A Privacy-Preserving Scheme for Multi-Party Vertical Federated Learning

FAN Mochan[1], ZHANG Zhipeng[1], LI Difei[1], ZHANG Qiming[2,3], YAO Haidong[2,3]

(1. School of Information & Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China;
2. ZTE Corporation, Shenzhen 518057, China;
3. State Key Laboratory of Mobile Network and Mobile Multimedia Technology, Shenzhen 518055, China)

DOI: 10.12142/ZTECOM.202404012

https://kns.cnki.net/kcms/detail/34.1294.TN.20241122.1626.002.html, published online November 22, 2024

Manuscript received: 2023-11-14

Abstract: As an important branch of federated learning, vertical federated learning (VFL) enables multiple institutions to train on the same user samples, bringing considerable industry benefits. However, VFL needs to exchange user features among multiple institutions, which raises concerns about privacy leakage. Moreover, existing multi-party VFL privacy-preserving schemes suffer from issues such as poor reliability and high communication overhead. To address these issues, we propose a privacy protection scheme for four institutional VFLs, named FVFL. A hierarchical framework is first introduced to support federated training among four institutions. We also design a verifiable replicated secret sharing (RSS) protocol $\binom{3}{2}$-sharing and combine it with homomorphic encryption to ensure the reliability of FVFL while ensuring the privacy of features and intermediate results of the four institutions. Our theoretical analysis proves the reliability and security of the proposed FVFL. Extended experiments verify that the proposed scheme achieves excellent performance with a low communication overhead.

Keywords: vertical federated learning; privacy protection; replicated secret sharing

Citation (Format 1): FAN M C, ZHANG Z P, LI D F, et al. A privacy-preserving scheme for multi-party vertical federated learning [J]. ZTE Communications, 2024, 22(4): 89 - 96. DOI: 10.12142/ZTECOM.202404012

Citation (Format 2): M. C. Fan, Z. P. Zhang, D. F. Li, et al., "A privacy-preserving scheme for multi-party vertical federated learning," *ZTE Communications*, vol. 22, no. 4, pp. 89 - 96, Dec. 2024. doi: 10.12142/ZTECOM.202404012.

1 Introduction

The development of big data has promoted the rise of artificial intelligence, which plays a vital role in modern society. In various fields, such as economics, climate research, personalized services, and medical services, the collection and analysis of data provide important support for researchers. However, with the massive data collection and analysis, some data privacy issues have arisen. As an emerging technology of artificial intelligence, federated learning[1] enables users to use private data for model training locally and share gradients under the coordination of the server, to obtain a higher-precision global model. Federated learning protects user data by eliminating the need for data disclosure.

Vertical federated learning (VFL)[2] enables multiple institutions to train on the same user samples and has received extensive attention from both industry and academia. For example, it facilitates federated analysis of financial data, where information about the same user may come from different banks. However, VFL needs to share user features or intermediate training results among multiple institutions, raising concerns about user data privacy leakage.

Some schemes propose to use secure multi-party computation[3] to address the VFL privacy leakage issue[4 - 6]. NI et al.[7] proposed FedVGCN, a federated graph convolutional network (GCN) learning paradigm suitable for node classification tasks. Participants exchange intermediate results under homomorphic encryption, thus protecting the data privacy of participants. Similarly, YANG et al.[8] proposed a distributed logistic regression privacy protection scheme using homomorphic encryption and eliminated the third-party coordinator. Although the above schemes guarantee the feature or label privacy of the participating parties, they only support two-party VFL and cannot be applied to multi-party joint training. Therefore, some multi-party VFL privacy protection schemes have been proposed[9 - 10]. LI et al.[9] proposed a tree-based multi-party VFL privacy-preserving system, using homomorphic encryption and differential privacy to protect histogram privacy. However, this scheme requires a large communication overhead, and the model performance suffers due to the addition of noise. XIE et

This work was supported in part by ZTE Industry-University-Institute Cooperation Funds under Grant No. 202211FKY00112, Open Research Projects of Zhejiang Lab under Grant No. 2022QA0AB02, and Natural Science Foundation of Sichuan Province under Grant No. 2022NSFSC0913.

al.[10] proposed a multi-party VFL privacy protection scheme MP-FedXGB using secret sharing. Each participant directly performs model training on the secret shares, resulting in a large communication overhead. These multi-party VFL schemes not only require a high communication overhead but also have poor reliability. Participants are not allowed to exit. Once a participant exits, model training will be interrupted.

Taking into account the issues of poor reliability and high communication overheads in existing schemes, we propose a privacy-preserving VFL scheme that supports four-party federated training, named FVFL. This scheme supports four institutions for VFL training, consisting of three institutions with intersection feature sets (passive and unlabeled) and one institution with a different feature set (active and labeled) from these three institutions. First, the three passive parties utilize the proposed repeated secret-sharing algorithm to realize the private summation of intersection features under overlapping user sets. The proposed repeated secret-sharing algorithm satisfies the requirement that the feature sums of three passive parties can still be recovered when one passive party quits the secret reconstruction process. Then, any of the three passive parties can perform model training with the active party to realize the function of four-party federated training. This ensures a low communication overhead and high reliability during VFL training in multiple institutions.

The contributions of this paper include the following aspects:

1) We propose an effective four-party VFL federated training framework, which reduces the system communication overhead through a hierarchical structure, and any of the three passive parties cooperates with the active party to achieve VFL training on four-party data.

