

Accelerating Vertical Federated Learning

Dongqi Cai, Tao Fan, *Member, IEEE*, Yan Kang, Lixin Fan, *Member, IEEE*, Mengwei Xu, *Member, IEEE*, Shanguang Wang, *Senior Member, IEEE*, and Qiang Yang, *Fellow, IEEE*

Abstract—Privacy, security and data governance constraints rule out a brute force process in the integration of cross-silo data, which inherits the development of the Internet of Things. Federated learning is proposed to ensure that all parties can collaboratively complete the training task while the data is not out of the local. Vertical federated learning is a specialization of federated learning for distributed features. To preserve privacy, homomorphic encryption is applied to enable encrypted operations without decryption. Nevertheless, together with a robust security guarantee, homomorphic encryption brings extra communication and computation overhead. In this paper, we analyze the current bottlenecks of vertical federated learning under homomorphic encryption comprehensively and numerically. We propose a straggler-resilient and computation-efficient accelerating system that reduces the communication overhead in heterogeneous scenarios by 65.26% at most and reduces the computation overhead caused by homomorphic encryption by 40.66% at most. Our system can improve the robustness and efficiency of the current vertical federated learning framework without loss of security.

Index Terms—Cross-Silo, Secure Cooperation, Homomorphic Encryption, Distributed Features.



1 INTRODUCTION

DATA fuels the growing popularity of the Internet of Things. With the ubiquitous deployment of dedicated and multi-purpose sensors, collecting real-time information about our environment is becoming more convenient. However, due to the specific limitations of dedicated sensors, information collected by different industries tends to have divergent feature dimensions. This divergence makes cross-silo training and inferring challenging. The traditional method is to establish a centralized data center led by relevant authorities. Although this can break the data isolation, it is not feasible today. Because of the tremendous value of this real-time information, such convenient access to monitor surroundings makes people concerned about the leakage and abuse of their private and sensitive data. Apart from the individual privacy concern, the implementation of user privacy laws such as GDPR [28] has set a strict limit on the usage of the collected data. Therefore, cross-silo data sharing becomes challenging, which hinders the development of the Internet of Things.

To solve this dilemma, federated learning (FL) [16] was proposed by Google in 2016. FL ensures that all parties can collaboratively complete the training task while the data is not out of the local, which is an ideal framework to alleviate privacy concerns. However, most FL research focuses on the sample partitioned scenario [32], [37], where all cooperators share the same feature dimension. The feature partitioned federated learning is rarely explored in literature, which is an equally important issue in real industrial scenarios, such as recommender system [13], credit evaluation [21], etc.

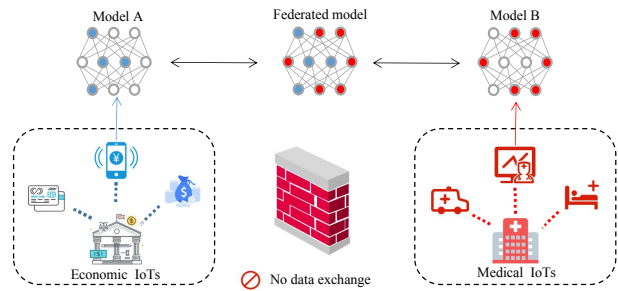


Fig. 1: Vertical federated learning. A specialization of federated learning for secure cross-silo cooperation learning.

Vertical federated learning (VFL) is a specialization of federated learning for distributed features [10]. As shown in Fig. 1, economic and medical institutions have divergent feature dimension. They build a federated model together without exchanging raw data. To further preserve privacy, federated model is in safekeeping by a third-party authority, which prohibits each institution from getting model structure of others. Unlike conventional FL (or noted horizontal FL) for distributed samples, which has been researched a lot due to its continuous line with traditional distributed learning, VFL is under-studied for the lack of datasets and benchmarks. Apart from the lack of related infrastructure, communication redundancy in VFL is considerably larger than in conventional FL. Because not only gradients are transmitted during cooperation, intermediate results such as labels and loss are needed to be exchanged due to distributed features [32], [37].

Homomorphic encryption (HE) [39], a common privacy-preserving technology, is applied to provide secure protection of exchanged information. Apart from HE, differential privacy [4], quantization [29], compression [30], and coded computing [25] can be applied to preserve the privacy as well. However, they are vulnerable to aggressive Bayesian

L. Fan and M. Xu are the corresponding authors. Email: lixin-fan@webank.com, mwx@bupt.edu.cn.

D. Cai, M. Xu and S. Wang are with State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China. L. Fan, T. Fan, Y. Kang are with Department of Artificial Intelligence, Webank, Shenzhen, China. Q. Yang is with both Webank and Hong Kong University of Science and Technology

restoration attacks due to their randomization privacy protection [5], [31], [36], [42]. Considering that cooperators in VFL training need to have fruitful data features for effective collaborative training, they tend to be competitive enterprises in certain fields, such as industry, finance, medicine, etc. Such a potential adversarial relationship resorts to the most strict privacy preservation for these precious data. HE ensures security by allowing encrypted operations without decryption and thus providing cryptographic privacy protection. Consequently, HE is studied as our focus of privacy preservation.

Nonetheless, in addition to the robust security guarantee, HE leads to two orders of magnitude size inflation of the original data after encryption, which burdens the communication and incurs massive delays. For example, a few-minute local training task will take dozens of minutes or even hours to complete in VFL under homomorphic encryption. This extra delay, together with the common heterogeneity [22] existing in cooperation learning, damages the runtime performance of VFL. To make matters worse, unlike conventional FL, parties in vertical federated learning can not compute gradients locally due to the lack of labels. To compute the gradient, they need to exchange the encrypted intermediate results. Computational costs under homomorphic encryption are much higher than that under plain text. Huge computation overhead further slows down vertical federated learning under homomorphic encryption.

In this paper, we analyze the main bottlenecks of current vertical federated learning numerically and implement a practical and efficient accelerating system under FATE¹. As for communication overhead, we use a straggler-resilient scheme based on backup worker [6] to alleviate the overhead caused by heterogeneity. For computation overhead, we compress the input matrix based on the principal component analysis [1] and reduce the computational redundancy caused by homomorphic encryption. Our system reduces the communication overhead in heterogeneous scenarios range 13.99%-65.26% with different network situations and backup worker settings, and reduces the computation overhead caused by HE range 21.95%-40.66% with different degrees of compression. Our system can significantly improve the efficiency of the current vertical federated learning without loss of security. In summary, we have made the following contributions:

- We carry out the first VFL system measurement on industrial federated learning framework FATE and demonstrate its main performance bottlenecks.
- We propose an accelerating system designed specifically for tackling VFL performance bottlenecks without loss of security.
- We demonstrate the effectiveness of our system through extensive experiments. Our system makes VFL more practical for real word settings.

