cancer wisconsion diagnostic (WDBC) datasets. The sample and feature numbers are listed in Table 2, while the values of each feature are standardized into [0, 1].

6.3 The algorithm

Affected by the limited calculation power, using a public key to perform repeated PHE operations becomes hard. To solve such problem, we set t as a fixed training period in Algorithm 1. Specifically, the value of t can attend the maximum number of PHE operations. Algorithm 2 gives the system of the VFL-R framework after T (T>t) iterations. In fact, the total period T will be divided into many small periods t. Each party will execute the VFL-R framework in the divided period.

7 Varying experiment setting

The performance assessment metrics are the convergence performance and the classification results on the benchmark datasets. Moreover, we explore the effects of various learning rates α and tuning parameters λ.

7.1 Varying learning rates α

In our experiment, it is hard to get the loss function curves due to the limits of the encrypted local parameters. Thus, we assume that all workers save the results of the encrypted parameters and jointly compute the loss function with the coordinator ($t = 1$). The loss function curves of the VFL-R framework under various learning rates are presented in

Table 2 Statistic of benchmark datasets

Dataset	#Samples	#Features
lonosphere	351	34
statlog (Heart)	270	13
sonar	208	60
WDBC	569	30

Figs. 7, 8, 9 and 10. These figures highlight that the loss function curves have a consistent overall trend regardless of the learning rates. Moreover, from $\alpha = 0.01$ to $\alpha = 0.3$, the convergence speed of VFL-R is improved, and the classification results under various learning rates are reported in Table 3. When $\alpha = 0.1$, the VFL-R framework achieves the best classification, while overall, the learning rate affects the classification performance. Therefore, the learning rate value must be appropriately tuned rather than selecting a large value.

Input: ClientA, ClientB, ClientC, ClientD, The coordinator
Output: $\theta_A, \theta_B, \theta_C, \theta_D$
initialization: encryption pairs, λ, α, t
if $i == 1$ **then**
  One:
  The coordinator: sends encryption key to the ClientA
  The ClientA: receives the public key, computes the encrypted results and sends to the ClientB
  ……
  The ClientD: receives the public key, computes the encrypted loss and the encrypted intermediate results for the local updating, sends to the ClientA
else
  for i *in* $\{2, \ldots, t\}$ **do**
    Two:
    The ClientA: receives the encrypted intermediate results for the local updating, local updating, sends the the encrypted intermediate results to the ClientB
    ……
    The ClientD: receives the intermediate results for the local updating, local updating
    The ClientA: computes the encrypted intermediate results for the local updating, sends to the ClientB
    ……
    The ClientD: computes the encrypted intermediate results for the local updating, sends to the ClientA
    if $i == t$ **then**
      Three:
      perform the Step 6 according to the VFL-R framework, get the decrypted parameters
    end
  end
end

Algorithm 1 The VFL-R framework with the t training period.

Input: ClientA, ClientB, ClientC, ClientD, The coordinator
Output: $\theta_A, \theta_B, \theta_C, \theta_D$
initialization: encryption pairs, λ, α, t, T
for i *in* $\{1, 2, \ldots, T\}$ **do**
  if $i\%t == 1$ **then**
    ClientA, ClientB, ClientC, ClientD and the coordinator perform the Step One according to the Algorithm 1
  end
  if $i\%t == 0$ **then**
    ClientA, ClientB, ClientC, ClientD and the coordinator perform the Step Two and Step Three according to the Algorithm 1
  else
    ClientA, ClientB, ClientC, ClientD perform the Step Two according to the Algorithm 1
  end
end

Algorithm 2 The VFL-R framework with the t iterations.

7.2 Varying tuning parameters λ

For this case, we set the learning rate to 0.1 and alter the tuning parameters. The loss function curves involving various tuning parameters λ are illustrated in Figs. 11, 12, 13 and 14 demonstrating that the loss function curves have a similar convergence trend. The classification results of VFL-R with different tuning parameters are reported in Table 4, indicating that when λ increases from 0.1 to 0.9, VFL-R achieves the high classification performance of 84.69% - 85.05% on the lonosphere dataset, 86.11% - 86.33% on the statlog (Heart) dataset, 81.43% - 82.05% on the snoar dataset, and 95.55% - 95.83% on the WDBC dataset.

8 Comparison with different VFL frameworks

At present, existing VFL frameworks pay little attention to the innovation of the communication architecture. In order to better highlight the performance of VFL-R, we challenge it against the VFL [12] and VFB^2 [20] framework in different aspects, including functionality analysis, test accuracy, and communication performance.

8.1 Functionality analysis

Table 5 reports the functional comparison of the above frameworks. Specifically, VFL is based on the C-S communication architecture and can defend against semi-honest attacks to preserve data security. However, such a C-S