2) We design $\binom{3}{2}$-sharing, a verifiable replicated secret sharing (RSS) protocol. Any two parties can cooperate to recover the sum of the three-party features, and the protocol only requires additional operations with a low computational overhead.

3) Our theoretical analysis proves the security of the scheme. Experimental results verify the advantages of FVFL in terms of model performance and communication overhead.

The remainder of this paper is organized as follows. Section 2 introduces the work related to VFL privacy protection. Section 3 provides an overview of the FVFL system model, threat model, and security requirements. Section 4 presents the FVFL construction details. Section 5 proves the safety of the proposed FVFL, and Section 6 demonstrates the effectiveness of FVFL through experiments. Finally, the paper is concluded in Section 7.

2 Related Work

VFL enables multiple institutions to conduct model training on the same user samples in a distributed manner. While this approach has received widespread attention in both academia and industry, concerns about privacy leakage among participants have become increasingly prominent. Some schemes[11-12] propose the use of cryptography technology to encrypt intermediate results to protect data privacy. However, they all require a third party to act as a coordinator for scheduling the training process. FANG et al.[13] proposed a VFL privacy protection scheme that cancels the third party and uses secret sharing to avoid leakage of intermediate information in the training process, thereby enabling safe model prediction. However, the solutions mentioned above only support VFL training between two institutions. Obviously, multi-party VFL would better meet actual needs.

Therefore, several VFL privacy-preserving schemes supporting multi-party joint training have been proposed[14-15]. WU et al.[16] proposed a vertical decision tree scheme to preserve the privacy of intermediate information. In this approach, each participant first uses homomorphic encryption to generate statistical information and then employs secret sharing to determine the best split of tree nodes. Finally, the secret is reconstructed, and each participant updates the model with encrypted data. However, this scheme requires secret segmentation and transmission of the homomorphic encrypted ciphertext, resulting in a significant communication overhead. HUANG et al.[17] proposed a multi-party VFL privacy protection scheme designed for generalized linear models. Participants first segment the gradients using secret sharing algorithms, and then homomorphically encrypt the segmented gradients and propagate them to each other. Ultimately, the receiver decrypts and reconstructs the gradient, thereby achieving gradient privacy protection. However, this scheme has a high communication overhead and is limited to simple linear models.

In addition, the schemes mentioned above face the issue of poor reliability. If any participant is accidentally disconnected, the training will be interrupted. Therefore, it is essential to design a multi-party VFL privacy protection scheme with high reliability and a limited communication overhead.

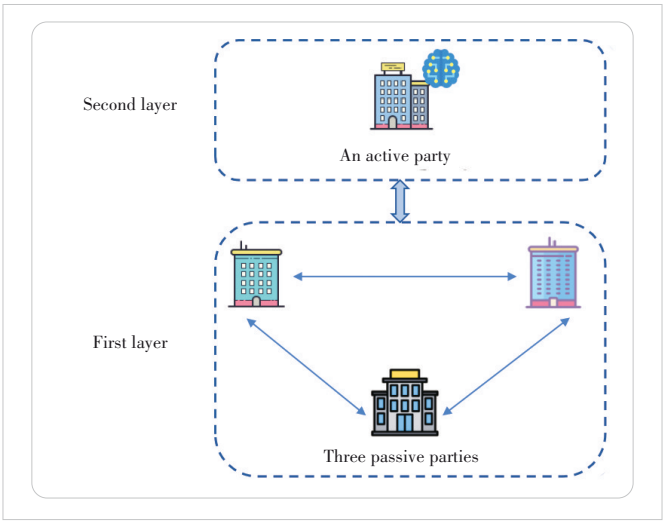
3 Problem Description

In this section, we outline the FVFL system model, analyze potential security threats, and then describe the security requirements.

3.1 System Model

The FVFL system model mainly includes two types of entities: one active party and three passive parties. The architecture of the FVFL system is shown in Fig. 1.

1) Active participant: The active party is the organization (such as an operator) that owns the label among the four organizations and has a different set of features compared to the three passive parties. The active party plays a leading role in the four-party VFL training, denoted as P_0. After the three passive parties perform $\binom{3}{2}$-sharing, P_0 and any of the parties use



▲Figure 1. Architecture of the proposed FVFL framework

the data of the four institutions for model training. Furthermore, the intermediate results are encrypted using a homomorphic encryption algorithm to ensure data privacy.

2) Passive participants: Three institutions with intersection features (such as three banks) are passive parties of FVFL and do not have labels, denoted as $P_i, i \in \{1, 2, 3\}$. Before conducting model training with P_0, the three passive parties utilize the proposed verifiable RSS protocol $\binom{3}{2}$-sharing for feature sharing. Through this protocol, P_i can obtain the sum of the intersection features of users overlapping in three institutions. Then, according to the hierarchical structure, any party in P_i performs model training with the active party P_0 of another layer.

3.2 Threat Model

Participants may not be completely trustworthy and could exhibit malicious behavior. In addition, there may be external attackers on the network that want to steal private data from participants or hinder model training. Next, we analyze the potential threats faced by VFL participants and the network.

First, assuming that the passive parties P_i are malicious in the RSS process, they may transmit illegal information. For example, during the secret split phase, they may distribute the wrong secret shares to other parties, making it impossible to reconstruct the secret. During the secret reconstruction phase, P_i may transmit the wrong secret shares to the partner or maliciously exit the reconstruction process, causing the other party to fail to recover the correct secret.

