Hindawi
Security and Communication Networks
Volume 2022, Article ID 5094830, 12 pages
https://doi.org/10.1155/2022/5094830

Research Article

Privacy-Preserving Vertical Collaborative Logistic Regression without Trusted Third-Party Coordinator

Xiaopeng Yu,[1] **Wei Zhao**,[1] **Dianhua Tang**,[1,2] **and Kai Liang**[3]

[1]*Science and Technology on Communication Security Laboratory, Chengdu 610041, China*
[2]*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*
[3]*State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China*

Correspondence should be addressed to Dianhua Tang; tangdianhua86@163.com

Received 28 July 2022; Accepted 13 September 2022; Published 12 October 2022

Academic Editor: Jianbo Du

Collaborative learning is an emerging distributed learning paradigm, which enables multiple parties to jointly train a shared machine learning (ML) model without causing the disclosure of the raw data of each party. As one of the fundamental collaborative learning algorithms, privacy-preserving collaborative logistic regression has recently gained attention from industry and academia, which utilizes cryptographic techniques to securely train joint logistic regression models across data from multiple parties. However, existing schemes have high communication and computational overhead, lose the ability to deal with high-dimensional sparse samples, cut down the accuracy of the model, or exist the risk of leaking private information. To overcome these issues, considering vertically distributed data, we propose a privacy-preserving vertical collaborative logistic regression (P^2 VCLR) based on approximate homomorphic encryption (HE), which enables two parties to jointly train a shared model without a trusted third-party coordinator. Our scheme utilizes batching method in approximate HE to encrypt multiple data into a single ciphertext and enable a parallel processing through single instruction multiple data (SIMD) manner. We evaluate our scheme by using three publicly available datasets, the experimental results indicate that our scheme outperforms existing schemes in terms of training time and model performance.

1. Introduction

Machine learning (ML) [1] is a method for analyzing large-scale data and is widely used in practice to train predictive models for practical applications. As one of the basic machine learning algorithms, logistic regression (LR) [2] has attracted much attention for its powerful ability to solve classification problems in practical applications, such as disease diagnosis [3], credit evaluation [4].

In recent years, in order to obtain massive data for training better-performing models [5], there is growing interest in machine learning by combining the data from different institutions [6]. For instance, different hospitals would like to combine health data to jointly train models to facilitate more accurate disease diagnosis; different financial companies want to collaborate to train more effective credit card scoring and fraud detection models. Unfortunately, due to regulatory and competitive reasons, it is difficult or even impossible to directly exchange data of different parties for model training in practice [7]. That is, the data of different organizations is isolated. To eliminate the issue of "data isolation", the idea of collaborative learning [8] is introduced. Its goal is to cooperatively train a shared ML model on distributed data while complying with regulation and protecting privacy. The security, privacy, and efficiency concerns remain main challenges for practical applications. Recently, as a fundamental collaborative learning algorithm, privacy-preserving collaborative logistic regression (PPCLR) [9–24] has received considerable attention recently, which utilizes cryptographic primitives such as homomorphic encryption (HE) algorithm [25] and multi-party computation (MPC) protocol [26] to securely train a joint logistic regression model across data from multiple parties. However, for the HE-based schemes [9–11], model weights are exposed to all parties at each iterative update of global model parameters during training, which is able to be utilized to deduce additional private information [27]; for the MPC-based schemes [12, 14], after using secret sharing (SS) [28] on training samples of all parties, even previously sparse samples become dense, so they are not able to efficiently handle sparse samples and require high communication complexity when the training data becomes large.

To solve the problems mentioned above, in a two-party setting, considering two vertically distributed training data with the same sample distributions but different feature distributions, we construct a privacy-preserving vertical collaborative logistic regression (P^2 VCLR) based on the HE for arithmetic of approximate numbers [29]. The main contributions are as follows:

(1) Firstly, we construct a P^2 VCLR framework for collaborative learning of vertical distributed features, which can securely realize the joint modeling of both parties without the assistance of a trusted third-party (TTP), and hence greatly reduces the system complexity.

(2) Secondly, to improve the training efficiency, using the batching technique in HE [29], the proposed scheme can pack multiple samples into a single plaintext with multiple slots, encrypt it into a single ciphertext, and enable a parallel computing through using SIMD.

(3) Finally, we conduct performance evaluations on three datasets [30], and the experimental evaluation results indicate that our scheme achieves a significant improvement in efficiency and performance than existing schemes [9, 21]. Specifically, the training time of the model is decreased by almost 32.3%-72.5%; the accuracy, F1-score, and AUC of the model have nearly 0.3% - 3.0%, 0.1% - 2.7% and 0 - 0.03 improvement, respectively. Furthermore, the security analysis indicates that the proposed P^2 VCLR scheme is secure against semi-honest adversaries, and neither of the both parties can know each other's raw data.

