Secure Logistic Regression for Vertical Federated Learning

He, Daojing; Du, Runmeng; Zhu, Shanshan; Zhang, Min; Liang, Kaitai; Chan, Sammy

DOI
10.1109/MIC.2021.3138853

Publication date
2022

Document Version
Accepted author manuscript

Published in
IEEE Internet Computing

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MIC.2021.3138853, IEEE Internet Computing

Secure Logistic Regression for Vertical Federated Learning

Daojing He, *Member, IEEE*, Runmeng Du, Shanshan Zhu, Min Zhang, Kaitai Liang, Sammy Chan

***Abstract*—Data island effectively blocks the practical application of machine learning. To meet this challenge, a new framework known as Federated Learning was born. It allows model training on a large amount of scattered data owned by different data providers. This paper presents a parallel solution for computing logistic regression based on distributed asynchronous task framework. Compared to the existing work, our proposed solution does not rely on any third party coordinator, and hence has better security and can solve the multi-training problem. The logistic regression based on homomorphic encryption is implemented in Python, which is used for vertical federated learning and prediction of the resulting model. We evaluate the proposed solution using the MNIST data set, and the experimental results show that good performance is achieved.**

***Index Terms*—Federated learning, multiparty privacy computation, logistic regression, homomorphic encryption.**

I. Introduction

Machine learning (ML) actively seeks regularity and validation rules through the input of massive data coming from various organizations, and finally comes up with suitable models. ML plays a very important role in a variety of applications including web search, online advertising, recommendation, mechanical fault prediction and insurance pricing. However, it has been increasingly difficult from a legislative perspective for different data holders to jointly train models. For example, the EU passed the General Data Protection Regulation (GDPR) Act, which states that all information related to individuals is personal data and the use of such data must be explicitly authorized by the owners. In addition, a lot of laws and regulations about how to protect private data are starting to be published.

Secure Multiparty Computation (MPC) has been proposed to compute encrypted data in model training such as random decision trees, Naive Bayes classification, k-means clustering. These work obviously incur considerable computing overhead. To address this challenge, Google introduced the federated learning (FL) system [1]. The definition of FL is that each party's data is kept locally, without breach of privacy or violation of regulations. Weights obtained by training using each participant's own data are combined to update a global model which could be used for prediction by every participants. Yang et al. [2] expanded the concept of FL to cover more scenarios, forming a comprehensive and secure FL framework including horizontal federated learning (HFL), vertical federated learning (VFL) and federated transfer learning (FTL). As a modeling method to guarantee data privacy, FL has many great application scenarios in sales, finance and other industries.

HFL is applicable to the situation where data sets owned by different parties share the same feature space but have different samples. Each party can train its own local model based on its own data set. Ultimately, all participants upload their model updates to an aggregator, which creates a global model by combining (for example, averaging) the model weights received from the individual participants. By contrast, VFL refers to collaborative scenarios, where individual party does not have a complete data feature matrix or the class labels, so the model needs to be trained by all participants. It should be noted that, on the one hand, private set intersection (PSI) is carried out on sample sets of all participants to create a complete feature vector [3]. On the other hand, in the process of cooperative training, the data feature matrix of each participant will not be disclosed. VFL is suitable for different feature vectors owned by different parties but sharing the same sample userID space, where userID refers to the unique identity of the user.

Thus, vertical federated learning is more complex and requires higher data processing methods. Parties involved in model training (for example, companies in different industries) usually do not hold training data with the same feature dimension, but rather want to use their features to obtain cross-domain model by associating their data with others' data of same samples. This is a practical requirement for vertical distributed datasets. At present, this need is also increasing. Due to the difficulty of implementation and the lack of existing work, VFL needs further research.

Logistic regression (LR) is a classic algorithm in machine learning and it is widely used in finance, Internet and other industries. Logistic function is one of the nonlinear activation functions commonly used in deep learning, and has been widely used in practical applications. Due to simplicity and wide usage in many binary classification tasks of logistic regression, privacy preserving logistic regression modeling based on vertical distributed datasets has attracted a lot of attention. The scheme of [4] is based on secret sharing and homomorphic encryption techniques to encrypt the raw data and distribute it to various computing parties who train the secret model using the encrypted data. In [5], homomorphic

D. He is with East China Normal University, and also with Harbin Institute of Technology, P.R.China (e-mail: hedaojinghit@163.com).