Second, during the model training process with the active party P_0, the passive party P_i will honestly conduct the training, but wants to steal the other party's private data from the intermediate results.

Finally, we assume that there is a malicious attacker in the network, which wants to steal secret shares or intermediate results by listening to the channel to infer the private data of the participants.

3.3 Security Requirements

Considering the possible security threats of FVFL, the following security requirements should be met.

1) Confidentiality: During the process of secret sharing, malicious P_i or attackers may infer the secrets of other participants from the received or stolen secret shares. In addition, P_0 and P_i may infer each other's privacy data through intermediate results of interaction during model training. Therefore, FVFL should be able to ensure the confidentiality of the participant data.

2) Reliability: When P_i is performing secret reconstruction or model training with P_0, it may subjectively or passively exit the system, which makes it impossible to reconstruct the secret or directly interrupt the model training. Therefore, it is necessary to ensure that even if any participants exit, FVFL can still reconstruct the secret and perform model training normally.

3) Verifiability: Malicious P_i may distribute or transmit wrong secret shares to other parties during secret splitting or reconstruction, making it impossible to recover the secret. Therefore, FVFL needs to ensure that P_i transmits the correct share of the secret so that the secret can be reconstructed.

4 Construction Details

In this section, we introduce the construction details of the proposed FVFL, and the symbols involved in this paper are summarized in Table 1.

4.1 $\binom{3}{2}$-Sharing Secret Split for FVFL

Three passive parties P_i generate random numbers a_1, a_2 and a_3 through pseudo-random functions[18], satisfying $a_1 + a_2 + a_3 = 0$, where a_1, a_2 and a_3 are held and saved by P_1, P_2 and P_3, respectively. Assume that P_1, P_2 and P_3 hold secrets x, y, and z, respectively (the values of each intersection feature at the three passive parties). They split the secret through a secret split algorithm and distribute the secret shares to the other two passive parties.

As shown in Algorithm 1, the secret split algorithm includes three steps. First, each passive party splits its respective secrets x, y, and z into multiple secret shares. Then, the

▼Table 1. Description of the symbols involved in this paper

Symbol	Description
P_0	Active participant (holding labels)
$P_i, i \in \{1, 2, 3\}$	Passive participants
x, y, z	Secrets held by P_1, P_2 and P_3 respectively
x_i, y_i, z_i	Secret shares
$H(\cdot)$	One-way hash function

passive party hashes the secret shares. Finally, the three passive parties distribute secret shares and corresponding hash values to each other. This $\binom{3}{2}$-sharing protocol ensures the privacy of the passive party's secret and the verifiability of the secret share by introducing random numbers and hash operations.

Algorithm 1. $\binom{3}{2}$**-Sharing Secret Split**

Input: a_1, a_2, a_3, x, y, and z
Output: Secret shares and corresponding hash values of three passive parties
1: P_1 splits secret x: $x = x_1 + x_2 + x_3$.
2: P_2 splits secret y: $y = y_1 + y_2 + y_3$.
3: P_3 splits secret z: $z = z_1 + z_2 + z_3$.
4: P_1 calculates $x\hat{} = H(x + a_1)_{p_1}$, $H(x_1)_{p_1}$, $H(x'_2)_{p_1}$, and $H(x_3)_{p_1}$, where $x'_2 = x_2 + a_1$.
5: P_2 calculates $y\hat{} = H(y + a_2)_{p_2}$, $H(y_1)_{p_2}$, $H(y'_2)_{p_2}$, and $H(y_3)_{p_2}$, where $y'_2 = y_2 + a_2$.
6: P_3 calculates $z\hat{} = H(z + a_3)_{p_3}$, $H(z_1)_{p_3}$, $H(z'_2)_{p_3}$, and $H(z_3)_{p_3}$, where $z'_2 = z_2 + a_3$.
7: P_1 sends (x_1, x'_2), $H(x_1)_{p_1}$, $H(x'_2)_{p_1}$, $H(x_3)_{p_1}$ and $x\hat{}$ to P_2, and sends (x'_2, x_3), $H(x_1)_{p_1}$, $H(x'_2)_{p_1}$, $H(x_3)_{p_1}$ and $x\hat{}$ to P_3.
8: P_2 sends (y_1, y'_2), $H(y_1)_{p_2}$, $H(y'_2)_{p_2}$, $H(y_3)_{p_2}$ and $y\hat{}$ to P_1, and sends (y'_2, y_3), $H(y_1)_{p_2}$, $H(y'_2)_{p_2}$, $H(y_3)_{p_2}$ and $y\hat{}$ to P_3.
9: P_3 sends (z_1, z'_2), $H(z_1)_{p_3}$, $H(z'_2)_{p_3}$, $H(z_3)_{p_3}$ and $z\hat{}$ to P_1, and sends (z'_2, z_3), $H(z_1)_{p_3}$, $H(z'_2)_{p_3}$, $H(z_3)_{p_3}$ and $z\hat{}$ to P_2.
Return: Secret shares and corresponding hash values of three passive parties.