The rest of this work is arranged as follows. Several works related to our scheme are introduced in Section [2]. In Section [3], we review some preliminaries. In Section [4], our scheme is described. In Section [5], the evaluations for our scheme are presented. The security analysis of our scheme is shown in Section [6]. In Section [7], we conclude this work.

2. Related Works

There are several works that have been made to joint train a LR model across multiple data owners. In general, a common approach is to implement secure logistic regression by using cryptographic primitives like HE [25] and MPC [26]. The existing works [9–24] can be divided into two categories: PPCLR with a TTP coordinator [9–16] and PPCLR without a TTP coordinator [17–24]. A summary of the existing works [9–24] follows.

As for the PPCLR with TTP coordinator [9–16], Hardy *et al.* [9] described a privacy-preserving federated LR scheme by employing additively HE scheme [25], which centralizes two vertically distributed training data in one TTP coordinator, but the approximation of non-polynomial function reduces the model accuracy. Yang *et al.* [10] shown a quasi-Newton way for achieving vertical federated LR model based on the additively HE scheme [25]. Using an additive HE [31] and an aggregation method [32], Mandal *et al.* [11] built a privacy-preserving regression analysis protocol on the horizontally distributed high-dimensional data. Employing an additive secret sharing technique [33], Zhang *et al.* [12] proposed a privacy-preservation collaborative learning for ensuring local training data and model information. Liu *et al.* [13] introduced a collaborative learning platform, which supports multiple institutions to build machine learning models collaboratively over large-scale horizontally and vertically partitioned data. By means of MPC from additive secret sharing [34, 35], Cock *et al.* [14] proposed a protocol for securely training LR model over distributed parties, where TTP initializer assigns relevant random values to two computing severs. Based on multi-key fully HE [36], Wang *et al.* [15] designed a secure cloud-edge collaborative LR system, which employs the cloud centre and edge nodes to collaboratively train a LR model over encrypted data. Zhu *et al.* [16] proposed a value-blind LR training method in a collaborative setting based on HE [25], where the central server updates model parameters without access to the training data and intermediate values, and model parameters are shared among the central server with collaborating parties. However, it's inherently difficult to establish a third party trusted by any data owners in a real-world scenario. Moreover, data interactions between data owners and TTP raise the risk of leakage of sensitive data of the data owner.

To decrease the complexity of training a joint model for any two parties, by removing the TTP coordinator, Yang *et al.* [17] constructed a parallel distributed LR method for vertical federated learning based on HE [25], which allows two parties to jointly train models without the help of a TTP coordinator. Using the secure MPC protocol and ciphertext domain conversion protocol [37], Chen *et al.* [18] presented a collaborative learning system for jointly building better models over vertically partitioned multiple data. Based on the HE scheme [29], Li *et al.* [19] introduced a collaborative learning method on encrypted data, which could securely train LR models over vertically distributed data from both data owners. Based on asynchronous gradient sharing and HE algorithm [29], Wei *et al.* [20] designed a two-parties collaborative LR protocol, which can train securely joint model on the vertically partitioned data. Chen *et al.* [21] combined the HE [25] and secret sharing [38] to build securely LR model on the vertically distributed large-scale sparse training data. Over the horizontally partitioned data, based on secure MPC protocol, Ghavamipour *et al.* [22] described two methods to train LR model in a privacy-preserving manner. However, each data owner requires to compute multiple shares of its sensitive training data and sends them separately to each non-collusion computation party, this leads to heavy communication costs. He *et al.* [23]

constructed a vertical federated LR method through a HE algorithm [25], which uses a piecewise function to ensure the accuracy of the loss function, but this results in a loss of efficiency. With the HE scheme [25] and differential privacy algorithm [39], Sun *et al.* [24] introduced a vertical federated LR solution, which alleviates the constraints on feature dimensions. However, the existing PPCLR schemes [17–24] without a TTP coordinator lead to high communication and computational overhead.

3. Preliminaries

3.1. System Architecture. For ease of reading, the definitions of the symbols in our P^2 VCLR scheme are displayed in Table 1. As is shown in Figure 1, the system architecture of our P^2 VCLR includes two semi-trusted entities: P_a and P_b. P_a and P_b hold the vertically distributed datasets S_a and S_b, respectively. S_a and S_b have the same sample space but different feature distribution, namely, P_a holds the part of the features, P_b holds another part of the features and the label. P_a cooperates with P_b to train a shared LR model without disclosing the privacy of training data. Specifically, P_a generates $\{sk_a, pk_a, rk_a, gk_a\} \leftarrow \text{KeyGen}(N, Q)$ [29], sends polynomial-degree N, coefficient-modulus Q, scaling factor Δ, public key pk_a, relinearization key rk_a, galois key gk_a to P_b, and securely store secret key sk_a. Then, P_a encrypts its own data with pk_a, and sends the encrypted data to P_b. Finally, P_a and P_b jointly execute P^2 VCLR algorithm to obtain the training result.