R. Du and S. Zhu are with East China Normal University, Shanghai, P.R. China (e-mail: 52205902012@stu.ecnu.edu.cn).

M. Zhang is with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, P.R.China (e-mail: minzhang@suda.edu.cn).

K. Liang is with the Cybersecurity group, Delft University of Technology, Netherlands (e-mail: Kaitai.Liang@tudelft.nl).

S. Chan is with the Department of Electrical Engineering, City University of Hong Kong, Kowloon, Hong Kong. (e-mail: eeschan@cityu.edu.hk)

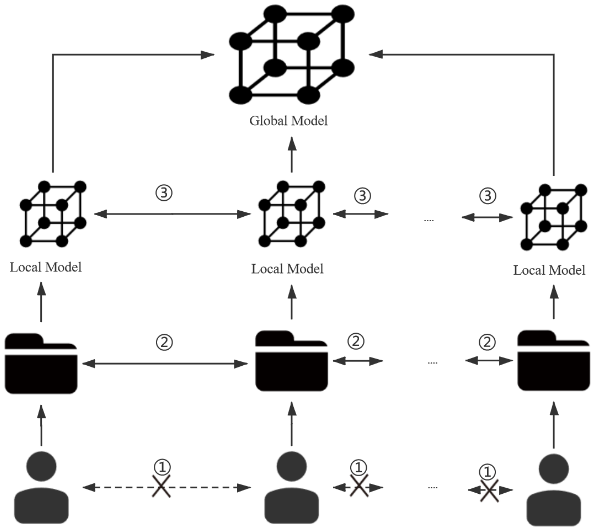

Fig. 1: Data flow in SecureLR

encryption is added to logistic regression, and the gradient information during the training process is transmitted to a third party. There are also some works where participants help each other to decrypt the gradient information which is used to update the model parameters in the training process, while the data is held locally [6]–[8]. Unfortunately, these solutions appear to incur high computational or communicational overheads and data insecurity of parties.

This paper presents a distributed solution for computing logistic regression avoiding disadvantages of the above mentioned schemes, called SecureLR. In SecureLR, the third party is removed and only the participants are retained, as shown in Fig. 1, where ① represents plaintext data that cannot be directly exchanged between participants, ② represents sample alignment and ③ represents privacy protection model training. There are two main reasons of removing the third party. First, it is hard to find a trustworthy third party. Second, the involvement of a third party increases the risk of data leakage, complexity of the system and the cost of establishing the model.

We address the challenge of how can the participants jointly build a model while the third party is removed. In our implementation, we consider to build a distributed system based on homomorphic encryption. There are two main types of participants involved in SecureLR. The first type owns features and labels and is called Guest. The second type only owns feature and is called Host. Guest and Host want to jointly build a machine learning model. However, since Host has no label, Guest will assist Host in gradient decryption and gradient update during model training.

The main contributions of this paper can be summarized as follows:

1) We design a privacy preserving logistic regression training scheme based on homomorphic encryption in vertical federated learning. The proposed scheme based on piecewise function improves security at an acceptable loss of efficiency. It also avoids the shortcomings of existing schemes, including not leaking label of party, protecting gradient information during training, and significantly protecting data features of each party.
2) In the process of logistic regression model training, we avoid using Taylor expansion formula and directly use new logical functions to ensure loss function unchanged. This way ensures the accuracy of the model. At the same time, the security modeling of parties can be completed without the coordination by the third party, which greatly reduces the complexity of the system.
3) We propose a multi-party VFL framework without the third party to solve multi-party logistic regression problem. The framework can deal with multi-participant model training problems.

The rest of this paper is organized as follows. Section II introduces related work of federated learning. Section III introduces related preliminaries. Section IV shows the framework of our proposed multiparty VFL system. Section V gives the concrete structure of vertical federated learning. Section VI shows our algorithm principle. Section VII evaluates the designed algorithm. Section VIII concludes the paper.

II. Related Work

Federated learning is a new machine learning mechanism which can train a model based on a large corpus of scattered data owned by different parties and maintain the data privacy at the same time. However, federated learning raises several types of issues, including the system challenges (e.g., a massive number of edge clients with limited network connections) [9], the statistic challenges (e.g., unbalanced and non-IID data distributions) [10], and the data privacy preservation [11], which have attracted a lot of recent research interests.

Logistic regression is a classic algorithm in machine learning. Our work belongs to privacy preserving machine learning for federated learning. At present, there are three types of privacy preserving logistic regression schemes including joint modeling using homomorphic encryption or secret sharing MPC technique [4], transmitting gradient information to the third party [5] during model training and Guest providing help for Host to decrypt gradient information [6]–[8].

However, the scheme of [4] based on homomorphic encryption or secret sharing MPC technique will take data out of the local area, and increase the risk of data leakage. Transferring gradient information to a third party who provides aggregation and distribution service will result in more frequent communication rounds [5]. In addition, though gradient information is encrypted and transmitted to a third party, it cannot be stored locally, thus increasing the potential risk of data leakage. This is because the third party may conspire with a participant to cause the data of other participants to be leaked. At the same time, in practice, it is almost impossible to find a completely trustworthy third party. This makes it challenging to apply this solution in a real production environment.