4.2 $\binom{3}{2}$-Sharing Secret Reconstruction for FVFL

Once each passive party receives the secret shares of the other two passive parties, it can recover the sum of the three-party secrets by executing the secret reconstruction algorithm. At this stage, even if a passive party actively or passively withdraws, the algorithm can still run normally. Assuming that P_3 is accidentally disconnected during the secret reconstruction process, the following describes how P_1 and P_2 reconstruct the sum of the three-party secrets.

As shown in Algorithm 2, the secret reconstruction algorithm includes three steps. First, P_1 and P_2 send the different secret shares to each other. Then, P_1 and P_2 respectively use the hash function to determine the consistency of the received secret shares. Finally, P_1 and P_2 carry out secret reconstruction, respectively, to obtain the sum of the secrets of the three parties.

Algorithm 2. $\binom{3}{2}$**-Sharing Secret Reconstruction**

Input: $x + a_1, (z_1, z'_2), y + a_2$ and (z'_2, z_3).
Output: $x + y + z$.
1: P_1 sends $x + a_1$ and (z_1, z'_2) to P_2.
2: P_2 sends $y + a_2$ and (z'_2, z_3) to P_1.
3: P_1 calculates $H(y + a_2)$, $H(z_1)$, $H(z'_2)$, and $H(z_3)$ and determines whether $H(y + a_2) = y\hat{}$, $H(z_1) = H(z_1)_{p_3}$, $H(z'_2) = H(z'_2)_{p_3}$, and $H(z_3) = H(z_3)_{p_3}$ are valid.
4: P_2 calculates $H(x + a_1)$, $H(z_1)$, $H(z'_2)$, and $H(z_3)$ and determines whether $H(x + a_1) = x\hat{}$, $H(z_1) = H(z_1)_{p_3}$, $H(z'_2) = H(z'_2)_{p_3}$, and $H(z_3) = H(z_3)_{p_3}$ are valid.
5: If all the equations hold, P_1 and P_2 calculate $x + a_1 + y + a_2 + z_1 + z'_2 + z_3 = x + y + z$ locally, respectively.
Return: $x + y + z$.

4.3 Homomorphic Encryption for FVFL

After the three passive parties perform the feature summation, any of them can cooperate with the active party P_0 to perform model training to achieve four-party VFL training. Here, the homomorphic encryption algorithm is used to ensure the privacy of the intermediate results of the interaction. The specific execution process is as follows.

- P_0 first uses the label to calculate the first-order derivative g_i and the second-order derivative h_i of the gradient, and then uses the homomorphic encryption algorithm to encrypt g_i and h_i, followed by sending the encrypted $<g_i>$ and $<h_i>$ to P_i.
- After receiving the ciphertexts $<g_i>$ and $<h_i>$, P_i uses them to calculate the local gradient histogram and sends the gradient histogram to P_0.
- P_0 decrypts the gradient histogram sent by P_i, finds the optimal split point, and then sends it to P_i.
- P_0 and P_i determine which party has the best split point and then receive the sample division result of the party with the best split point.
- Both parties update the index between samples and tree nodes, as well as their respective tree models.

5 Security Analysis

In this section, we analyze the security and reliability of the proposed FVFL scheme.

Theorem 1. Although the passive party P_i is malicious, it must share a clear secret during RSS. In addition, if malicious P_i transmits incorrect secret shares, other passive parties will discover it.

Proof: According to the design of the secret sharing protocol, each P_i will transmit the secret shares and their corresponding hash values to other passive parties during the secret distribution phase. For example, in the secret distribution phase, P_1 must transmit (x_1, x'_2), $H(x_1)_{p_1}$, $H(x'_2)_{p_1}$, $H(x_3)_{p_1}$ and $x\hat{}$ to P_2, as well as (x'_2, x_3), $H(x_1)_{p_1}$, $H(x'_2)_{p_1}$, $H(x_3)_{p_1}$ and $x\hat{}$ to P_3. If P_1 is compromised, it may transmit illegal $x^{\circ} + a_1$ to P_2 or P_3 during the secret reconstruction phase. At this

time, P_2 or P_3 can judge that $H(x^{\circ} + a_1) = x^{\wedge}$ is not established through the hash, and thus perceive the malicious behavior of P_1. Therefore, even if P_1 is malicious, it must share a clear secret to make $H(x + a_1) = H(x + a_1)_{p_1}$ valid. Similarly, P_2 and P_3 can verify the legitimacy of the secret shares transmitted by P_1 by judging whether $H(x_1) = H(x_1)_{p_1}$, $H(x'_2) = H(x'_2)_{p_1}$, etc. are established.

Theorem 2. The proposed FVFL scheme will not leak the private data of any participant during the secret sharing and model training stages.

Proof: Firstly, assume that P_1 and P_2 cooperate to reconstruct the sum of the three-party secrets. In the secret reconstruction phase, P_1 and P_2 need to transmit $x + a_1$ and $y + a_2$ to each other. According to the protocol settings, P_1 holds $x + a_1$ and (z_1, z'_2), and P_2 holds $y + a_2$ and (z'_2, z_3). They can calculate the secret sum of the three parties $(x + a_1) + (y + a_2) + (z_1 + z'_2 + z_3) = x + y + z$, respectively. Because P_1 does not know the random numbers a_2 and a_3 of P_2 and P_3, and P_2 does not know the random numbers a_1 and a_3 of P_1 and P_3, they can only obtain $y + a_2$ and $z + a_3$, respectively, but fail to obtain y or z, thus ensuring the privacy of the passive party's secret values.