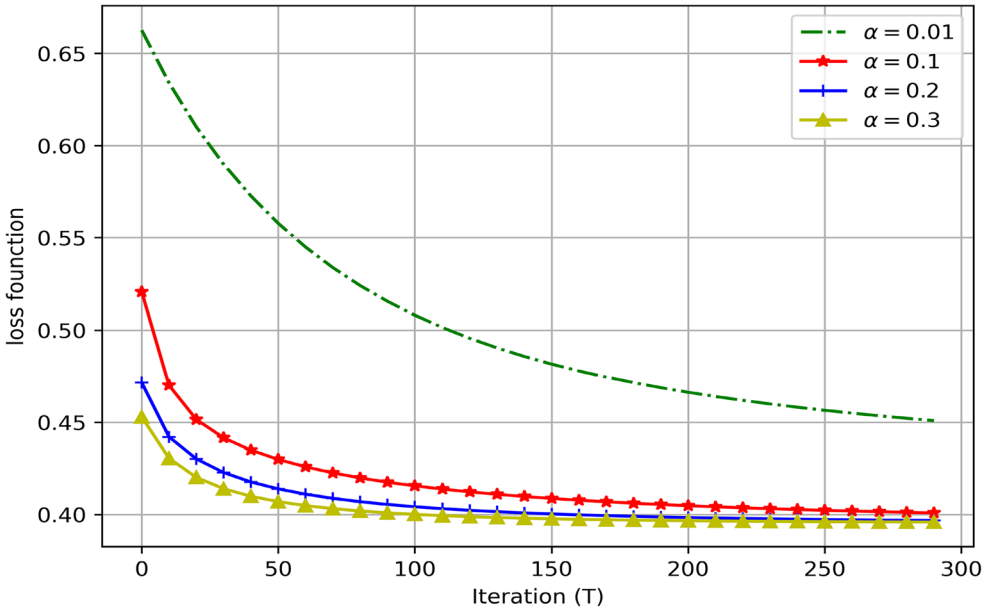

Fig. 7 Loss function curve with various α in the lonosphere dataset, where $\alpha = \{0.01, 0.1, 0.2, 0.3\}$

communication architecture is inefficient, especially when many parties are involved. Regarding the VFB^2 framework, it relies on the tree communication architecture and supports distributed learning. Although the tree communication architecture can significantly reduce the number of communications during the modeling process, its privacy protection can not guarantee high privacy security without using encryption technology [35].

Furthermore, VFL and VFB^2 frameworks impose a significant communication burden for the coordinator, as during the modeling process, the coordinator sends the gradient or other parameters, involving an unnecessary communication cost and a high risk of information disclosure. In contrast, our proposed framework balances the two frameworks and reduces the coordinator's communication burden.

8.2 Test accuracy

To demonstrate the test accuracy of the VFL-R framework, we challenge it against the VFL and VFB^2 frameworks. Furthermore, we test the accuracy gap of different loss functions by considering the non-federated (NonF)

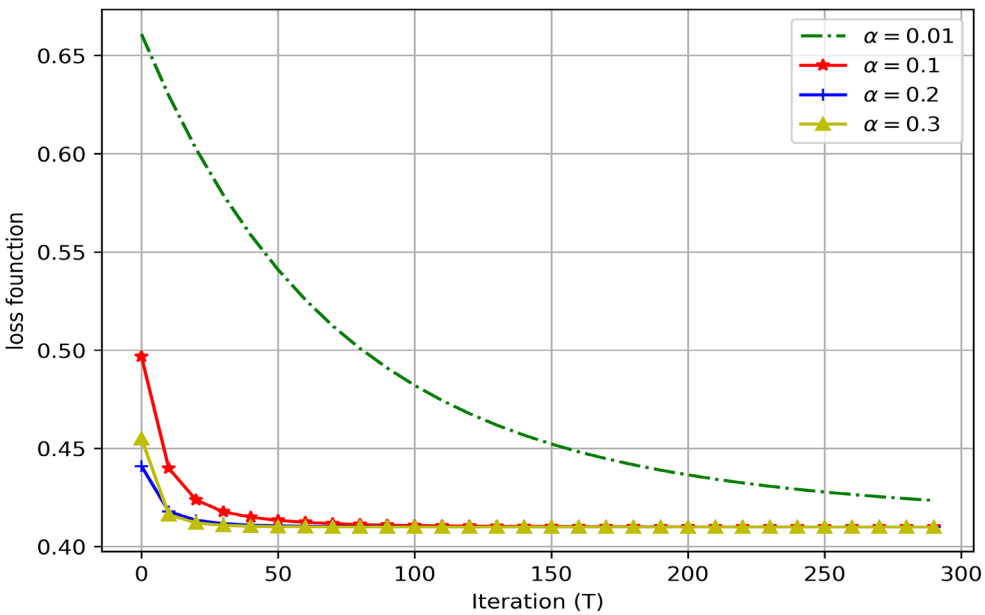

Fig. 8 Loss function curve with various α in the statlog (Heart) dataset, where $\alpha = \{0.01, 0.1, 0.2, 0.3\}$