3.2. Homomorphic Encryption. HE allows direct operations on ciphertext without decryption, and can ensure that the computation on the ciphertext is consistent with the computation on the plaintext. Cheon *et al.* [29] introduced an approximate HE algorithm from ring learning with errors (RLWE) [40], which supports the following operations.

(1) $\{sk_i,\ pk_i,\ rk_i,\ gk_i\} \leftarrow \text{Key\_Gen}(N, Q)$: Given the parameters $\{N, Q\}$, it generates sk_i, pk_i, rk_i, gk_i for P_i.

(2) $x \leftarrow \text{Enc}(x, pk_i)$: Given a message vector x and pk_i, it generates a ciphertext x.

(3) $x \leftarrow \text{Dec}(x, sk_i)$: Given x and sk_i, it generates a message vector x.

(4) $x + y \leftarrow \text{Add}(x, y)$: Given x and y, it generates a ciphertext $x + y = x + y$.

(5) $x + y \leftarrow \text{Add\_Plain}(x, y)$: Given x and a message vector y, it generates a ciphertext $x + y = x + y$.

(6) $x_0 + \cdots + x_{n-1} \leftarrow \text{Add\_Many}(x, y)$: Given a ciphertext list $X = \{x_0, \ldots, x_{n-1}\}$, it generates a ciphertext $x_0 + \cdots + x_{n-1} = x_0 + \cdots + x_{n-1}$.

(7) $x - y \leftarrow \text{Sub}(x, y)$: Given x and y, it generates a ciphertext $x - y = x - y$.

(8) $x - y \leftarrow \text{Sub\_Plain}(x, y)$: Given x and y, it generates a ciphertext $x - y = x - y$.

(9) $x * y \leftarrow \text{Mul}(x, y, rk_i)$: Given x, y and rk_i, it generates a ciphertext $x * y = x * y$.

(10) $x * y \leftarrow \text{Mul\_Plain}(x, y, rk_i)$: Given x, y and rk_i, it generates a ciphertext $x * y = x * y$.

(11) $y \leftarrow \text{Rot\_Vector}(x, k, gk_i)$: Given $x = [x_0, \ldots, x_{N/2-1}]$, k and gk_i, it rotates x left by rotation value k, and generates a ciphertext $y = [x_k, \ldots, x_{N/2-1}, x_0, \ldots, x_{k-1}]$.

3.3. Logistic Regression. Let a dataset S includes m samples $\{x_{i,1}, \ldots, x_{i,n}, y_i | i \in [m]\} = \{x_i, y_i | i \in [m]\}$, where an input x_i maps to a binary dependent variable $y_i \in \{0, 1\}$, the goal of binary LR is to compute weights $\xi = \{\xi_0, \xi_1, \ldots, \xi_n\}$ that minimizes the log-likelihood loss function $J(\xi) = -1/m \cdot \sum_{i=1}^{m} ((1 - y_i) \cdot (1 - \log(\sigma(z_i \cdot \xi))) + y_i \cdot \log(\sigma\ (z_i \cdot \xi)))$, where $z_i = \{1, x_i\}$. Assuming that $\xi^{(k)}$ and $\alpha^{(k)}$ denote the model weights and learning rate at iteration k, respectively, the gradient descent (GD) is able to be used to compute the extremum of $J(\xi)$ by $\xi^{(k+1)} \leftarrow \xi^{(k)} - \alpha^k/m \cdot \sum_{i=1}^{m} ((\sigma(z_i \cdot \xi^{(k)}) - y_i) \cdot z_i)$. Since the HE scheme (CKKS) [29] is not able to effectively support non-polynomial arithmetic operations, we use a 7-degree polynomial function $f(x) = w_0 + w_1 x + w_3 x^3 + w_5 x^5 + w_7 x^7$ to approximate sigmoid function $\sigma(x) = 1/(1 + e^{-x})$ over the domain [-8, 8], where $w_0 = 1/2$, $w_1 = 1.73496/8$, $w_3 = 4.19407/8^3$, $w_5 = 5.43402/8^5$, and $w_7 = 2.50739/8^7$.

4. Privacy-Preserving Vertical Collaborative Logistic Regression

Over vertically distributed datasets S_a and S_b, we propose a P^2 VCLR scheme based on an approximate HE [29]. Using batching method in approximate HE, the proposed scheme packs a message vector with multiple messages into a plaintext with multiple plaintext slots, and performs parallel training based on SIMD. For ease of readability, we give the Algorithm , which can be found in Appendix. We assume that the samples of S_a and S_b held by P_a and P_b have been aligned, namely,

Table 1: The definition of the symbol.