Secondly, when the active party P_0 conducts model training with any of the three passive parties P_i, P_0 performs homomorphic encryption on the intermediate results to ensure data privacy. Moreover, P_i uses the sum of three-party data to perform model training, and it will not leak the data privacy of a single passive party.

Theorem 3. If one or two passive parties exit during secret reconstruction or model training, the proposed FVFL can still run as usual.

Proof. Assuming P_1 exits the secret reconstruction phase maliciously or passively, the remaining P_2 and P_3 can still successfully reconstruct the sum of the three-party secrets. This is because P_2 and P_3 not only hold their respective secrets y and z, but also hold P_1's secret shares (x_1, x'_2) and (x'_2, x_3), respectively. Therefore, P_2 and P_3 can collaborate to calculate $(y + a_2) + (z + a_3) + (x_1 + x'_2 + x_3) = x + y + z$. Furthermore, during model training, since all three passive parties hold the sum of their secrets, P_0 only needs to collaborate with any of the three passive parties to achieve model training.

6 Performance Evaluation

In this section, we evaluate the advantages of the proposed FVFL in terms of communication overhead and performance through extended experiments.

6.1 Experimental Setup

Experiments ran on the Ubuntu 18.04.6 LTS operating system, equipped with 62 GB of memory and an Intel(R) Xeon(R) CPU E5-2650 v4 clocked at 2.20 GHz. We performed SecureBoost model training on the FATE v1.9.0 platform to verify the effectiveness of FVFL. SecureBoost is a decentralized vertical federated learning security tree model based on gradient-boosting decision trees. It supports multi-party cooperation, that is, federated training of multiple unlabeled data holders and one labeled data holder. MP-FedXGB is the benchmark, as it also supports four-party VFL federated training.

The experiments were carried out on two datasets: GiveMeSomeCredit and UCI Credit Card. The data in the GiveMeSomeCredit dataset was used to determine whether users would suffer financial difficulties in the future. It included 150 000 data samples and 10 features. The data in the UCI Credit Card dataset was used to judge whether a person would default, and it included 30 000 data samples and 24 features. We split both datasets into 25 000 training samples and 5 000 testing samples. Features were converted into multi-dimensional vectors using one-hot encoding, distributed to active and passive parties at a ratio of [0.5, 0.5], followed by the passives being distributed at a ratio of [0.2, 0.3, 0.5].

The experiments verified the training and prediction time of FVFL under different hyperparameter settings. The parameters and default settings involved in the experiment are shown in Table 2. Each experiment was performed five times, and the average results were reported.

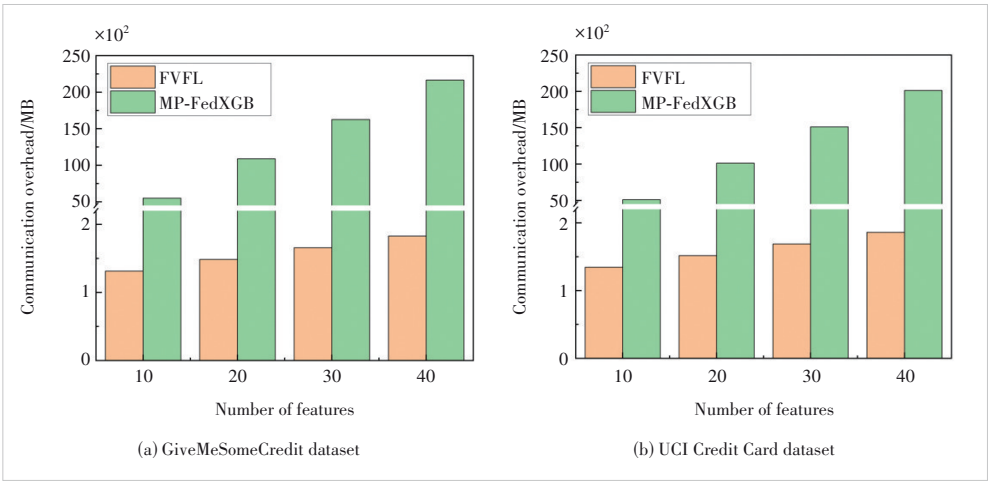
6.2 Communication Overhead Comparisons

In this section, we discuss the communication overhead required for FVFL to perform four-party federated training, including feature secret sharing among the three passive parties, as well as intermediate result interaction between the active party and any passive party. We compare the communication costs of FVFL and MP-FedXGB under the same settings.

1) Varying F. Fig. 2 shows the trends of communication costs for FVFL and MP-FedXGB on the two datasets as F increases. Regardless of which data set, as F increases, the communication overheads of the two schemes will gradually increase. However, compared to MP-FedXGB, the communication overhead of FVFL is significantly lower. For example, when F=40, the amount of communication MP-FedXGB needs to transmit on both datasets is 118.4 times and 108.0 times that of FVFL, respectively. This is because according to the design of the FVFL scheme, only the feature shares need to be transmitted between the three passive parties, and then any of the three passive parties and the active party can perform model training. However, MP-FedXGB must use feature

▼Table 2. Parameters and their default values

Parameter	Description	Value	Default
F	Number of features	{10, 20, 30, 40}	10
T	Number of trees	{3, 4, 5, 6, 7}	3
D	Number of depths	{3, 4, 5, 6, 7}	3
I	Number of data samples	{5k, 10k, 15k, 20k, 25k}	25k