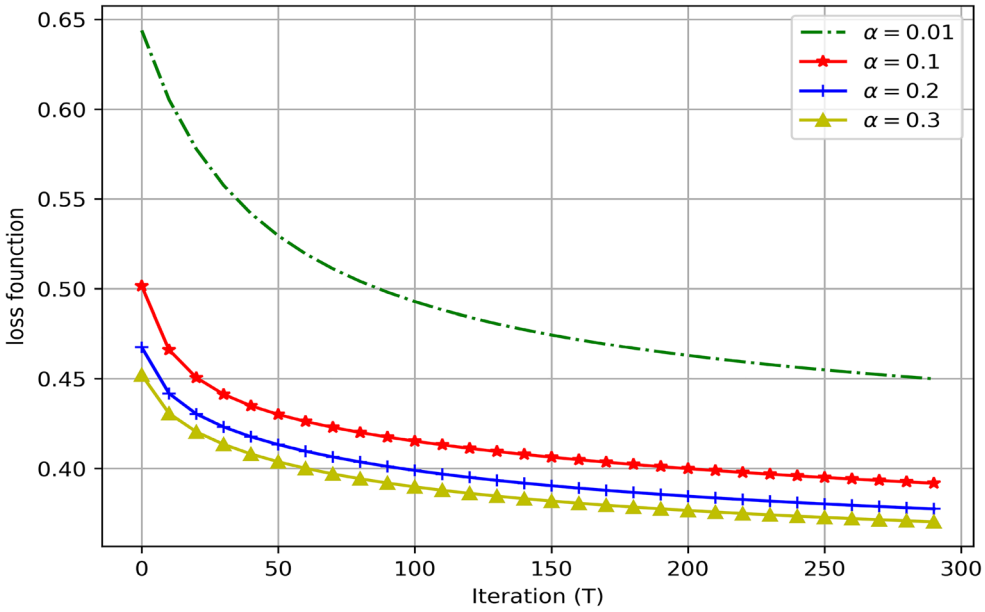

Fig. 9 Loss function curve with various α in the snoar dataset, where $\alpha = \{0.01, 0.1, 0.2, 0.3\}$

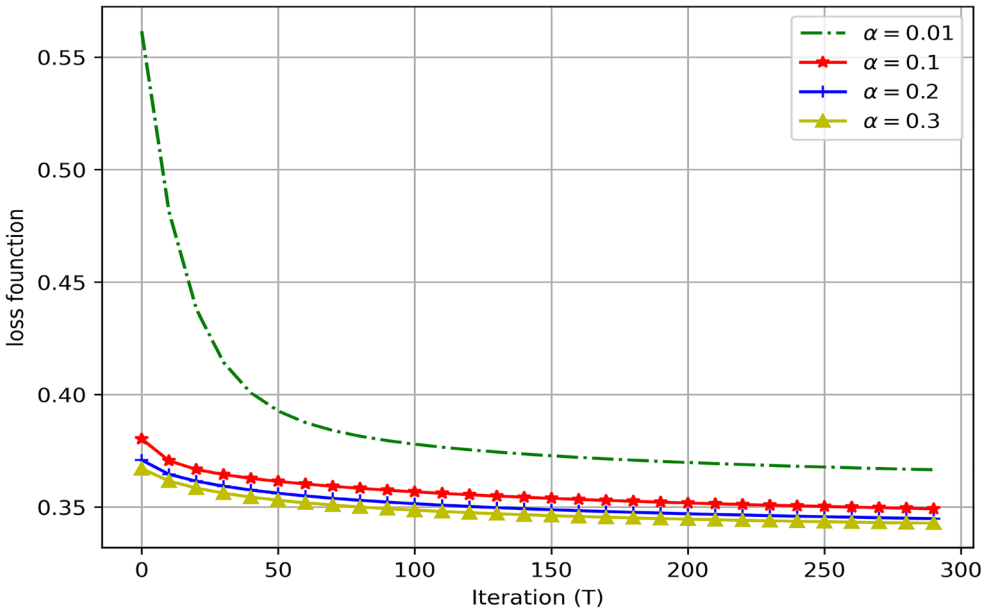

Fig. 10 Loss function curve with various α in the WDBC dataset, where $\alpha = \{0.01, 0.1, 0.2, 0.3\}$

Table 3 Classification results of the VFL-R framework under various learning rates on the benchmark datasets for T=300

Datasets	lonosphere Accuracy	statlog (Heart) Accuracy	snoar Accuracy	WDBC Accuracy
VFL-R ($\alpha = 0.01$)	70.75%	83.95%	78.44%	95.61%
VFL-R ($\alpha = 0.1$)	73.58%	85.60%	81.44%	95.88%
VFL-R ($\alpha = 0.2$)	72.64%	85.18%	80.84%	95.61%
VFL-R ($\alpha = 0.3$)	71.70%	85.18%	81.40%	95.83%

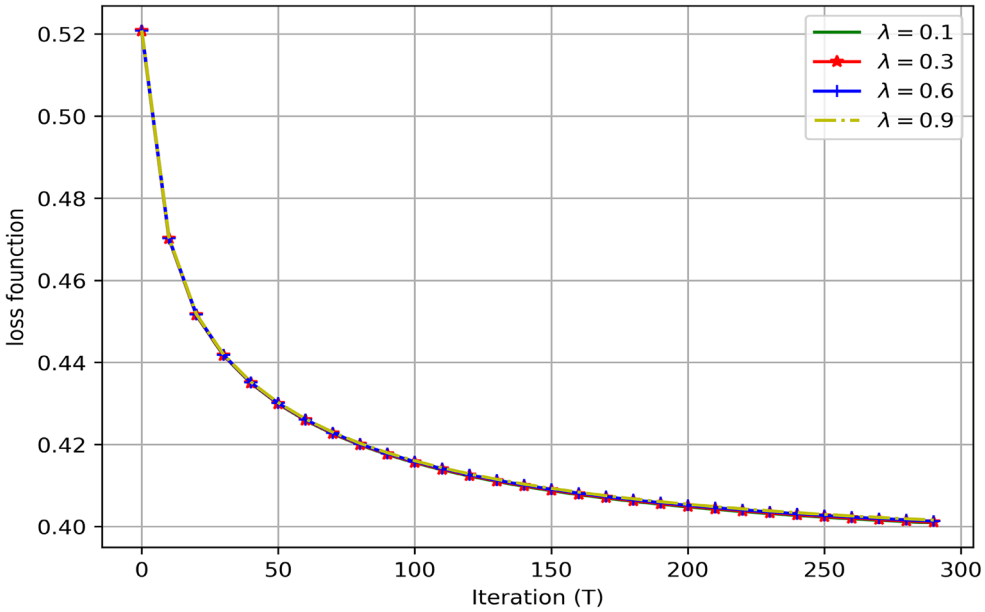

Fig. 11 The loss function curve with different λ in the lonosphere dataset, where λ = {0.1, 0.3, 0.6, 0.9}

experiment where all data are integrated for modeling with the logistic loss function.