Instead of transmitting encrypted data to a third party, some intermediate results could be encrypted with homomorphic encryption and transmitted during the training process. This brings some significant advantages. First, the raw data are kept locally by each party. Second, the amount of data to be encrypted is minimized through careful design. Third, the overall computation overhead could be reduced greatly. In this research direction, Hardy et al. [6] proposed a solution for federated logistic regression based on vertically partitioned data. Yang et al. [7] presented a solution for parallel distributed logistic regression for VFL, and the role of third-party coordinator is removed as in our proposed solution. Wei et al. [8] proposed a protocol that can complete the logistic regression modeling of vertically partitioned data by asynchronous gradient sharing, However, schemes of [7], [8] have the risk of leaking information because participants can recover feature information by constructing a large number of linear equations [12]. The works of [7], [8] are most similar to the research content of this paper. The differences between the work of [7], [8] and ours lie in the difference of logical functions, the computation method of gradient and the different decryption ways of gradient information.

III. Background Knowledge and Security Model Architecture

A. Vertical Federated Learning

VFL refers to multiple data sets where users overlap more and user features overlap less, which is applicable to users who share the same sample in data sets owned by different parties. For example, if there are two different organizations, one is a bank in one place and the other is an e-commerce company in the same place. Their user base is likely to include most of the residents of the area, so the intersection of users is large. However, since banks only record users' payment behavior and credit ratings, while e-commerce companies keep users' browsing and purchase history, there is less intersection between their user features. VFL is a federated learning that aggregates these different features in encrypted state to enhance model capabilities.

Consider there are m participats in VFL, that is, one Guest and $m-1$ Hosts are defined. Let $\{\mathbf{X}^k \in \mathbb{R}^{n \times d_k}\}_{k=1}^m$ be the feature matrix distributed on m private parties with each row $\mathbf{X}_i^k \in \mathbb{R}^{1 \times d_k}$ being a userID data instance, where $n_1, \ldots, n_m$ must be the same userID set. This is because they perform Private Set Intersection (PSI) [13]. Let $\mathcal{F}^k = \{f_1, \ldots, f_{d_k}\}$ denote the feature set of the corresponding feature matrix, where $\mathcal{F}^p \cap \mathcal{F}^q = \emptyset, \forall p \neq q \in \{1, \ldots, m\}$. This is determined by the characteristic of VFL. Also when building a model for a common task, VFL considers that Guest has a label for classification or regression. Let $\mathbf{y} \in \mathbb{R}^{1 \times d_k}$ denote the label matrix.

VFL can be expressed follows.

Given: The Guest owns the feature matrix and the label matrix, while the Hosts only owns the feature matrix.

Learn: A machine learning model $Model$, in which data matrix information of any party is not provided to others during the learning process. The model $Model$ has a function that projects $Model_i$ on each side, and $Model_i$ takes input from its own feature X_i.

Lossless and Eiffient Constraint: We require the the model $Model$ to ensure the efficiency of execution without loss of precision. In this paper, $Model$ is a logistic regression model. Here we do not consider the situation where some of the Hosts (clients) is missing or corrupted.

B. Secure Multiparty Computation and Security Modle

Secure Multiparty Computation. The problem that MPC solves is that in the environment of n participants, each participant has its own private input x_i. Then they work together to compute a function $f(x_1, \ldots, x_n)$. MPC ensures the independence of the input and the correctness of the computation. In the end, each participant cannot get more information other than the output. Most previously proposed protocols are based on the security protocol under the semi-honest model. This model assumes that all participants in the MPC protocol will faithfully execute the protocol, but they will record the information received during the protocol execution and attempt to use these information to infer privacy input of other participants after the protocol execution [14]. The semi-honest model is a less secure model, but it accurately depicts many real application scenarios. The security protocol under the semi-honesty model is the key and foundation of designing privacy protection protocol.

Semi-honest model. Semi-honest model is an important MPC model in which every participant's behavior is consistent with the requirements of the algorithm. However, it will retain information about the computation process and try to use these information to obtain more private information about other participants

Homomorphic encryption. It is a cryptographic technique based on computational complexity theory. Processing the homomorphic encrypted data to get an output, and decrypting this output, we can find that the decryption result is the same as the output from processing unencrypted raw data in the same way. Paillier cryptosystem has additive homomorphism [15], that is, $E(m_1) + E(m_2) = E(m_1 + m_2)$ and $E(m_1)m_2 = E(m_1 m_2)$, where m_1, m_2 are plain message. Paillier cryptosystem has three stages including Key generation (pk, sk), Encryption $E(\cdot)$ and Decryption $D(\cdot)$.