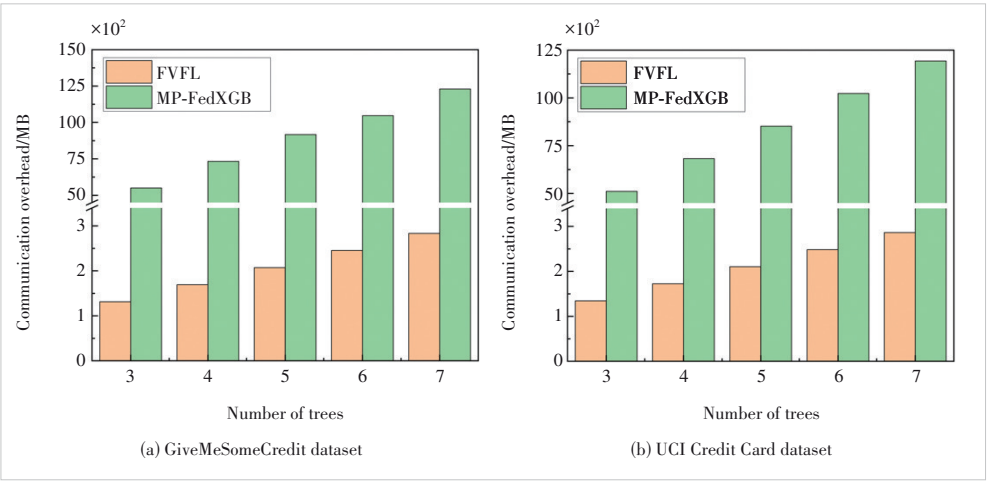
▲Figure 2. Communication overheads for different numbers of features (*F*)

shares to perform model training among four parties, and its communication cost will be higher. The results show that our FVFL is more suitable for model training with more features.

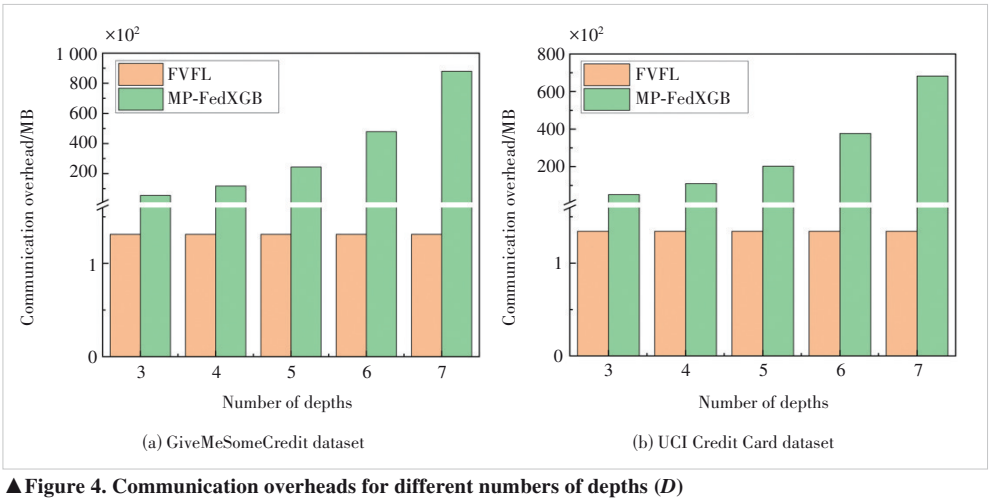
2) Varying *T*. Fig. 3 shows the trends of communication costs of FVFL and MP-FedXGB on the two data sets as *T* increases. Although the communication overheads of FVFL and MP-FedXGB increase with the increase of *T*, the communication cost of FVFL is always smaller than that of MP-FedXGB. For example, when MP-FedXGB has three trees, its traffic on the two datasets is 41.9 times and 37.9 times that of FVFL, respectively. This is because MP-FedXGB requires multiple rounds of iterations using parameter shares among the four participants and the coordinator to build a tree model, while FVFL only needs to iterate among two participants. The results show that FVFL is more suitable for multiple decision tree models.

3) Varying *D*. Fig. 4 shows the trends of communication costs of FVFL and MP-FedXGB on the two data sets as *D* increases. It can be seen that the communication overhead of FVFL will not change as *D* increases, but the communication cost of MP-FedXGB will increase rapidly as *D* changes. Regardless of the value of *D*, the traffic of MP-FedXGB is significantly higher than that of FVFL. As *D* increases, the communication advantages of FVFL will become more prominent. The communication cost of FVFL will not change with the increase in *D*, because *D* is not involved in secret sharing among the three passive parties. *D* is only involved in model training between the active party and any passive party, but *D* does not affect the interaction of gradient information after homomorphic encryption.