Figures 15, 16, 17 and 18 plot the test accuracy of four frameworks based on benchmark datasets. For the Taylor loss function, the VFL-R framework achieves a similar test accuracy to the VFL and VFB^2 frameworks, with the test accuracy of each framework deviating at most by 4% in the lonosphere dataset. Considering the logistic loss function, the VFL-R framework attends the small test accuracy gap.

8.3 Communication cost

We assume that each VFL framework includes N parties. Enc $(\cdot)$ is defined as encryption operation and $|\cdot|$ denotes the data size of each party during the modeling process. w_i and g_i represent the intermediate results for the i-th party, which are used to compute the loss function and gradient, respectively. G is the gradient for modeling process.

For the VFL-R framework. During the aggregation process, each party sends the Enc (w_i, g_i) to the next party and receives the Enc (w_{i-1}, g_{i-1}) from the last party. For the VFL-R framework, the coordinator does not participate in the modeling process. Thus, the communication cost of the third-coordinator is $O(1)$.

For the VFL framework Each party needs to send the Enc (w_i, g_i) to the major party and receives the Enc (w_n, g_n)

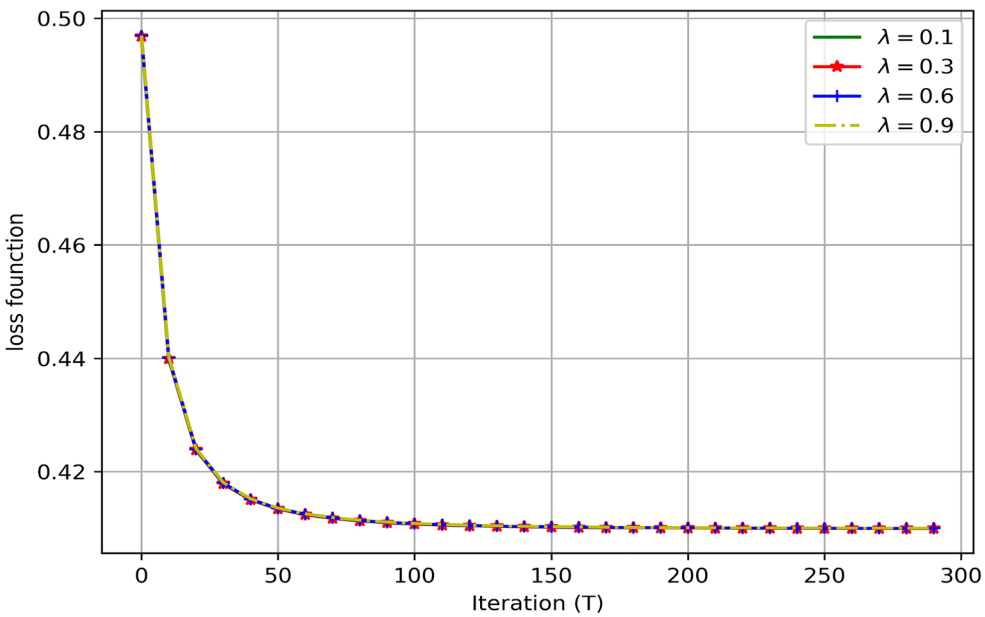

Fig. 12 Loss function curve with various λ in the statlog (Heart) dataset, where λ = {0.1, 0.3, 0.6, 0.9}

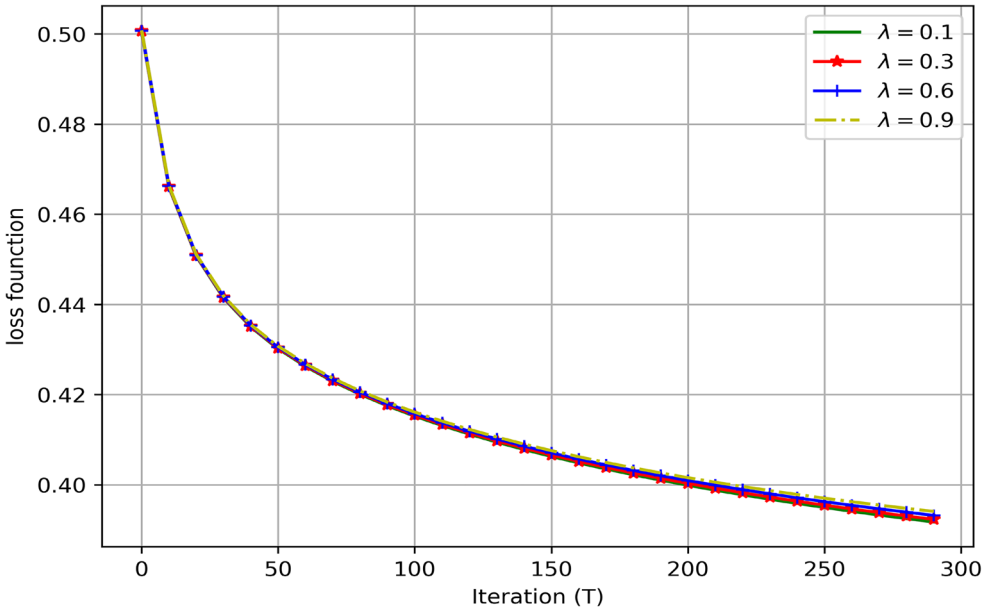

Fig. 13 Loss function curve with various λ in the snoar dataset, where λ = {0.1, 0.3, 0.6, 0.9}