The structure of this paper is as follows. In Section 2, we analyze the inspiration of this paper from the perspective of

the advantages and bottlenecks of vertical federated learning under homomorphic encryption. Then we propose our system to overcome the existing bottlenecks and accelerate vertical federated learning in Section 3. The experimental results showing the efficiency and robustness of our system are shown in Section 4. Related work is introduced in Section 5. Finally, we conclude our paper and put forward several directions for future work in Section 6.

2 BACKGROUND AND MOTIVATION

This part introduces the vertical federated learning framework adopted by FATE [20] and its efficiency in privacy-preservation. Then we dive into the bottlenecks of the existing framework and raise our motivation.

2.1 Vertical Federated Learning Architecture

Different from conventional FL, each cooperator has a different feature dimension in vertical federated learning. Apart from distributed features, in order to preserve privacy, labels of cooperators will not be shared with others. In other words, only the cooperator who seeks collaboration owns the labels. We note that the cooperator seeking collaboration and owing labels as Guest. And those who attend the collaboration are called Hosts. The lack of labels and features directly leads to the fact that cooperators can not compute loss locally and thus fail to compute gradients. So conventional federated learning procedure is not feasible in vertical federated learning.

Hardy et al. propose a secure cooperative learning framework [10] for vertical data using additional homomorphic encryption. The framework is as precise as a naive non-private solution that brings all data in one place and scales to problems with millions of entities with hundreds of features. It consists of two parts. The first part is encrypted entity alignment. Considering that data samples of different partners are not the same, the system utilizes many encryption allocation schemes to locate these common data samples in multiple parties. In this process, the data information that does not coincide with each other will not be disclosed. The second part is encrypted model training, which is the focus of our work. After encrypted entity alignment, it can be assumed that the data samples of all parties are consistent, but the feature spaces of all parties do not coincide. As shown in Fig. 2, before the training, the authoritative arbiter randomly generates the private key and public key. Then the arbiter distributes the public key to each cooperator for encryption. Note that cooperators have no right or opportunity to obtain the private key. That is, cooperators have only the right to encrypt, but not the right to decrypt, which ensures that the interactive data is presented in ciphertext. During one iteration, each cooperator first completes the forward propagation and obtains the intermediate results in plain mode. After encrypting the intermediate result with their own public key, cooperators exchange the encrypted intermediate results. Then they can gather a complete encrypted loss, containing the data feature information of other parties and label information. They use the encrypted loss to compute encrypted gradients and send encrypted gradients to arbiter for decryption. After

1. Federated AI Technology Enabler (FATE) is an open-source industrial federated learning framework, which has been widely applied in the financial and medical fields. Git repository: <https://github.com/FederatedAI/FATE>.

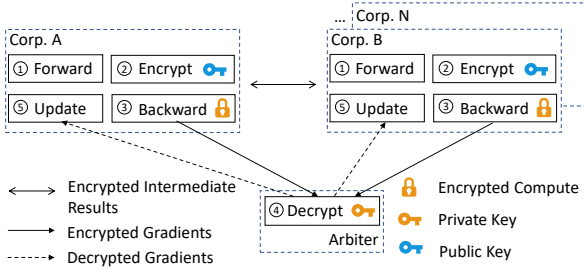


Fig. 2: The framework of vertical federated learning, where homomorphic encryption is applied to preserve privacy.

decrypting their respective gradients, the arbiter will return the separated gradient parameters to each cooperator.

2.2 Encryption mechanism

Homomorphic encryption allows direct operations on the ciphertext without decryption, which has attracted a large number of researchers to explore and study in recent years [2], [7], [21], [39]. The most commonly used homomorphic encryption algorithm is Paillier [27]. In paillier, additive homomorphic is satisfied, which means that plaintext addition is realized through the overloaded operator of ciphertext. More precisely, the result of ciphertext multiplication after decryption is equivalent to plaintext addition:

$$D([x] \bullet [y]) = x + y, \quad (1)$$

where $[x]$ is the ciphertext of x , function D represents decryption. Similarly, the result of ciphertext exponentiation after decryption is equivalent to plaintext multiplication. Note that the size of ciphertext tends to be excessively bigger than the size of plaintext. For example, the commonly used plaintext is 8-bit or 16-bit, while the shortest ciphertext is 2048-bit till 2019 [39]. Homomorphic encryption will greatly increase the complexity of the operation, especially for multiplication.

2.3 Performance Bottleneck

We carry out some preliminary experiments² to show the performance bottleneck intuitively. As shown in Fig. 3, vertical federated learning mainly consists of four parts, namely *Computation*, *Encryption*, *Communication* and *Other* operation. *Other* operation is the time of IO operations and task scheduling. Since encryption has a negligible impact on IO operations, we can reasonably assume that time consumption of *Other* is static (100s) in all three modes, and use it as a comparing reference. Plain mode stands for vertical federated learning without homomorphic encryption and heterogeneity, where *Other* covers the most running time. Encrypt mode stands for vertical federated learning under homomorphic encryption without network heterogeneity, which means that no straggler will appear. In encrypt mode, *Computation* consumes most of time, *Encryption* consumes second. Hetero-encrypt mode is vertical

² Experiments are on *Epsilon* [9] dataset with 5000 samples and 2000 features separated for four cooperators. Computing capacity remains constant during training. Training performance of three modes is identical after 50 iterations.

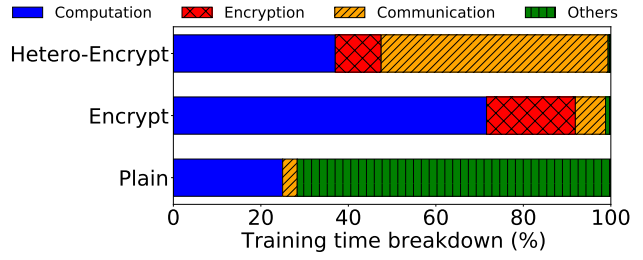


Fig. 3: Runtime breakdown under different VFL settings.

federated learning in practical deployment, where both homomorphic encryption and heterogeneity are involved. In hetero-encrypt mode, *Communication* surpasses in reverse. Among those three bottlenecks, the increase in encryption time is common, which has been widely studied [2], [39]. So we dive into those two specific bottlenecks: computation overhead and communication overhead.