4) Varying *I*. Fig. 5 shows the trends of communication

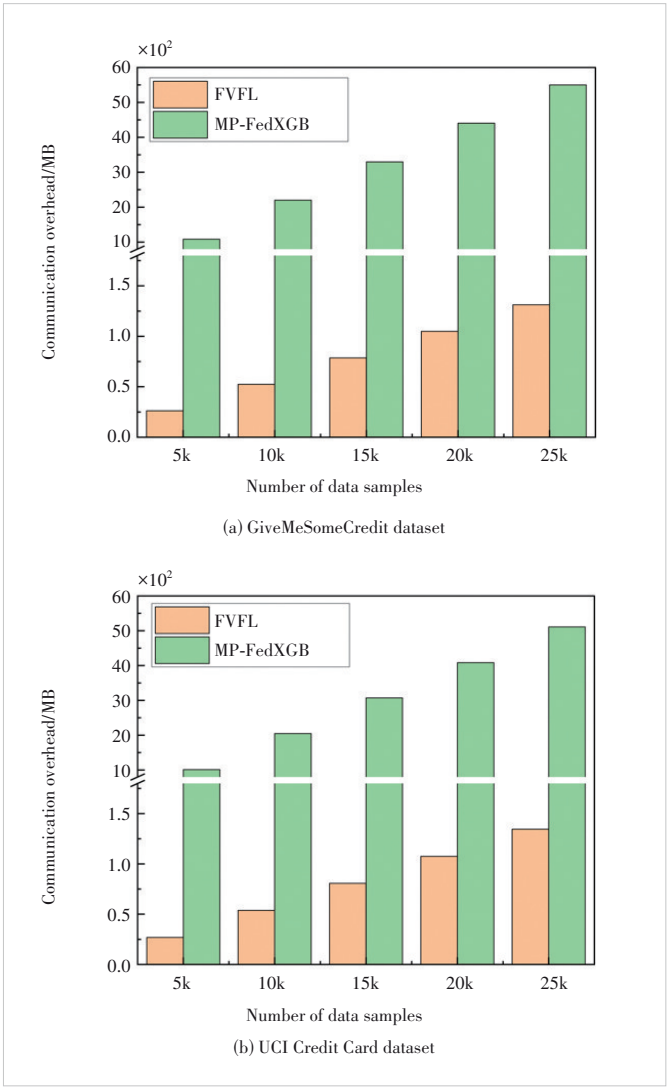


▲Figure 3. Communication overheads for different numbers of trees (*T*)



▲Figure 4. Communication overheads for different numbers of depths (*D*)

costs of FVFL and MP-FedXGB on the two data sets as I increases. Although the communication overheads of FVFL and MP-FedXGB increase with increasing I on both datasets, it is obvious that the communication overhead of FVFL is lower and increases slowly. This is because, to build the tree model, MP-FedXGB has to complete multiple rounds of iterations among the four parties under the coordination of a third party. The results show FVFL is well suited for training on large datasets.



▲ **Figure 5. Communication overheads for different numbers of data samples (I)**

6.3 Model Performance Comparisons

We compare the performance of FVFL and MP-FedXGB in training SecureBoost models on the GiveMeSomeCredit and UCI Credit Card datasets, where the number of decision tree T and depth D increases from 3 to 5, respectively. The comparison is carried out mainly from three aspects: accuracy (ACC), F1 score, and area under curve (AUC), and the results are shown in Table 3. It can be seen that under the two variables T and D, the performance of FVFL is almost better than that of MP-FedXGB in various indicators. This can be attributed to the scheme design of FVFL, which uses the features of the four parties directly for model training by the active party and any of the three passive parties. However, the participants in MP-FedXGB have to reshape the model splitting criterion based on the secret share of each party's private data.

7 Conclusions

This paper proposes an FVFL privacy protection scheme that supports the federated training of four parties. FVFL introduces a layered framework with three passive parties as one layer and active parties as another layer. Furthermore, a verifiable RSS algorithm is designed so that three passive parties can achieve the private summation of feature intersection sets. Moreover, the algorithm ensures that when a passive party exits the secret reconstruction stage, the remaining two parties can still restore the sum of the three-party features. The active party cooperates with any of the three passive parties to achieve four-party VFL training. This ensures a low communication overhead and high reliability of the FVFL. Theoretical analysis and extended experiments have verified the security and effectiveness of our FVFL.

Acknowledgment

We acknowledge Prof. YU Hongfang, XIONG Xiankui, Dr.

▼**Table 3. Model performance comparison under different parameters**

		GiveMeSomeCredit						UCI Credit Card					
		FVFL			MP-FedXGB			FVFL			MP-FedXGB		
		ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC
T	3	**0.933 3**	**0.245 7**	**0.827 0**	0.930 6	0.213 1	0.719 4	**0.823 2**	**0.466 7**	0.752 5	0.822 5	0.458 8	**0.768 6**
	4	**0.933 9**	**0.244 2**	**0.829 5**	0.932 6	0.235 8	0.746 5	**0.824 2**	0.447 2	0.760 9	0.823 7	**0.475 2**	**0.770 1**
	5	**0.934 2**	0.247 4	**0.829 9**	0.933 4	**0.292 9**	0.749 7	**0.825 8**	0.462 0	0.763 4	0.824 9	**0.473 8**	**0.773 5**
D	3	**0.933 3**	**0.245 7**	**0.827 0**	0.930 6	0.213 1	0.719 4	**0.823 2**	**0.466 7**	0.752 5	0.822 5	0.458 8	**0.768 6**
	4	**0.934 7**	**0.280 5**	**0.828 6**	0.931 8	0.201 4	0.737 9	**0.824 4**	**0.451 8**	0.762 8	0.823 9	0.446 1	**0.769 0**
	5	**0.935 4**	**0.278 2**	**0.847 9**	0.932 2	0.264 6	0.738 3	**0.824 8**	0.453 4	0.764 7	0.824 3	**0.454 6**	**0.770 7**

ACC: accuracy AUC: area under curve

LI Zonghang, and JI Kailai for their invaluable guidance and efforts in the research presented in this paper.