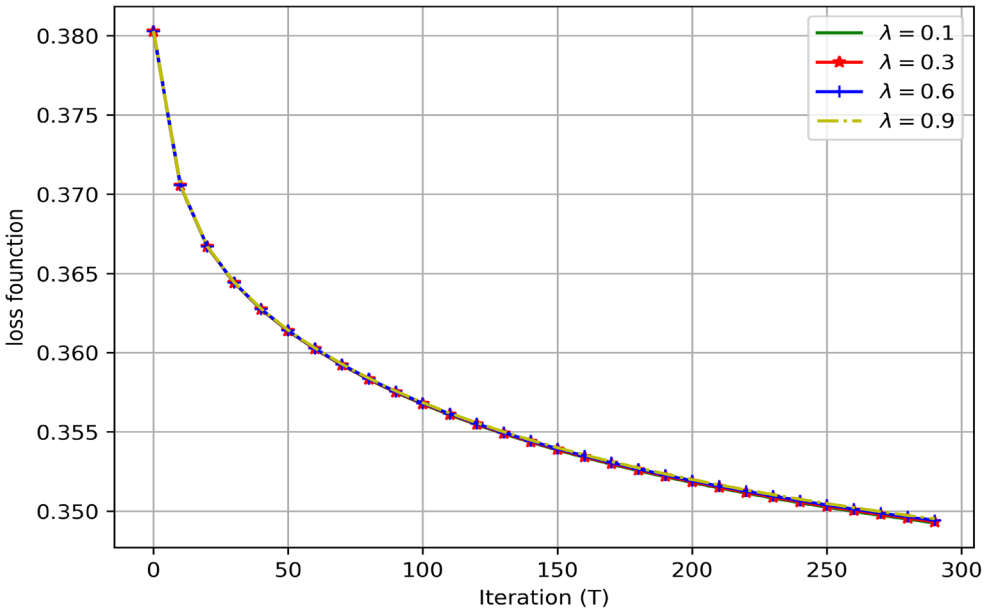

Fig. 14 Loss function curve with various λ in the WDBC dataset, where λ = {0.1, 0.3, 0.6, 0.9}

Table 4 Classification results of the VFL-R framework with various tuning parameters on four datasets for T=300

Datasets	lonosphere Accuracy	statlog (Heart) Accuracy	snoar Accuracy	WDBC Accuracy
VFL-R ($\lambda = 0.1$)	84.69%	86.11%	81.43%	95.83%
VFL-R ($\lambda = 0.3$)	84.66%	86.33%	82.05%	95.55%
VFL-R ($\lambda = 0.6$)	84.66%	86.11%	81.44%	95.80%
VFL-R ($\lambda = 0.9$)	85.05%	86.20%	81.43%	95.80%

Table 5 Comparison analysis of the VFL and VFB^2 frameworks

Framework	VFL	VFB^2	VFL-R
Encryption protection	√	×	√
Decentralization	×	√	√
Defend against semi-honest attacks	√	×	√
Efficient communication architecture	×	√	√
Low coordinator's effect	×	×	√

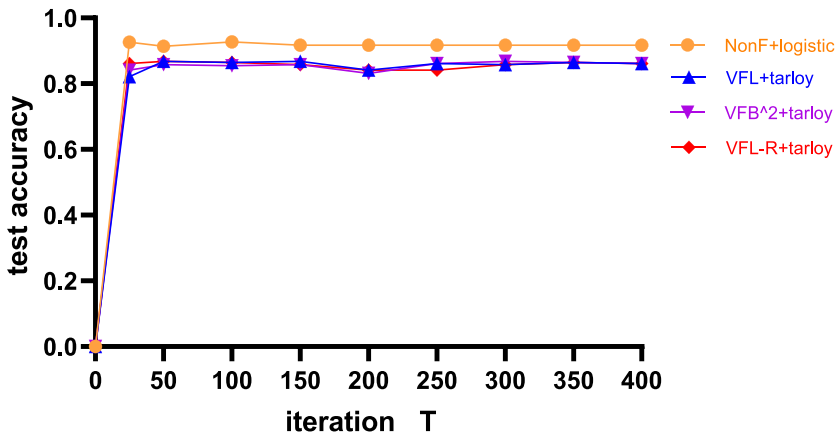

Fig. 15 Test accuracy on the lonosphere dataset with various VFL frameworks, where $T = \{25, 50, 100, 150, 200, 250, 300, 350, 400\}$, $\alpha = 0.1$ and $\lambda = 0.3$