2.3.1 Computation Overhead

Different from the horizontal federated learning, the parameters transmitted in the vertical federated learning are not only gradients, but also intermediate results. Moreover, due to the potential competitive relationship, all parties do not have the authority to decrypt the parameters to preserve privacy. This means that homomorphic encryption not only brings the redundancy contained in the encryption and decryption computation itself, but also forces all parties to do computation on ciphertext instead of plaintext. And the ciphertext computation of homomorphic encryption is very time-consuming, especially for multiplication.

2.3.2 Communication overhead

Heterogeneous networks and huge geographical spans are the main reasons for communication overhead. More specifically, multiple institutions with different features may be distributed in all corners of the earth, and their geographical positions are generally considerably different. Moreover, the network condition of a single organization is difficult to ensure stability due to the network heterogeneity [38]. Because each institution itself has limited communication resources [23], such as carrier, bandwidth, e.t.c. These resources may be allocated to other jobs, as cooperators do not exist exclusively for federated learning. These services will bring huge network traffic during the peak hours, and even lead to network congestion. What is worse, homomorphic encryption will expand the data size, which amplifies the impact of network fluctuation.

The above two bottlenecks cause much idle time in vertical federated learning. That is, one party may spend more time waiting for the calculation results than other parties. Such random and unbalanced idle time brings difficulty to the application and promotion of vertical federated learning.

3 PROPOSED SYSTEM

In order to reduce the communication and computation overhead in vertical federated learning caused by homomorphic encryption and heterogeneity, we propose a prac-

tical and effective optimization system under the industrial federated learning framework FATE [20].

3.1 Problem Definition

We suppose one cooperator seeks help for K cross-industry cooperators. For simplicity, we call cross-industry cooperators without labels as Hosts, denoted as A_k , for $k = 1, 2, \dots, K$. Similarly, we call the cooperator seeking help and owing labels as Guest, denoted as B . Assuming learning rate μ , regularization parameter λ , dataset set $\{x_i^{A_k}\}_{i \in \mathcal{D}_{A_k}}, \{x_i^B, y_i\}_{i \in \mathcal{D}_B}$, and model parameters Θ_{A_k}, Θ_B corresponding to the feature space of $x_i^{A_k}, x_i^B$ respectively, the training objective is:

$$\min_{\Theta_{A_k}, \Theta_B} \sum_i \left\| \sum_{k=1}^K \Theta_{A_k} x_i^{A_k} + \Theta_B x_i^B - y_i \right\|^2 + \frac{\lambda}{2} \left(\sum_{k=1}^K \|\Theta_{A_k}\|^2 + \|\Theta_B\|^2 \right). \quad (2)$$

Let $u_i^{A_k} = \Theta_{A_k} x_i^{A_k}, u_i^B = \Theta_B x_i^B$, the encrypted loss is:

$$[[\mathcal{L}]] = \left[\left[\sum_i \left\| \sum_{k=1}^K u_i^{A_k} + u_i^B - y_i \right\|^2 + \frac{\lambda}{2} \left(\sum_{k=1}^K \|\Theta_{A_k}\|^2 + \|\Theta_B\|^2 \right) \right] \right], \quad (3)$$

where additive homomorphic encryption is denoted as $[[\cdot]]$. Let $[[\mathcal{L}_A]] = \left[\left[\sum_i \left(\sum_{k=1}^K u_i^{A_k} \right)^2 + \frac{\lambda}{2} \sum_{k=1}^K \Theta_{A_k}^2 \right] \right]$, $[[\mathcal{L}_B]] = \left[\left[\sum_i \left((u_i^B - y_i)^2 \right) + \frac{\lambda}{2} \Theta_B^2 \right] \right]$, and $[[\mathcal{L}_{AB}]] = 2 \sum_i \sum_{k=1}^K \left([[u_i^{A_k}]] (u_i^B - y_i) \right)$, then

$$[[\mathcal{L}]] = [[\mathcal{L}_A]] + [[\mathcal{L}_B]] + [[\mathcal{L}_{AB}]]. \quad (4)$$

Note that an additively homomorphic encryption scheme provides an operation that produces the sum of two numbers under encryption. For simplicity, we overload the notation and we denote the operator with "+" as well. Similarly, let $[[d_i]] = \sum_{k=1}^K [[u_i^{A_k}]] + [[u_i^B - y_i]]$, then gradients are:

$$\left[\left[\frac{\partial \mathcal{L}}{\partial \Theta_{A_k}} \right] \right] = \sum_i [[d_i]] x_i^{A_k} + [[\lambda \Theta_{A_k}]], \quad (5)$$

$$\left[\left[\frac{\partial \mathcal{L}}{\partial \Theta_B} \right] \right] = \sum_i [[d_i]] x_i^B + [[\lambda \Theta_B]].$$

So, for iteration $j = 1, 2, \dots$, updated model parameters are:

$$\Theta_{A_k}^{j+1} = \Theta_{A_k}^j - \mu \frac{\partial \mathcal{L}}{\partial \Theta_{A_k}}, \quad (6)$$

$$\Theta_B^{j+1} = \Theta_B^j - \mu \frac{\partial \mathcal{L}}{\partial \Theta_B}.$$

3.2 Backup Worker

Cooperators in vertical federated learning need to exchange encrypted intermediate results. As can be seen from the formula (5), each cooperator can calculate the gradients with local data after getting the global d . The critical point of vertical federated learning is to calculate the global d cooperatively. As mentioned in Section 2, the network environment is likely to be heterogeneous and volatile. The existing synchronization framework will lead to much idle time, which will damage the efficiency and robustness of the training process.

The simple idea is to discard the unreachable parameters, which is effective in distributed learning because only gradients are transmitted from partial workers [6]. But considering the difference between VFL and conventional distributed learning, such an arbitrary drop scheme may not perform well in VFL. As shown in Fig. 2, cooperators in VFL exchange gradients with arbiter to decrypt gradient instead of aggregating gradients. This means that dropping gradients can block the learning progress or degrade the clients to training locally without collaboration. Variables transmitted between cooperators are encrypted intermediate results, which will be used for gradient computation. So arbitrary discard will inevitably damage training performance. The disadvantage of this method is magnified in the longitudinal federated learning, because some clients will accumulate such damages and turn into performance stragglers, which means its misleading drags the global performance.