C. Logistic Regression

Logistic regression model is a classification model, which is widely used in machine learning. Machine learning often uses the logistic function to achieve this purpose: $p(\mathbf{y} = 1 | \mathbf{X}; \mathbf{W}) = h_w(\mathbf{X}) = \frac{1}{1+e^{-\mathbf{WX}}}$.

In the setting of VFL of m parties, assuming data matrix $\{\mathbf{X}^k \in \mathbb{R}^{n \times d_k}\}_{k=1}^m$, label matrix $\mathbf{Y} \in \mathbb{R}^{n \times 1}$ and parameter matrix $\{\mathbf{W}^k \in \mathbb{R}^{d_k \times 1}\}_{k=1}^m$ corresponding to the feature matrix, where $\mathbf{X}_i \in \mathbb{R}^{1 \times \Sigma_{k=1}^m d_k}$ is a complete user data instance.

The loss function is $L(\mathbf{W}) = -\frac{1}{n}\sum_{i=1}^n \mathbf{Y}_i \log h_w(\mathbf{X}_i) + (1 - \mathbf{Y}_i)\log(1 - h_w(\mathbf{X}_i))$, where $\frac{\partial L_i(\mathbf{W})}{\partial \mathbf{W}} = (h_w(\mathbf{X}_i) - \mathbf{Y}_i)\mathbf{X}_i$. And the gradients formula is $\frac{\partial L(\mathbf{W})}{\partial\, \mathbf{W}} = -\frac{1}{n}\sum_{i=1}^n \frac{\partial L_i(\mathbf{W})}{\partial \mathbf{W}} = \frac{1}{n}\sum_{i=1}^n (h_w(\mathbf{X}_i) - y_i)\mathbf{X}_i$. In the setting of VFL of m parties, assuming data matrix $\{\mathbf{X}^k \in \mathbb{R}^{n \times d_k}\}_{k=1}^m$, $\mathbf{y} \in \mathbb{R}^{n \times 1}$, model parameters $\{\mathbf{W}^k \in \mathbb{R}^{1 \times d_k}\}_{k=1}^m$ corresponding to the data matrix, and $\mathbf{X}_i \in \mathbb{R}^{1 \times \Sigma_{k=1}^m d_k}$ is a complete userID data instance.

D. Gradient Descent

Stochastic gradient descent (SGD) is an effective approximation algorithm for approaching a local minimum of a loss function, step by step. In addition, SGD can be generalized to work for logistic regression and neural network training, where no closed-form solution exists for the corresponding optimization problems. As a result, SGD is the most commonly used approach to train such models in practice and the main focus of this work. The SGD algorithm works as follows: $\mathbf{W}$ is initialized as a vector of random values. In each iteration, a complete userID data instance $(\mathbf{X}_i, y_i)$ is selected randomly and $\mathbf{W}$ is updated as $\mathbf{W} = \mathbf{W} - \eta\frac{\partial L_i(\mathbf{W})}{\partial \mathbf{W}}$.

E. Piecewise Function

In addition to solve the problem of linear regression, mainly the extra challenge of logistic regression algorithm is to compute logistic function, where function $h_w(\mathbf{X}_i)$ involving division and exponentiation are difficult to use the MPC technique to solve. Therefore, previous work used Taylor expansion polynomials to approximate the function $h_w(\mathbf{X}_i)$, and it has been shown that the approximation using a higher degree polynomial is very accurate [16]. However, for efficiency reasons, the degree of approximation polynomial is set to 2 or 3 in the secure computation. As a result, the accuracy loss of the training model is greater than that of logistic regression. The piecewise function is that if $u < -1/2$, $f(u) = 0$; If $u > 1/2$, $f(u) = 1$; Otherwise, $f(u) = u + 1/2$. This piecewise function can not only perform efficient computation, but also greatly protect the privacy of data.

In addition, with the new logistic function, there are two options when computing the back propagation. First, we can use the same update function as the logistic function, that is, we can continue to use the logistic function to compute the partial derivatives. Second, we can compute the partial derivatives of the new function. We are in line with [17], that is, we continue to use the logistic function to compute partial derivatives. This is because the accuracy of the first method matches that of using the logistic function. Based on piecewise function and participant pattern for VFL (one Guest and multiple Hosts), we can derive that if $\sum_{k=1}^{m}(\mathbf{X}_i^k\mathbf{W}^k) < -1/2$, $f(\mathbf{XW}_i) = 0$; If $\sum_{k=1}^{m}(\mathbf{X}_i^k\mathbf{W}^k) > 1/2$, $f(\mathbf{XW}_i) = 1$; Otherwise, $f(\mathbf{XW}_i) = \frac{1}{2} + \mathbf{XW}_i$. In addition, [17] proves that the piecewise function $f(\mathbf{XW}_i)$ achieves almost the same accuracy as the logistic function $h_w(\mathbf{X}_i)$.