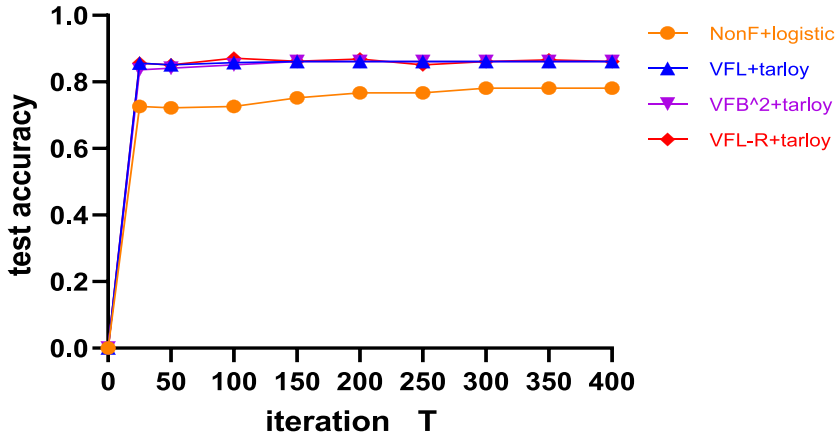

Fig. 16 Test accuracy on the heartstalog dataset with different VFL frameworks, where $T = \{25, 50, 100, 150, 200, 250, 300, 350, 400\}$, $\alpha = 0.1$ and $\lambda = 0.3$

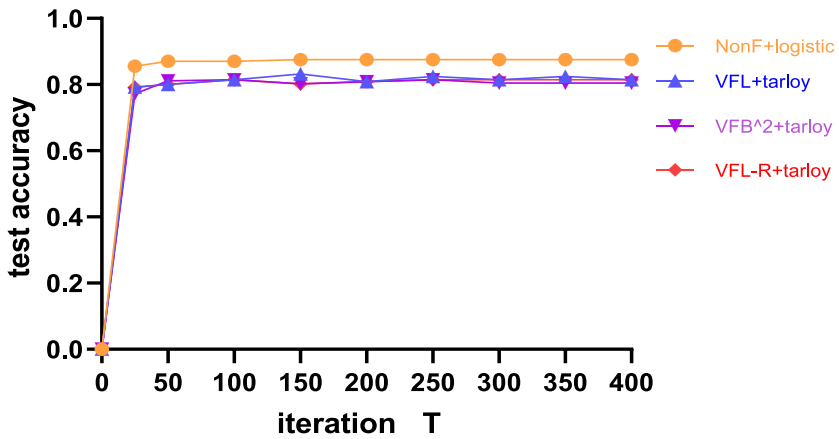

Fig. 17 Test accuracy on the snaor dataset with different VFL frameworks, where $T = \{25, 50, 100, 150, 200, 250, 300, 350, 400\}$, $\alpha = 0.1$ and $\lambda = 0.3$

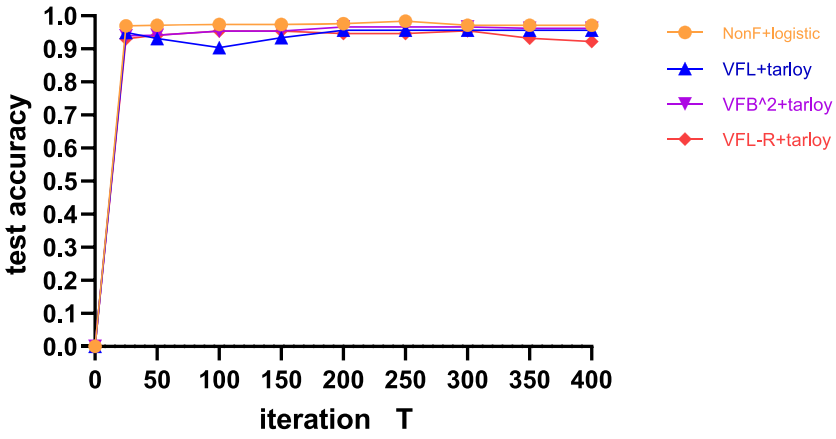

Fig. 18 Test accuracy on the wdbc dataset with different VFL frameworks, where $T = \{25, 50, 100, 150, 200, 250, 300, 350, 400\}$, $\alpha = 0.1$ and $\lambda = 0.3$

Table 6 Communication cost of the VFL-R framework compared with the VFL and VFB^2 frameworks in one communication round

Framework	Each party	The coordinator
VFL-R	$O(\|\text{Enc}(w_i, g_i)\|)+O(\|\text{Enc}(w_{i-1}, g_{i-1})\|)$	$O(1)$
VFL	$O(\|\text{Enc}(w_i, g_i)\|)+\sum_{n=1}^{i-1} O(\|\text{Enc}(w_n, g_n)\|)$	$O(\|\text{Enc}(G)\| \cdot N)$
VFB^2	$O(\|w_i\|) + O(\|g_i\|) + O(\|G\|)$	$O(\|G\| \cdot N)$