Therefore, in order to enhance the stability of communication without excessive loss of accuracy, we combine the Stale Synchronous Parallel Parameter Server (SSP) [12] scheme with the original backup scheme [6], that is, using the old backup data to fill in the parameters corresponding to the missing positions in this round. The detailed algorithm is shown in Algorithm 1. Assuming K hosts with dataset $x_i^{A_k}$ and one guest with dataset x_i^B attend the cooperation, we set the number of backup workers as β . Hosts compute $u_i^{A_k}$ with local dataset, and encrypt it with homomorphic encryption for secure exchange. Guest waits for $u_i^{A_k}$ from A_k . After receiving the first $K - \beta$ parameters in each iteration, the guest stops waiting for data from other parties, and then uses the stale backup of the unreceived party to compute $[[\mathcal{L}]]$ and $[[d_i]]$.

$$[[\mathcal{L}]] = \left[\left[\sum_i \left\| \sum_{s=1}^t u_i^{A_{j_s}} + \sum_{s=t+1}^K u_i^{\hat{A}_{j_s}} + u_i^B - y_i \right\|^2 + \frac{\lambda}{2} \left(\sum_{s=1}^t \|\Theta_{A_{j_s}}\|^2 + \sum_{s=t+1}^K \|\Theta_{\hat{A}_{j_s}}\|^2 + \|\Theta_B\|^2 \right) \right] \right], \quad (7)$$

$$[[d_i]] = \sum_{s=1}^t [[u_i^{A_{j_s}}]] + \sum_{s=t}^K [[u_i^{\hat{A}_{j_s}}]] + [[u_i^B - y_i]], \quad (8)$$

where $u_i^{\hat{A}_{j_s}}, \Theta_{\hat{A}_{j_s}}$ is the j_{th} iteration's backup variables, $t = K - \beta$, j_s is the receive sequence of j_{th} iteration. The backup scheme can be various, such as average, local prediction and so on. For the convenience of discussion, the simplest method is chosen, that is, to fill the unreceived

parameters with the results in the previous iteration. The experimental part below shows that even this simple scheme has better effects than the arbitrary discard scheme.

Algorithm 1: Backup for VFL

Input: dataset $x_i^{A_k}$, dataset x_i^B , backups β
Output: Model parameters $\Theta_{A_k}, \Theta_B (k = 1, 2..K)$
 Party A_k, B initialize Θ_{A_k}, Θ_B ;
for each iteration $j = 1, 2 \dots \mathbf{do}$
 | Parallel: Host(j), Guest(j)
end for
 Host(j):
for $k=1,2..K$ **in parallel do**
 | A_k computes $u_i^{A_k}$, and send $[[u_i^{A_k}]], [[\Theta_{A_k}^2]]$ to B
 | Get $[[d_i]]$ from B
 | Update Θ_{A_k} with (5) (6)
end for
 Guest(j):
if $[[[u_i^{A_k}]]] < K - \beta$ **then**
 | Waiting for $u_i^{A_k}$ from A_k
else
 | B computes d_i with (8) and sends $[[d_i]]$ to A_k
end if
 Update Θ_B with (5) (6)

Remark. Backup worker allows distributed workers to use older, stale versions of model’s values (gradients and logits). This significantly helps workers to spend more time on computing instead of idle waiting. Furthermore, the SSP model [12] can ensure the correctness of ML algorithm by limiting the maximum age of the stale values. Considering a nearly half proportion of encrypted computation in VFL training (shown in Fig. 3), common network fluctuation (the best network bandwidth is within one magnitude of the worst) will not incur the age of stale value ≥ 2 . In this circumstance, the SSP model can perform on par with vanilla synchronous training. Note that our system does not use any intelligence client selection mechanism [26], [33], [34], but is compatible with above techniques. These techniques can be injected to further ensure the upper bound of the stale age.

3.3 Dynamic Feature Selection

Multiplication of homomorphic encryption consumes a huge computation cost. We assume that the host data set $x_i^{A_k}$ has m data samples and n features, noted as $X_{m \times n}$. So a gradient calculation $[[d_i]]x_i^{A_k}$ in (5) requires $m \times n$ times of encrypted multiplication. This leads to the huge computation overhead in vertical federated learning, the experimental results have been shown in Fig. 3.

We compress the feature quantity by the dimension reduction method called Principal Component Analysis (PCA) [1] to reduce the frequency of multiplication, shown in (9).

$$Z_{m \times k} = f(X_{m \times n}), k < n, \quad (9)$$

where $Z_{m \times k}$ is the compressed input matrix, f is a mapping matrix to maximize the variance of data under the specified dimension K , note f as $W_{k \times n}^T$.

The traditional principal component analysis is only a one-time dimensionality reduction, which can not be organically combined with current vertical federated learning. Therefore, we design a scheme shown in Algorithm 2 so that principal component analysis can be effectively integrated into vertical federated learning. At the beginning of each iteration, cooperators generate a compression matrix $W_{k \times n}^T$, the target dimension k can be set flexibly to find a balance in efficiency and accuracy. First, cooperators compress the input matrix $\Theta_{1 \times n}, X_{n \times m}$ into $\theta_{1 \times k}, Z_{k \times m}$.

$$\begin{aligned} \theta_{1 \times k} &= \Theta_{1 \times n} W_{k \times n}^T, \\ Z_{k \times m} &= W_{k \times n} X_{n \times m}. \end{aligned} \quad (10)$$

Then they can multiply the compressed input matrix with encrypt $[[d_i]]$ using (4) (5) to get gradients $[[\frac{\partial \mathcal{L}}{\partial \Theta_{A_k}}]]$ with k gradients, noted as $g_{1 \times k}$. The compression mechanism turns the encrypted matrix from $X_{n \times m}$ to $Z_{k \times m}$ and thus reduces $m \times n$ times of homomorphic encryption multiplication to $m \times k$ times.

At the end of each iteration, cooperators will decompress the decrypted gradient.

$$g_{1 \times n} = g_{1 \times k} W_{k \times n}^{-1}. \quad (11)$$

Remark Compression and decompression are both performed on local datasets in plaintext. The computation overhead of plaintext is nearly negligible compared to that of ciphertext multiplication. And our compression behavior is recoverable by storing the PCA matrix locally. It avoids the inexplicability caused by compressing features on the air, and ensures that the algorithm can perfectly fit the existing VFL framework without enormous modification.

Algorithm 2: PCA for VFL

Input: dataset $X_{m \times n}$
Output: Model parameters $\Theta_{1 \times n}$
 Initialize $\Theta_{1 \times n}$;
for each iteration $j = 1, 2 \dots \mathbf{do}$
 | Generate the PCA matrix $W_{k \times n}^j$
 | Compress the input matrix with (10)
 | VFL Train with (4) (5)
 | Unzip gradient $g_{1 \times k}$ with (11)
 | Update $\Theta_{1 \times n}^j$ with (6)
end for

4 EXPERIMENTS

We carry out experiments on the industrial federated learning framework FATE [20]. Our accelerating system dramatically improves the efficiency and robustness of the current vertical federated learning without loss of security.