The main reason that logistic regression algorithms work well for classification problems is that the predicted range is between 0 and 1. Therefore, it is very important that the two tails of the activation function converge to 0 and 1. Both the logistic function $h_w(\mathbf{X}_i)$ and the function $f(\mathbf{XW}_i)$ would converge to 0 and 1. When $\mathbf{XW}_i$ approaches negative infinity, $h_w(\mathbf{X}_i)$ converges to 0, and $f(\mathbf{XW}_i)$ is equal to 0. When $\mathbf{XW}_i$ approaches positive infinity, $h_w(\mathbf{X}_i)$ converges to 1 and $f(\mathbf{XW}_i)$ is equal to 1. In contrast, the approximation of a low-order Taylor expansion polynomial may approach a logistic function over a certain interval, but the tail is unbounded. In addition, [17] proves that the piecewise function $f(\mathbf{XW}_i)$ achieves almost the same accuracy as the logistic function $h_w(\mathbf{X}_i)$.

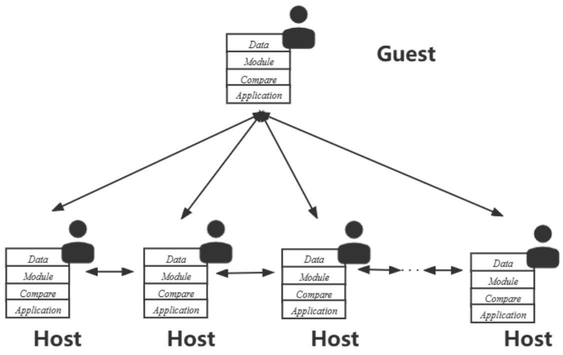

Fig. 2: System Framework

F. The blind millionaire problem

According to $f(\mathbf{XW}_i)$, it is noted that the millionaire problem is involved in the secure logistic regression model training. The millionaire problem refers to that two millionaires Alice and Bob have wealth t_1 and t_2, respectively, and they want to secretly determine the size relationship between t_1 and t_2. Sometimes it is necessary to make a secure comparison of the size of the corresponding inscription data when participants are only aware of the encrypted data. We call it the blind millionaire problem and it has higher security requirements than the millionaire problem. We construct a solution to solve blind millionaire problem by improving the method of [18]. Based on the piecewise function, the blind millionaire can be defined as: if $E(\sum_{k=1}^{m}(\mathbf{X}_i^k\mathbf{W}^k)) < E(-1/2)$, $f(\mathbf{XW}_i) = 0$; If $E(\sum_{k=1}^{m}(\mathbf{X}_i^k\mathbf{W}^k)) > E(1/2)$, $f(\mathbf{XW}_i) = 1$; Otherwise, $f(\mathbf{XW}_i) = E(\frac{1}{2} + \mathbf{XW}_i)$.

IV. System Framework

Fig. 2 shows the framework of our proposed multiparty VFL system, which consists of a Guest and multiple Hosts. Any two can communicate with each other. Multiparty VFL framework consists of four layers. Starting from the bottom layer, the first layer is the Data layer $Data$, which integrates MySQL database. MySQL is used to store original Data. We designed a unified communication model to facilitate subsequent reconstruction and optimization. The second layer $Module$ contains library methods such as pandas and numpy of Python, phe for encrypting and decrypting data, and gmpy for handling large integer operations. The third layer is the computation layer $Compare$. We extracted the low-level abstraction of the blind millionaire problem and implemented the Engine respectively to facilitate the application layer algorithmn invocation. The fourth layer is $Application$, which is mainly secure logistic regression model training, where secure logistic regression model training can call computation layer $Compare$ to get corresponding result. Distributed framework adopts multi-threading, based on Python language implementation, to better support high concurrency. SecureLR is implemented in the framework. In Fig. 2, next to the connections between guest and hosts, four layers involved in the VFL can not be incorporated because the layers are independent of each other. The piecewise function $f(\mathbf{XW}_i)$ is the loss function used in the implemented framework. The arrows indicate that participants can communicate with each other.

V. OVERVIEW OF PROPOSED METHOD

In this section, we give the concrete structure of SecureLR. According to the architecture of VFL, the steps of SecureLR should include initialization, data preparation and privacy protection training. A brief overview follows.

Initialization: During initialization, the public parameters are generated by the Guest. The Guest also generates secret key pairs and publishes the public key to Hosts and keep the private key locally. In addition, Guest and Hosts initialize parameter matrix $\{\mathbf{W}^k \in \mathbb{R}^{1\times d_k}\}_{k=1}^m$.

Data preparation: Before the model training, Guest and Hosts encrypt the data $\mathbf{X}^k\mathbf{W}^k$ and then call the blind millionaire to get the corresponding comparison matrix $f(\mathbf{XW})$.