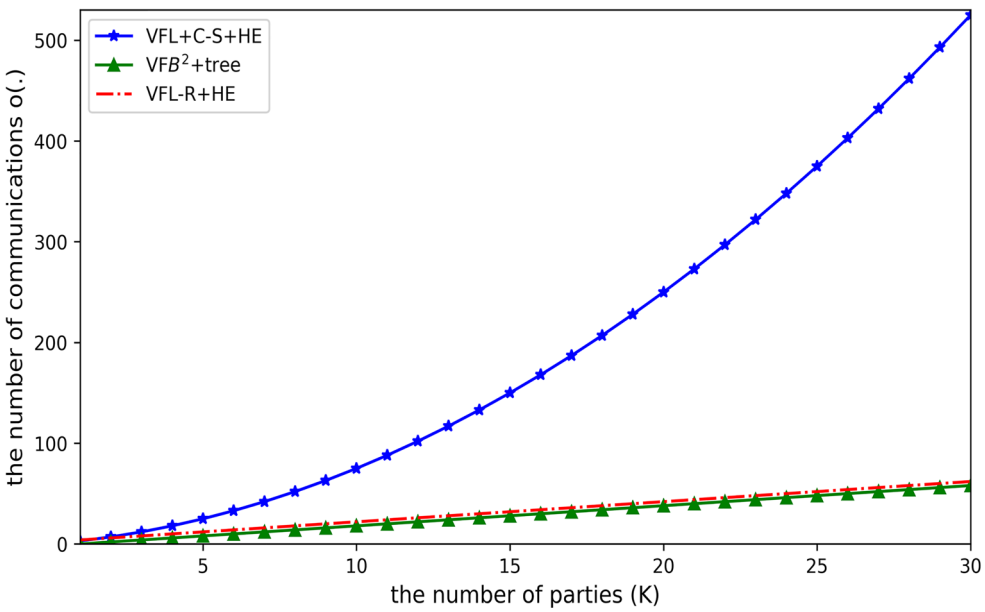

Fig. 19 The number of communications about three VFL frameworks in one communication round

from the other party, where $n = \{1, 2, \ldots, i-1\}$. The coordinator has to send the Enc (G) to each party and therefore the communication cost of the coordinator can be formulated as $O(|\text{Enc}(G)| \cdot N)$.

For the VFB^2 framework Each party sends the (w_i, g_i) to the next party based on the tree communication architecture. Meanwhile, each party will receive the G from the coordinator. Thus, the communication cost of the coordinator is $O(|G| \cdot N)$.

In Table 6, we compare in detail the communication cost of all competitor frameworks, demonstrating that our framework reduces the coordinator's communication cost. Meanwhile, compared to the VFL framework, our method greatly reduces the communication cost for each party.

8.4 The number of communications

Next, we compare the number of communications of the three VFL frameworks per communication round. In Fig. 19, the horizontal axis shows the number of parties K and the vertical axis shows the number of communications in one communication round. It reveals that the VFL framework requires $O(K^2)$ communications as the number of participants increases. However, our proposed framework requires $O(K)$ communications, similar to the VFB^2 framework. Nevertheless, as mentioned in Section 8.1, the VFB^2 framework has poor privacy security, which is not a concern in our framework.

9 Conclusion and future work

This work proposes VFL-R, a new VFL framework that utilizes a ring communication architecture to simplify the intricate communication architecture among each party. In particular, the ring communication architecture reduces the coordinator's communication burden, and our novel communication architecture reduces the number of communications in one communication round. Furthermore, our framework employs HE-based technology to guarantee privacy security. Functionality analysis and extensive experiments demonstrate that VFL-R effectively reduces the communication cost and achieves high accuracy on all benchmark datasets.

Our framework is limited by the necessary assumptions regarding the loss function and gradient. Therefore, future work will aim for improvements utilizing more complicated machine learning approaches or other methods solving these problems. Meanwhile, we will continue our research in designing an efficient framework that further enhances communication performance in VFL scenario.

References

1. Hamon R, Junklewitz H, Sanchez G (2022) Bridging the gap between AI and explainability in the GDPR: towards trustworthiness-by-design in automated decision-making. IEEE Comput Intell Mag 17(1):72–85
2. McMahan B, Moore E, Ramage D (2017) Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th international conference on artificial intelligence and statistics, PMLR, vol 54. pp 1273–1282
3. Raza A, Tran KP, Koehl L, Li SJ (2022) Designing ECG monitoring healthcare system with federated transfer learning and explainable AI. Knowl-Based Syst 326:107763
4. Singh S, Rathore S, Alfarraj O, Tolba A (2022) A framework for privacy-preservation of IoT healthcare data using federated learning and blockchain technology. Future Gener Comput Syst 129:380–388
5. Wang YC, Tian YY, Yin XY, Hei XH (2020) A trusted recommendation scheme for privacy protection based on federated learning. CCF Trans Netw 3(3-4):218–228
6. Liu Y, Ben T, Vincent WZ, Chen K (2020) Federated recommendation systems. Federated Learn Priv Incent 12500:225–239
7. Jiang X, Zhou XB, Jens G (2022) Privacy preserving high-dimensional data collection with federated generative autoencoder. Trauma Surg Acute Ca 2022:481–500
8. Xin BZ, Yang W, Geng YY, Chen S (2020) Private fl-gan: Differential privacy synthetic data generation based on federated learning. In: 2020 IEEE international conference on acoustics, speech and signal processing, pp 2927–2931
9. Paragliola G, Coronato A (2022) Definition of a novel federated learning approach to reduce communication costs. Expert Syst Appl 189:116109
10. Abdellatif AA, Mhaisen N, Mohamed A (2022) Communication-efficient hierarchical federated learning for IoT heterogeneous systems with imbalanced data. Future Gen Comput Syst 128:406–419
11. Feng CS, Liu B, Yu KP, Goudos SK (2022) Blockchain-empowered decentralized horizontal federated learning for 5G-Enabled UAVs. IEEE Trans Ind Informs 18(5):3582–3592
12. Yang Q, Liu Y, Chen TJ, Tong YX (2019) Federated machine learning: concept and applications. ACM T Intel Syst Tec 10:1–19
13. Ou W, Zeng JH, ZJ G, Yan WQ (2020) A homomorphic-encryption-based vertical federated learning scheme for rick management. Comput Sci Inf Syst 17:819–834
14. Hou J, Su M, Fu A, Yu Y (2021) Verifiable privacy-preserving scheme based on vertical federated random forest. IEEE Internet Things 9461157:1–1
15. Li QB, Wen ZY, Wu ZM, Hu SX (2021) A survey on federated learning systems: vision, hype and reality for data privacy and protection. IEEE T Knowl Data En 9599369:1–1
16. Cheng KW, Fan T, Jin LY (2021) Secureboost: a lossless federated learning framework. IEEE Intell Syst 36(6):87–98
17. Zhu HY, Wang R, Jin YC, KT L (2021) PIVODL: Privacy-Preserving vertical federated learning over distributed labels. IEEE Tai 9664283:1–1
18. Zhang X, Ma Z, Wang A (2021) Lstfcfedlear: a LSTM-FC with vertical federated learning network for fault prediction. In: WCMC, p 2021