4.1 Setting

Federated Learning Platform: We build our system on FATE [20]. FATE is the most popular federated learning framework in Linux and has built cooperation with many enterprises in many fields, such as industry, finance, medical, etc.

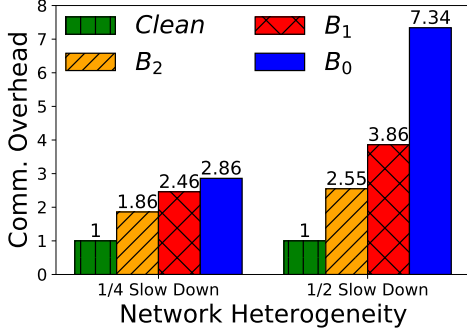


Fig. 4: Effect of backup workers: communication speedup. *Clean* means no heterogeneity, B_i means the number of backup workers is i in various heterogeneity (different possibilities of slow down). Dataset: MIMIC-III.

Datasets: We use three datasets MIMIC-III [14], Epsilon [9] and NUS-WIDE [8]. MIMIC-III is a real-time medical database comprising information about patients admitted to critical care units at a large tertiary care hospital, including vital signs, medications, survival data and more. Following the data processing procedures of [11], we get the final training and testing sets of 14681 data samples and 714 features. Then we shuffle it into 4 datasets with 114, 200, 200, 200 features. Datasets are dispatched to 4 cooperators: Guest, Host₁, Host₂ and Host₃ respectively. Epsilon dataset is instance-wisely scaled to unit length and feature-wisely normalized to mean zero and variance one. The raw dimension of Epsilon is 5000 data samples and 2000 data features. We shuffle it into 4 datasets with 200, 600, 600, 600 features and dispatch the shuffled datasets to guest and hosts respectively. The NUS-WIDE dataset consists of 634 low-level images features extracted from Flickr images [24] as well as their associated tags and ground truth labels. We put 34 low-level image features on party Guest and 200 features on three Host parties respectively³.

Models: We use Logist Regression, a common specialization of Linear Regression, to verify the validity of our system. We set the best hyper-parameter at our best considering the efficiency and training performance. Default hyper-parameters are as follows: optimizer is rmsprop; learning rate is 0.05; batchsize for MIMIC-III and NUS-WIDE is 1024, and 5000 for Epsilon.

4.2 Backup Worker

Our system is robust under various network environments and can reduce considerable communication overhead. We evaluate our system performance under two different heterogeneous network environments with a baseline bandwidth of 10Mb. The degree of heterogeneity is quantified by the possibility of slow down, i.e., reaching the bottom bandwidth (1/10 of baseline bandwidth) during network

3. Our micro experiments show that unbalanced features have the almost consistent runtime performance as balanced features. Because in the forward propagation, the computation overhead caused by redundant features is in plaintext, which has negligible impact on runtime performance. Hence, we reasonably assume that hosts' features are balanced.

fluctuation [34]. There are three Hosts and one Guest attending the cooperation training. As shown in Fig. 4, we use a stable bandwidth network as a baseline, called *Clean*. In the situation of 1/4 slow down, the bandwidth will be reduced to 1Mb in the possibility of 1/4. B_0 represents the number of backup workers is 0, which equals the vanilla vertical federated learning in a heterogeneous network. The communication overhead is 2.86 times that of *Clean* situation. We set the number of backup workers to 1, 2, it will reduce the communication overhead by 13.99% and 34.97% respectively. When the probability of network fluctuation rises to 1/2, which means that the network condition becomes more unstable, network heterogeneity will lead to a sharp increase in network communication overhead. Vertical federated learning under 1/2 slow down without optimization (B_0) will be $7.34\times$ slower than *Clean*. The effect of our system will be more obvious. When the number of backups is 1, 2 respectively, the communication overhead can be reduced by 47.41% and 65.26% respectively.

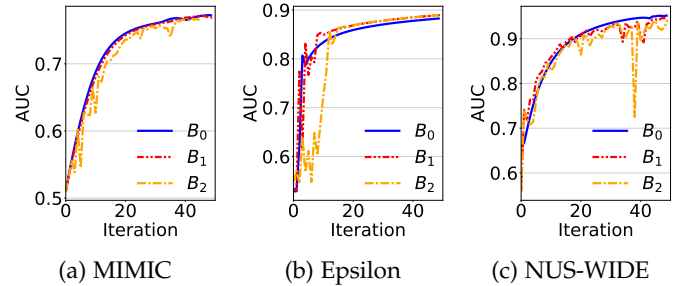


Fig. 5: Effect of backup workers: AUC vs Iteration.

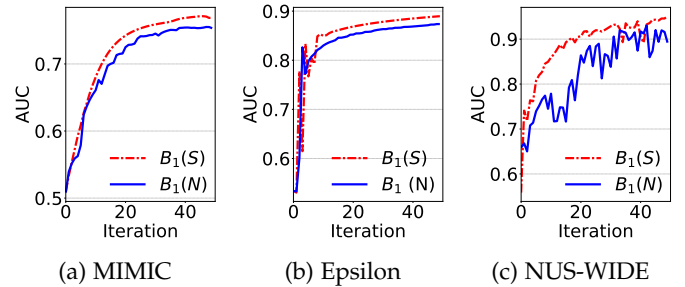


Fig. 6: Effect of different backup schemes. Stale scheme means to compensate the unreceived parameters with the stale variant, Null scheme means unreceived parameters are arbitrary dropped.

And this dramatic reduction in communication overhead does not affect the training performance. As shown in Fig. 5, when the number of backup workers is set to 1, the training process and final performance are almost the same as vanilla VFL (B_0). When the number of backup workers is set to 2, the final training performance is almost the same as B_0 , but the training process will become unstable. In Fig. 6, we compare the effect of two different backup schemes. The conventional backup scheme ignoring the unreceived parameters has a noticeable loss of convergence performance in three datasets. This is because some critical information of the global training loss is dropped, leading to a huge deviation when computing encrypted gradients. In

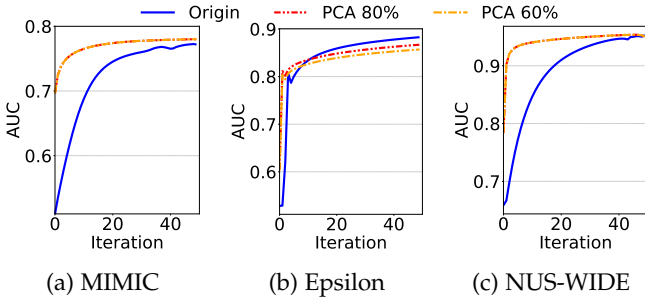


Fig. 7: Effect of PCA: AUC vs Iteration. Origin means vanilla VFL. PCA converts the input dimension to the target compressed dimension.