Privacy protection model training: The Guest owns the feature matrix and the label matrix, while the Hosts only own the feature matrix. The logistic regression model M has a function that projects M_i on each side, and M_i takes input from its own feature matrix $\mathbf{X}_{i*}^k \in \mathbb{R}^{1\times d_k}$. In the process of model training, Guest and Hosts interact and communicate with each other, and finally all participants get the respective local model.

VI. DETAILED DESIGN

The process of model training is that, first, we need to define a loss function and get a prediction matrix based on forward propagation according to corresponding comparison matrix. Comparing with the real sample to get the loss value, we use the back propagation to update the weight (parameter), iterate back and forth until the loss function is very small and accuracy rate can reach the ideal value. The parameters in this case are the parameters required by the model. Privacy protection model training is as follows.

1) Guest and Host$^k, \in [2, m]$ encrypt the data $\mathbf{X}^k\mathbf{W}^k$ and then call the blind millionaire algorithm to get the corresponding comparison matrix, that is a prediction matrix $f(\mathbf{XW})$.
2) Then Guest computes loss value $\bigtriangledown\mathbf{Y} = f(\mathbf{XW}) - \mathbf{Y}$ by comparing $f(\mathbf{XW}_i)$ to $\mathbf{Y}_i$. At this point, Guest can compute gradient value $\frac{\partial L(\mathbf{W}^1)}{\partial \mathbf{W}^1}$ and update model parameters $\mathbf{W}^1 \in \mathbb{R}^{1\times d_1}$.
3) Guest computes $E(\bigtriangledown\mathbf{Y})$ and sends it to the Hosts so that Hosts do not know the loss value. Hosts can compute own gradient value $E(\frac{\partial L(\mathbf{W}^k)}{\partial \mathbf{W}^k})$ and update model parameters $\{E(\mathbf{W}^k) \in \mathbb{R}^{1\times d_k}\}_{k=2}^m$.
4) Model training ends until the maximum number of iterations is reached or some convergence conditions are satisfied. Guest sends a signal that the model iteration is terminated. Host$^k, \in [2, m]$ randomly selects a matrix $\mathbf{R}^k \in \mathbb{R}^{d_k\times 1}$, computes $E(\mathbf{W}^k\mathbf{R}^k) = E(\mathbf{W}^k)\mathbf{R}^k$ and sends it to Guest. Guest computes $\mathbf{W}^k\mathbf{R}^k = D(E(\mathbf{W}^k\mathbf{R}^k))$ and send it to Host$^k, k \in [2, m]$. Finally, Host$^k, k \in [2, m]$ computes $\mathbf{W}^k = \mathbf{W}^k\mathbf{R}^k/\mathbf{R}^k$.
5) Guest and Host$^k, \in [2, m]$ return the parameter matrix $\mathbf{W}^k$ respectively.

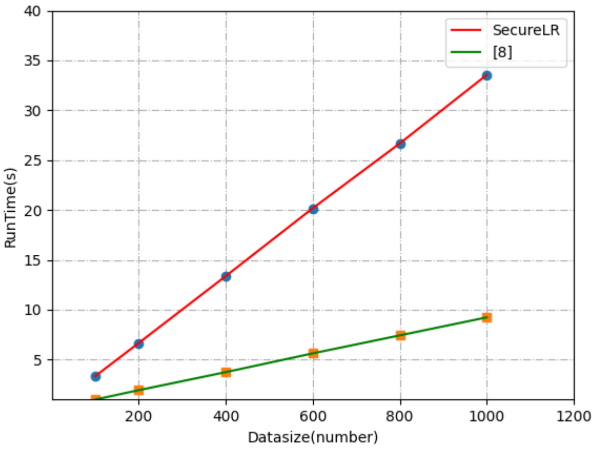

Fig. 3: Results Comparison between SecureLR and [8]

The above shows the main steps of our proposed privacy protection model training.

VII. EVALUATION

We have set up a simple two-party VFL framework. All the algorithms mentioned in this paper are implemented in the framework. The framework focuses on the improvement of data privacy protection, and the improvement is very obvious.

Experimental equipment include two PCs based on x64 processor with 16.0 GB RAM (15.4GB available), Windows10 64-bit operating system and AMD Ryzen 5 5500U with Radeon Graphics 2.10 GHz. The algorithms are deployed by running programs on the two experimental machines respectively. The algorithms are implemented using Python language and related libraries MySQL, and other related tools.

Dataset is MNIST. Since the labels of MNIST dataset are multi-classification labels and this paper deals with the dichotomous problem, we use the tag value after the tag data is dichotomized as the actual value $\mathbf{Y}$. The sampling interval is 100, 200, 400, 600, 800 and 1000. The abscissa is the number of samples and the ordinate is the running time of the algorithm. In terms of security, we are more secure than traditional logistic regression schemes. Data security is protected, data islands are broken, and data availability is improved. Compared with other vertical logistic regression federated learning frameworks, on the one hand, our scheme eliminates the third party and greatly protects the data privacy of the participants, namely, Guest and Host.