19. Chen XL, Zhou S, Guan B, Yang K (2021) Fed-EINI: an efficient and interpretable inference framework for decision tree ensembles in vertical federated learning. In: 2021 IEEE international conference on big data (big data), pp 1242–1248
20. Gu B, Xu A, Huo ZY, Deng C (2021) Privacy-preserving asynchronous vertical federated learning algorithms for multiparty collaborative learning. IEEE T Neur Netw Learn 9463409:1–13
21. Amanda CDR, Diego DFA (2021) Faster unbalanced private set intersection in the semi-honest setting. J Cryptogr Eng 11(1):21–38
22. Somchai P (2019) Database secure manipulation based on paillier's homomorphic encryption (DSM-PHE). Int J Interact Mob Technol 13(12):136–151
23. Wang ZW, Zhang Y (2020) Malicious code detection for trusted execution environment based on paillier homomorphic encryption. IEICE Trans Commun 103(3):155–166
24. Yuan W, Hu F, Lu LF (2022) A new non-adaptive optimization method: stochastic gradient descent with momentum and difference. Appl Intell 52(4):3939–3953
25. Tang YJ, Vikram R, Zhang JS, Li N (2022) Communication-efficient distributed SGD with compressed sensing. IEEE Control Syst Lett 6:2054–2059
26. Ritesh N, Nihar BS, Ariel DP (2021) Loss functions, axioms, and peer review. Appl Intell 70:1481–1515
27. Wan L, Han SG (2007) Privacy-preservation for gradient descent methods. In: Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining, pp 775–783
28. Vale T, Stacey T, Mehmet EG, Liu L (2020) Data poisoning attacks against federated learning systems. LNCS 12308:480–501
29. Jere M, Farnan T, Koushanfar F (2021) A taxonomy of attacks on federated learning. IEEE Secur Priv 19(2):20–28
30. Lu SW, Li RH, Chen X, Ma YN (2022) Defense against local model poisoning attacks to byzantine-robust federated learning. Front Comput Sci 16(6):166337
31. Malgorzata L, Jan M, Pawel T (2021) Estimating the class prior for positive and unlabelled data via logistic regression. Adv Data Anal Classif 15(4):1039–1068
32. Cheng SS, Liu JJ, Shi X, Wang K (2022) Rare variant association tests for ancestry-matched case-control data based on conditional logistic regression. Brief Bioinform 23(2):bbab572
33. Pietro M, Alessandro T (2017) Calibration of time-interleaved ADCs via hermitianity-preserving taylor approximations. IEEE Trans Circ Syst II Expr Briefs 64(4):357–361
34. Muhammad T, Chandan G, Ponnuthural NS (2019) Comprehensive evaluation of twin SVM based classifiers on UCI datasets. Appl Soft Comput 83:133–146
35. Jiang X, Zhou XB, Jens G (2022) Comprehensive analysis of privacy leakage in vertical federated learning during prediction. Proc Priv Enhancing Technol 2:263–281

Jialin Li was born in Shandong, China, in 1996. He is currently pursuing the M.S. degree in applied statistics from College of Science, China University of Petroleum. His research interests include federated learning and blockchain.

Tongjiang Yan was born in 1973. He graduated from the Department of Mathematics, Huaibei Coal-Industry Teachers College in 1996. He received the M.S. degree in mathematics from Northeast Normal University, Lanzhou, China, in 1999, and the Ph.D. degree from Xidian University, in 2007. He is the author of more than 80 articles. He holds two patents. He is the Reviewer of the journal the IEEE TRANSACTION INFORMATION THEORY, Finite Field and Their Applications, Cryptography and Communications, and so on. His research interests include cryptography, coding and information theory and federated learning. Dr. Yan was a recipient of General fund of National Natural Science of China.

Pengcheng Ren was born in Shandong, China, in 1995. He received the B.S. degree from School of Mathematics and Statistics, Shandong University of Technology, in 2018. He received the M.S. degree in applied statistics from College of Science, China University of Petroleum, in 2022. His research interests include federated learning and blockchain.