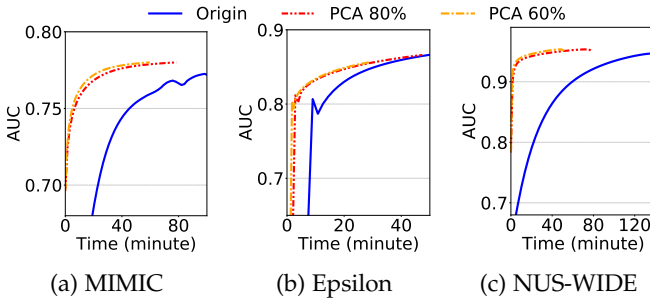


Fig. 8: Runtime performance of PCA: AUC vs Time.

contrast, compensating the unreceived parameters with the stale variants can effectively narrow the deviation when the staleness age is limited [12] and perform on par with vanilla VFL.

4.3 Principle Component Analysis

As shown in Fig. 7, the feature compression method called principal component analysis has different performance on different datasets. Origin means vanilla VFL without optimization. PCA converts the input dimension to the target compressed dimension. In Fig. 7 (a) and (c), our feature compression scheme that reduces the feature dimension to 60% does not cause a significant loss of accuracy on MIMIC-III and NUS-WIDE. It even converges faster due to the pre-analysis brought from PCA. While on the Epsilon dataset shown in Fig. 7 (b), training performance under PCA compression leads initially but fails a little when convergence. In other word, although the data after the initial dimensionality reduction using principal component analysis performs better than the original method, the final training performance still suffers a certain degree of accuracy loss. With the increase of compression degree, the loss of accuracy will become more apparent. Reducing the feature dimension to 60% on Epsilon will lose nearly 3% precision when convergence. Such a ‘dataset affinity’ may come from the limitations of the principal component analysis, which compressed the feature dimension before training. The compression degree of which should be cautiously decided according to the intrinsic correlation of datasets.

Nonetheless, the possible accuracy loss cannot conceal the enormous increase of runtime performance brought by feature compression. As shown in Fig. 8, after compressing

Methods		Origin	Backup	PCA	Ours
Time (minute)	Comp.	98.2	93.6	58.5	59.9
	Comm.	141.0	48.3	137.6	46.4
	Sum	239.2	141.9	196.1	106.3

TABLE 1: Runtime performance under different modes. Origin means vanilla VFL without any optimization. Number of backup workers is set to 2 in Backup mode. PCA mode reduces the feature dimension to 60%. Ours means the whole accelerating system composed of both Backup and PCA. All the four methods are carried out for 50 iterations on MIMIC-III.

the input dimension via PCA, runtime performance is significantly improved for all three datasets. For example, compressing the features dimension of dataset MIMIC-III to 80% and 60%, the convergence speed is increased by 21.95% and 40.66% compared with the original scheme. Analogously, training performance of Epsilon and NUS-WIDE is significantly improved as well. This performance speedup comes from the linearly reduction of encrypted computation, which accounts for around half of the VFL training time.

4.4 Mixture

The above two methods are orthogonal and can be used in combination. The global efficiency and robustness of the system can be enhanced by mutual compensation, which means when backup scheme detects stragglers, we can set a higher compression degree to speed up their computation. As shown in TABLE 1, the backup scheme reduces the communication delay by up to 65%, the PCA scheme reduces the calculation delay by up to 40.6%. Idle time caused by communication and computation overhead is reduced by 55.6% using two schemes jointly, which will significantly accelerate current vertical federated learning.

5 RELATED WORK

5.1 Related Framework

In order to get rid of the computation and communication redundancy caused by homomorphic encryption, Hu et.al propose a novel vertical federated learning framework on the assumption of shared labels [13]. This assumption may be valid within a single cilo. But when the scenario is extended to cross-cilo cooperation, many labels shall not be shared due to privacy regulation and corporate competition, limiting it to be applied in large-scale industrial scenarios. Therefore, to the best of our knowledge, VFL under homomorphic encryption is still the safest framework for cross-silo cooperating training except from multi-party secure computing [41], the efficiency of which is far from usability in practical. Some works have been proposed to make homomorphic encryption more suitable for federated learning. In the horizontal scenario, Wang et.al designed a new coding scheme for the homomorphic encryption scheme [39], which greatly reduces the redundancy of the homomorphic encryption scheme. However, the assumption that all parties can decrypt data and calculate in local plaintext is difficult to hold in vertical federated learning.

Therefore, it can not solve the huge bottleneck of encrypted computation in vertical federated learning.

5.2 Network heterogeneity and computing heterogeneity

Heterogeneity is the main bottleneck of cooperative learning. Conventional distributed learning adopts many efficient schemes to eliminate the overhead caused by heterogeneity. One traditional way is to reduce the impact of heterogeneity by increasing the degree of parallelism, such as pipeline [3], or selecting those similar clients or data to prevent heterogeneity ahead [17]. Asynchronous training [18] is another common paradigm. It solves the heterogeneity problem at the cost of introducing the staleness, which damages the convergence performance of training. However, in the typical VFL framework, jobs are limited to be trained in a synchronized way. So common asynchronous optimization comes out of effect. A new paradigm called backup worker was introduced in [6]. Assuming there is $n + b$ worker, backup scheme makes aggregator get the first n gradients and drop the other b gradients as backup. While in vertical federated learning, the parameters transmitted are not only the updated gradients, but also the encrypted intermediate results, which drag the communication the most.

6 CONCLUSION AND FUTURE WORK

In this paper, we dive into the bottleneck of vertical federated learning under homomorphic encryption, and propose a practical and effective optimization system to accelerate current vertical federated learning through backup worker and principal component analysis. The backup scheme improves the efficiency and robustness of the communication network in the heterogeneous scenarios. Principal component analysis alleviates the computation overhead caused by homomorphic encryption. Our system dramatically accelerates current vertical federated learning and promotes its practical deployment in safe cross-industry cooperation. We will further optimize the performance and efficiency of vertical federated learning in the future, mainly in the following four directions.