We compare SecureLR to [8], which solved a two-party logistic regression in VFL. For fair comparison, we uniform the parameters. Threshold $l = 0.01$. Learning rate $\eta = 0.3$. Participant pattern: one Guest and one Host. The comparison results are shown in Fig. 3.

Fig. 3 shows that the efficiency of [8] is higher than SecureLR. Our computational load is mainly consumed in the

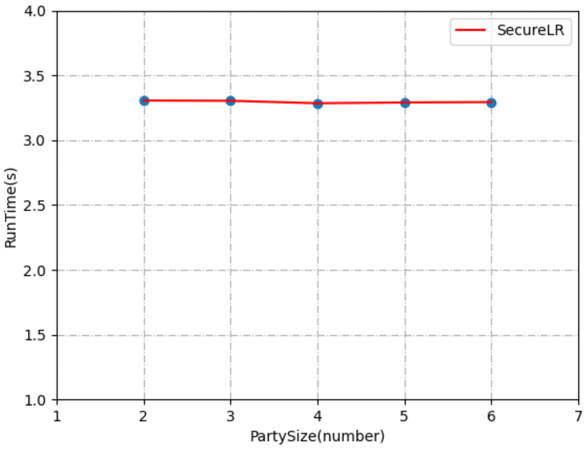

Fig. 4: Experimental results of SecureLR

computation of the comparison matrix, as in [8], which is directly transmitted to Guest with product matrix in plaintext. Since Guest has label matrix, Guest can completely recover the data information of Host by constructing a large number of linear equations. SecureLR guarantees security at the expense of efficiency. However, Fig. 3 shows that our efficiency loss is within acceptable limits. Currently, trading off data security with some acceptable costs is an inevitable trend in the data field.

Since SecureLR mainly solves the problem of multi-party training, in our experiment each participant has a dataset of size 100 and the number of participants is 2, 3, 4, 5 and 6. The experimental results are shown in Fig. 4.

Fig. 4 shows that the prediction time of SecureLR remain the same with the growth in the number of participants under the condition that our algorithm holds the dataset of invariant size.

VIII. Conclusion

The goal of federated learning is to achieve common modeling while ensuring data privacy and legal compliance. This paper has presented a parallel distributed logistic regression algorithm for vertical federated learning. Unlike existing solutions, we not only remove the third party coordinator from the system, but also ensure that there is no information leakage during the training process. This system design brings some significant advantages, such as avoiding the practical difficulty of finding an authoritative third party coordinator that all participants can trust, and improving the security of system deployment. We have evaluated the implementation of the system, and the experimental results demonstrated that the system has high security. In addition, our implementation can be easily extended to support federated model training with multiple participants. In the future research, we will study the vertical federated learning method combining logistic regression and deep neural network.

IX. Acknowledgment

This research is supported by the National Natural Science Foundation of China (Grants: U1936120, U1636216), the National Key R&D Program of China (2017YFB0802805 and 2017YFB0801701), the Fok Ying Tung Education Foundation of China (Grant 171058), the Basic Research Program of State Grid Shanghai Municipal Electric Power Company (52094019007F), and the University Grants Committee of the Hong Kong Special Administrative Region of China (CityU 11201421). Daojing He is the corresponding author of this article.

References

[1] J. Konecný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *CoRR*, vol. abs/1610.05492, 2016. [Online]. Available: http://arxiv.org/abs/1610.05492

[2] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 12:1–12:19, 2019. [Online]. Available: https://doi.org/10.1145/3298981

[3] Z. Wang, K. Banawan, and S. Ulukus, "Multi-party private set intersection: An information-theoretic approach," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 1, pp. 366–379, 2021. [Online]. Available: https://doi.org/10.1109/JSAIT.2021.3057597

[4] A. Gascón, P. Schoppmann, B. Balle, M. Raykova, J. Doerner, S. Zahur, and D. Evans, "Privacy-preserving distributed linear regression on high-dimensional data," *Proc. Priv. Enhancing Technol.*, vol. 2017, no. 4, pp. 345–364, 2017. [Online]. Available: https://doi.org/10.1515/popets-2017-0053

[5] Y. Fan, J. Bai, X. Lei, Y. Zhang, B. Zhang, K. Li, and G. Tan, "Privacy preserving based logistic regression on big data," *J. Netw. Comput. Appl.*, vol. 171, p. 102769, 2020. [Online]. Available: https://doi.org/10.1016/j.jnca.2020.102769