References

[1] KAIROUZ P, MCMAHAN H B, AVENT B, et al. Advances and open problems in federated learning [J]. Foundations and trends® in machine learning, 2021, 14(1 – 2): 1 – 210. DOI: 10.1561/2200000083
[2] YANG Q, LIU Y, CHEN T J, et al. Federated machine learning: concept and applications [J]. ACM transactions on intelligent systems and technology (TIST), 2019, 10(2): 1 – 19. DOI: 10.1145/3298981
[3] CHOUDHURY A, PATRA A. Secure multi-party computation against passive adversaries [M]. Cham: Springer International Publishing, 2022. DOI: 10.1007/978-3-031-12164-7
[4] OU W, ZENG J H, GUO Z J, et al. A homomorphic-encryption-based vertical federated learning scheme for rick management [J]. Computer science and information systems, 2020, 17(3): 819 – 834. DOI: 10.2298/csis190923022o
[5] WANG C, LIANG J, HUANG M, et al. Hybrid differentially private federated learning on vertically partitioned data [J]. 2020. DOI: 10.48550/arXiv.2009.02763
[6] HE D J, DU R M, ZHU S S, et al. Secure logistic regression for vertical federated learning [J]. IEEE Internet computing, 2022, 26(2): 61 – 68. DOI: 10.1109/MIC.2021.3138853
[7] NI X, XU X L, LYU L J, et al. A vertical federated learning framework for graph convolutional network [EB/OL]. (2021-06-22)[2023-06-13]. https://arxiv.org/abs/2106.11593
[8] YANG S W, REN B, ZHOU X H, et al. Parallel distributed logistic regression for vertical federated learning without third-party coordinator [EB/OL]. (2019-11-22)[2023-06-13]. https://arxiv.org/abs/1911.09824v1
[9] LI Q B, WU Z M, CAI Y Z, et al. FedTree: a federated learning system for trees [C]//Proc. Machine Learning and Systems 5. MLSys, 2023
[10] XIE L C, LIU J Q, LU S T, et al. An efficient learning framework for federated XGBoost using secret sharing and distributed optimization [J]. ACM transactions on intelligent systems and technology, 2022, 13(5): 1 – 28. DOI: 10.1145/3523061
[11] XU R H, BARACALDO N, ZHOU Y, et al. FedV: privacy-preserving federated learning over vertically partitioned data [C]//Proc. 14th ACM Workshop on Artificial Intelligence and Security. ACM, 2021. DOI: 10.1145/3474369.3486872
[12] YANG K, FAN T, CHEN T, et al. A quasi-newton method based vertical federated learning framework for logistic regression [EB/OL]. (2019-12-01)[2023-07-11]. https://arxiv.org/abs/1912.00513
[13] FANG W J, ZHAO D R, TAN J, et al. Large-scale secure XGB for vertical federated learning [C]//Proc. 30th ACM International Conference on Information & Knowledge Management. ACM, 2021. DOI: 10.1145/3459637.3482361
[14] CHEN W J, MA G Q, FAN T, et al. Secureboost+: a high performance gradient boosting tree framework for large scale vertical federated learning[EB/OL]. (2021-10-21) [2023-07-12]. https://arxiv. org/abs/2110.10927v2
[15] SHI H R, XU Y H, JIANG Y L, et al. Efficient asynchronous multi-participant vertical federated learning [J]. IEEE transactions on big data, 2024, 10(6): 940 – 952. DOI: 10.1109/TBDATA.2022.3201729
[16] WU Y C, CAI S F, XIAO X K, et al. Privacy-preserving vertical federated learning for tree-based models [J]. Proceedings of the VLDB endowment, 2020, 13(12): 2090 – 2103. DOI: 10.14778/3407790.34078112020.
[17] HUANG Y M, FENG X Y, WANG W W, et al. EFMVFL: an efficient and flexible multi-party vertical federated learning without a third party [EB/OL]. (2022-01-17)[2023-07-15]. https://arxiv.org/abs/2201.06244
[18] ABSPOEL M, DALSKOV A, ESCUDERO D, et al. An efficient passive-to-active compiler for honest-majority MPC over rings [C]//International Conference on Applied Cryptography and Network Security. ACNS, 2021: 122 – 152. DOI: /10.1007/978-3-030-78375-4_6

Biographies

FAN Mochan is a PhD candidate at University of Electronic Science and Technology of China (UESTC). She received her BS degree from Suzhou University of Science and Technology, China and MS degree from Jiangxi University of Science and Technology, China. Her research interests include network security, blockchain, and federated learning.

ZHANG Zhipeng (2474297092@qq.com) is pursuing a master's degree at University of Electronic Science and Technology of China (UESTC). He received his bachelor's degree from UESTC. His research interests include network security and distributed machine learning.

LI Difei is pursuing a master's degree in communication and information system at University of Electronic Science and Technology of China. His research focuses on machine learning.

ZHANG Qiming is a senior system architect at ZTE Corporation. He received his bachelor's degree from Zhejiang University, China in 1992. His research interests include MEC and heterogeneous computing.

YAO Haidong is a senior system architect at ZTE Corporation. He is engaged in the research and design of deep learning, large model network architecture, and compilation conversion technology.