6.1 AUTO-ML

The underlying algorithm of FATE is completely redeveloped to ensure security. Hence, model hyper-parameters that perform well in traditional distributed learning can not be applied directly. During our experiments, it took us a lot of effort to adjust these hyper-parameters and keep the training performance of VFL almost consistent with local training. Auto-ML [35] can be used to automatically adjust parameters, so that the potential of VFL can be maximized.

6.2 Encryption Optimization

While reduction of computation overhead is significantly apparent, it is at the cost of possible accuracy loss, which is far from perfect. This tradeoff between computational efficiency and model accuracy is related to the tradeoff between privacy preserving and model accuracy for horizontal federated learning, which has been theoretically investigated

by [40]. Considering the computation overhead in VFL is mainly caused by the multiplication of homomorphic encryption, we will further optimize the compression scheme through finding a multiplication-friendly encryption strategy to eliminate computation overhead in the current vertical federated learning without losing security.

6.3 More datasets and metrics

The existing vertical federated learning does not have a typical dataset or standard for people to refer. Lack of datasets and metrics forces researchers in VFL to segment and generate data sets according to their own understanding and compare them with themselves. However, this method of segmentation is probably unscientific, unprofessional, and impractical. A professional and practical vertical dataset provided by specialists in related fields will be of great help.

6.4 Incentive mechanism

The cooperators in VFL training need to have fruitful data features for effective collaborative training. They tend to be competitive enterprises in certain fields, such as industry, finance, medicine, etc. Attracting these enterprises to build collaboration is non-trivial. The incentive mechanism design for VFL is a promising research topic to pave the way for practical VFL. For example, rewarding the model owners based on their marginal contributions can ensure the stability of a federation [19]. And how to measure the reliability and contributions of the cooperators is a sophisticated optimization problem as well [15]. Those important topics researched in conventional FL are equally meaningful in the VFL scenario. We will design a reasonable and efficient incentive mechanism for modern VFL in the future.

ACKNOWLEDGMENTS

This research is supported by National Key Research and Development Program of China under grant number 2020YFB1805500, the Fundamental Research Funds for the Central Universities, and National Natural Science Foundation of China under grant numbers 62032003, 61922017, 61921003.

REFERENCES

- [1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.
- [2] Yoshinori Aono, Takuya Hayashi, Lihua Wang, Shiho Moriai, et al. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13(5):1333–1345, 2017.
- [3] Zhihao Bai, Zhen Zhang, Yibo Zhu, and Xin Jin. Pipeswitch: Fast pipelined context switching for deep learning applications. In *USENIX Symposium on Operating Systems Design and Implementation*, pages 499–514, 2020.
- [4] Borja Balle, Gilles Barthe, Marco Gaboardi, Justin Hsu, and Tetsuya Sato. Hypothesis testing interpretations and renyi differential privacy. In *International Conference on Artificial Intelligence and Statistics*, pages 2496–2506, 2020.
- [5] Zhiqi Bu, Jinshuo Dong, Qi Long, and Weijie J Su. Deep learning with gaussian differential privacy. *Harvard Data Science Review*, 2020(23), 2020.
- [6] Jianmin Chen, Xinghao Pan, Rajat Monga, and Rafal Jozefowicz. Revisiting distributed synchronous sgd. In *International Conference on Learning Representations Workshop Track*, 2016.

- [7] Kewei Cheng, Tao Fan, Yilun Jin, Yang Liu, Tianjian Chen, Dimitrios Papadopoulos, and Qiang Yang. Secureboost: A lossless federated learning framework. *IEEE Intelligent Systems*, 2021.
- [8] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: a real-world web image database from national university of singapore. In *Proceedings of the ACM international conference on image and video retrieval*, pages 1–9, 2009.
- [9] M Everingham, LV Gool, CKI Williams, J Winn, and A Zisserman. The pascal visual object classes challenge results, 2008. URL: <http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2005/results.pdf>, 2015.
- [10] Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Richard Nock, Giorgio Patrini, Guillaume Smith, and Brian Thorne. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *arXiv preprint arXiv:1711.10677*, 2017.
- [11] Hrayr Harutyunyan, Hrant Khachatryan, David C Kale, Greg Ver Steeg, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6(1):1–18, 2019.
- [12] Qirong Ho, James Cipar, Henggang Cui, Seunghak Lee, Jin Kyu Kim, Phillip B Gibbons, Garth A Gibson, Greg Ganger, and Eric P Xing. More effective distributed ml via a stale synchronous parallel parameter server. In *International Conference on Neural Information Processing Systems*, pages 1223–1231, 2013.
- [13] Yaochen Hu, Di Niu, Jianming Yang, and Shengping Zhou. Fdml: A collaborative machine learning framework for distributed features. In *International Conference on Knowledge Discovery & Data Mining*, pages 2232–2240, 2019.
- [14] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3(1):1–9, 2016.
- [15] Jiawen Kang, Zehui Xiong, Dusit Niyato, Shengli Xie, and Junshan Zhang. Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. *IEEE Internet of Things Journal*, 6(6):10700–10714, 2019.
- [16] Jakub Konečný, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.
- [17] Anran Li, Lan Zhang, Juntao Tan, Yaxuan Qin, Junhao Wang, and Xiang-Yang Li. Sample-level data selection for federated learning. In *IEEE International Conference on Computer Communications*, pages 1–10, 2021.
- [18] Xiangru Lian, Wei Zhang, Ce Zhang, and Ji Liu. Asynchronous decentralized parallel stochastic gradient descent. In *International Conference on Machine Learning*, pages 3043–3052, 2018.
- [19] Wei Yang Bryan Lim, Zehui Xiong, Chunyan Miao, Dusit Niyato, Qiang Yang, Cyril Leung, and H Vincent Poor. Hierarchical incentive mechanism design for federated machine learning in mobile networks. *IEEE Internet of Things Journal*, 7(10):9575–9588, 2020.
- [20] Yang Liu, Tao Fan, Tianjian Chen, Qian Xu, and Qiang Yang. Fate: An industrial grade platform for collaborative learning with data protection. *Journal of Machine Learning Research*, 22(226):1–6, 2021.
- [21] Yang Liu, Yan Kang, Xinwei Zhang, Liping Li, Yong Cheng, Tianjian Chen, Mingyi Hong, and Qiang Yang. A communication efficient collaborative learning framework for distributed features. In *International Conference on Neural Information Processing Systems Workshop Track*, 2019.
- [22] Qinyi Luo, Jiaao He, Youwei Zhuo, and Xuehai Qian. Prague: High-performance heterogeneity-aware asynchronous decentralized training. In *International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 401–416, 2020.
- [23] Zeyu Meng, Hongli Xu, Min Chen, Yang Xu, Yangming Zhao, and Chunming Qiao. Learning-driven decentralized machine learning in resource-constrained wireless edge computing. In *IEEE International Conference on Computer Communications*, pages 1–10, 2021.
- [24] Alan Mislove, Hema Swetha Koppula, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. Growth of the flickr social network. In *Proceedings of the first workshop on Online social networks*, pages 25–30, 2008.
- [25] Jer Shyuan Ng, Wei Yang Bryan Lim, Zehui Xiong, Xianbin Cao, Dusit Niyato, Cyril Leung, and Dong In Kim. A hierarchical incentive design toward motivating participation in coded federated learning. *IEEE Journal on Selected Areas in Communications*, 40(1):359–375, 2021.
- [26] Takayuki Nishio and Ryo Yonetani. Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC 2019-2019 IEEE international conference on communications (ICC)*, pages 1–7. IEEE, 2019.
- [27] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 223–238, 1999.
- [28] General Data Protection Regulation. Regulation eu 2016/679 of the european parliament and of the council of 27 april 2016. *Official Journal of the European Union*, 2016.
- [29] Amirhossein Reiszadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics*, pages 2021–2031. PMLR, 2020.
- [30] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 31(9):3400–3413, 2019.
- [31] Aleksei Triastcyn and Boi Faltings. Bayesian differential privacy for machine learning. In *International Conference on Machine Learning*, pages 9583–9592, 2020.
- [32] Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021.
- [33] Su Wang, Mengyuan Lee, Seyyedali Hosseinalipour, Roberto Morabito, Mung Chiang, and Christopher G Brinton. Device sampling for heterogeneous federated learning: Theory, algorithms, and implementation. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2021.
- [34] Jie Xu and Heqiang Wang. Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective. *IEEE Transactions on Wireless Communications*, 20(2):1188–1200, 2020.
- [35] Mengwei Xu, Yuxin Zhao, Kaigui Bian, Gang Huang, Qiaozhu Mei, and Xuanzhe Liu. Federated neural architecture search. *arXiv preprint arXiv:2002.06352*, 2020.
- [36] Qiang Yang, Lixin Fan, and Han Yu. *Federated Learning: Privacy and Incentive*, volume 12500. Springer Nature, 2020.
- [37] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.
- [38] Chen Yu, Hanlin Tang, Cedric Renggli, Simon Kassing, Ankit Singla, Dan Alistarh, Ce Zhang, and Ji Liu. Distributed learning over unreliable networks. In *International Conference on Machine Learning*, pages 7202–7212, 2019.
- [39] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning. In *USENIX Annual Technical Conference*, pages 493–506, 2020.
- [40] Xiaojin Zhang, Hanlin Gu, Lixin Fan, Kai Chen, and Qiang Yang. No free lunch theorem for security and utility in federated learning. *arXiv preprint arXiv:2203.05816*, 2022.
- [41] Wenting Zheng, Ryan Deng, Weikeng Chen, Raluca Ada Popa, Aurojit Panda, and Ion Stoica. Cerebro: A platform for multi-party cryptographic collaborative learning. In *USENIX Security Symposium*, 2021.
- [42] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in neural information processing systems*, 32, 2019.