[6] S. Hardy, W. Henecka, H. Ivey-Law, R. Nock, G. Patrini, G. Smith, and B. Thorne, "Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption," *CoRR*, vol. abs/1711.10677, 2017. [Online]. Available: http://arxiv.org/abs/1711.10677

[7] S. Yang, B. Ren, X. Zhou, and L. Liu, "Parallel distributed logistic regression for vertical federated learning without third-party coordinator," *CoRR*, vol. abs/1911.09824, 2019. [Online]. Available: http://arxiv.org/abs/1911.09824

[8] Q. Wei, Q. Li, Z. Zhou, Z. Ge, and Y. Zhang, "Privacy-preserving two-parties logistic regression on vertically partitioned data using asynchronous gradient sharing," *Peer Peer Netw. Appl.*, vol. 14, no. 3, pp. 1379–1387, 2021. [Online]. Available: https://doi.org/10.1007/s12083-020-01017-x

[9] F. Sattler, S. Wiedemann, K. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *CoRR*, vol. abs/1903.02891, 2019. [Online]. Available: http://arxiv.org/abs/1903.02891

[10] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020.* OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=HJxNAnVtDS

[11] T. Li, B. Li, X. Chen, Z. Liu, and T. Hou, "Npmml: A framework for non-interactive privacy-preserving multi-party machine learning," *IEEE Transactions on Dependable and Secure Computing*, vol. PP, no. 99, pp. 1–1, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8981947/

[12] Y. Li, Y. Bao, L. Xiang, J. Liu, C. Chen, L. Wang, and X. Wang, "Privacy threats analysis to secure federated learning," *CoRR*, vol. abs/2106.13076, 2021. [Online]. Available: https://arxiv.org/abs/2106.13076

[13] P. H. Le, S. Ranellucci, and S. D. Gordon, "Two-party private set intersection with an untrusted third party," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, L. Cavallaro, J. Kinder, X. Wang, and J. Katz, Eds. ACM, 2019, pp. 2403–2420. [Online]. Available: https://doi.org/10.1145/3319535.3345661

[14] D. Butler, D. Aspinall, and A. Gascón, "Formalising oblivious transfer in the semi-honest and malicious model in crypthol," in *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2020, New Orleans, LA, USA, January 20-21, 2020*, J. Blanchette and C. Hritcu, Eds. ACM, 2020, pp. 229–243. [Online]. Available: https://doi.org/10.1145/3372885.3373815

[15] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, ser. Lecture Notes in Computer Science, J. Stern, Ed., vol. 1592. Springer, 1999, pp. 223–238. [Online]. Available: https://doi.org/10.1007/3-540-48910-X_16

[16] R. Livni, S. Shalev-Shwartz, and O. Shamir, "On the computational efficiency of training neural networks," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., 2014, pp. 855–863. [Online]. Available: https://proceedings.neurips.cc/paper/2014/hash/3a0772443a0739141292-a5429b952fe6Abstract.html

[17] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," in *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*. IEEE Computer Society, 2017, pp. 19–38. [Online]. Available: https://doi.org/10.1109/SP.2017.12

[18] X. Liu, K. R. Choo, R. H. Deng, R. Lu, and J. Weng, "Efficient and privacy-preserving outsourced calculation of rational numbers," *IEEE Trans. Dependable Secur. Comput.*, vol. 15, no. 1, pp. 27–39, 2018. [Online]. Available: https://doi.org/10.1109/TDSC.2016.2536601

Daojing He (S'07-M'13) received the B.Eng.(2007) and M. Eng. (2009) degrees from Harbin Institute of Technology (China) and the Ph.D. degree (2012) from Zhejiang University (China). He is currently a professor at East China Normal University, P.R. China. His research interests include network and systems security.

Runmeng Du was born in 1994. She is currently a PhD student in the School of Software Engineering, East China Normal University, P.R. China.

Shanshan Zhu was born in 1985. She is currently working at East China Normal University, P.R. China. She received the Bachelor degree from Huazhong University of science and technology (China) and the Master degree from Zhejiang University (China) , in 2007 and 2010, respectively.

Min Zhang received the bachelor's and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 1991 and 1997, respectively. He is currently a Distinguished Professor at Harbin Institute of Technology, Shenzhen, P.R.China. His current research interests include natural language processing and artificial intelligence.

Kaitai Liang received the Ph.D. degree in computer science from the City University of Hong Kong. He is currently an Assistant Professor with the Department of Intelligent Systems, Delft University of Technology, The Netherlands. His current research interests include applied cryptography, data security, and privacy-enhancing technology.

Sammy Chan (S'87-M'89) received a Ph.D. degree in communication engineering from the Royal Melbourne Institute of Technology, Australia, in 1995. Since December 1994 he has been with the Department of Electrical Engineering, City University of Hong Kong, where he is currently an associate professor.