Dongqi Cai is currently a Ph.D. candidate at the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. His research interests include mobile computing and federated learning.



Mengwei Xu is an Assistant Professor with the Beijing University of Posts and Telecommunications, Beijing, China. His research interests include mobile and edge computing. He received the B.S. and Ph.D. degrees from Peking University, Beijing, China.



of Science and Technology in 2010.

Tao Fan is a Principal Researcher and Tech Lead in the AI Department of WeBank, Shenzhen, China. He is now responsible for FATE, an famous industrial level federated learning open source project. Before joining WeBank, he worked at Baidu, Tencent. He has more than 8 years of experience in large-scale machine learning system and big data application fields. He received his Master degree from University of Science and Technology of China in 2013 and his Bachelor's degree from Shandong University



Yan Kang is currently a research team lead with the AI Department of WeBank, Shenzhen, China. His works focus on the research and implementation of privacy-preserving machine learning and federated learning. His research was authored or coauthored in well-known conferences and journals including IEEE Intelligence Systems, IJCAI, and ACM TIST, and coauthored the Federated Learning book.



journals and conference articles and has received more than 7,000 citations. He has worked at Nokia Research Center and Xerox Research Center Europe. His research includes the well-known Bag of Keypoints image classification method. He has participated in NIPS/NeurIPS, ICML, CVPR, ICCV, ECCV, IJCAI and other top artificial intelligence conferences for a long time, served as area chair of AAAI, and organized workshops in various technical fields. He is also the inventor of nearly one hundred patents filed in the United States, Europe and China, and the chairman of the IEEE P2894 Explainable Artificial Intelligence (XAI) Standard Working Group.

Lixin Fan is the Principal Scientist of Artificial Intelligence at WeBank and the Chairman of the Federal Learning Industry Ecological Development Alliance. His research fields include machine learning and deep learning, computer vision and pattern recognition, image and video processing, 3D big data processing, data visualization and rendering, augmented and virtual reality, mobile computing and ubiquitous computing, and intelligent man-machine interface. He is the author of more than 60 international



Committee on Cloud Computing (2020-). He also served as General Chairs or Program Chairs of 10+ IEEE conferences. He is a Fellow of the IET, and Senior Member of the IEEE. For further information on Dr. Wang, please visit: <http://www.sguangwang.com>

Shangguang Wang is a Professor at the School of Computer Science and Engineering, Beijing University of Posts and Telecommunications, China. He received his Ph.D. degree at Beijing University of Posts and Telecommunications in 2011. He has published more than 150 papers. His research interests include service computing, mobile edge computing, and satellite computing. He is currently serving as Chair of IEEE Technical Committee on Services Computing (2022-2023), and Vice-Chair of IEEE Technical



Transactions on Big Data and ACM Transactions on Intelligent Systems and Technology.

Qiang Yang is Chief Artificial Intelligence Officer of WeBank and Chair Professor of CSE Department of Hong Kong Univ. of Sci. and Tech. He is the Conference Chair of AAAI-21, President of Hong Kong Society of Artificial Intelligence and Robotics (HKSAR), the President of Investment Technology League (ITL) and a former President of IJCAI (2017-2019). He is a fellow of AAAI, ACM, IEEE and AAAS. His research interests include transfer learning and federated learning. He is the founding EIC of two journals: IEEE