Notation	Definition
x	Message vector $[x_0, \ldots, x_{N/2-1}]$
$x_{[i]}$	The i-th element of x
x	A ciphertext of x
X	A list of message vectors $\{x_0, \ldots, x_{n-1}\}$
$X_{[i]}$	The i-th message vector of X
X	A list of ciphertexts $\{x_0, x_1, \ldots, x_{n-1}\}$
$X_{[i]}$	The i-th ciphertext of X
$x * y$	$[x_0 \cdot y_0, \ldots, x_{N/2-1} \cdot y_{N/2-1}]$
$x + y$	$[x_0 + y_0, \ldots, x_{N/2-1} + y_{N/2-1}]$
$x - y$	$[x_0 - y_0, \ldots, x_{N/2-1} - y_{N/2-1}]$
$x \cdot y$	$[x_0 \cdot y_0 + \cdots + x_{N/2-1} \cdot y_{N/2-1}]$

$$S_a = \begin{bmatrix} x_{0,1} \\ x_{1,1} \\ \vdots \\ x_{m-1,1} \\ x_{0,2} \\ x_{1,2} \\ \vdots \\ x_{m-1,2} \\ \cdots \\ \cdots \\ \ddots \\ \cdots \\ x_{0,n_1} \\ x_{1,n_1} \\ \vdots \\ x_{m-1,n_1} \end{bmatrix} \begin{bmatrix} x_{0,n_1+1} \\ x_{1,n_1+1} \\ \vdots \\ x_{m-1,n_1+1} \\ \cdots \\ \cdots \\ \ddots \\ \cdots \\ x_{0,n_1+n_2} \\ x_{1,n_1+n_2} \\ \vdots \\ x_{m-1,n_1+n_2} \\ y_0 \\ y_1 \\ \vdots \\ y_{m-1} \end{bmatrix} = S_b. \tag{1}$$

S_a and S_b consist of m samples of the form $\{x_{i,1}, x_{i,2}, \ldots, x_{i,n_1}\}$ and $\{x_{i,n_1+1}, \ldots, x_{i,n_1+n_2}, y_i\}$, respectively, where $i = 0, 1, \ldots, m-1$. Each column of S_a denote the features. The last column of S_b represents the label, and other columns of S_b represent the features. P_a cooperates with P_b to train a shared LR model without revealing the data privacy. Suppose $2(n_1 + n_2 + 1) \leq N$, the details of the proposed P^2 VCLR are described below.

Input: S_a and S_b for P_a and P_b respectively

Output: $[\xi_0^{(s)}, \xi_1^{(s)}, \ldots, \xi_{n_1}^{(s)}]$ and $[\xi_{n_1+1}^{(s)}, \xi_{n_1+2}^{(s)}, \ldots, \xi_{n_1+n_2}^{(s)}]$ for P_a and P_b respectively

Preprocessing:

1: P_a computes $l = 2^{\log(n_1+n_2+1)}$, $u = N/2l$, $v = m/u$, lets $\{x_i' = [1, x_{i,1}, x_{i,2}, \ldots, x_{i,n_1}, 0, 0, \ldots, 0] \mid i = 0, 1, \ldots, m-1\}$, generates $\{sk_a, pk_a, rk_a, gk_a\} \leftarrow \mathrm{KeyGen}(N, Q)$, encrypts dataset S_a into v ciphertexts

$$\left\{ x_{a,i} = \mathrm{Enc}(\underbrace{[x_{i\cdot u}', x_{i\cdot u+1}', \ldots, x_{(i+1)\cdot u-1}']}_{N/2}, pk_a) \mid \quad i = 0, 1, \ldots, v-2 \right\},$$

$x_{a,v-1} = \mathrm{Enc}(\underbrace{[x_{(v-1)\cdot u}', x_{(v-1)\cdot u+1}', \ldots, x_m', 0, 0, \ldots, 0]}_{N/2}, pk_a)$,

lets

$\xi'^{(0)} = [\xi_0^{(0)}, \xi_1^{(0)}, \ldots, \xi_{n_1}^{(0)}, 0, 0, \ldots, 0]$,

encrypts the initial weight $\xi'^{(0)}$ into one ciphertext

$$\xi_a^{(0)} = \mathrm{Enc}(\underbrace{[\xi'^{(0)}, \xi'^{(0)}, \ldots, \xi'^{(0)}]}_{N/2}, pk_a),$$

and sends N, Q, Δ, pk_a, rk_a, gk_a, s, $\{x_{a,i} \mid i = 0, 1, \ldots, v-1\}$, $\xi_a^{(0)}$ to P_b.

2: P_b computes $l = 2^{\log(n_1+n_2+1)}$, $u = N/2l$, $v = m/u$, lets

$$\left\{ x_i'' = \underbrace{[\underbrace{0, 0, \ldots, 0}_{n_1+1}, x_{i,n_1+1}, x_{i,n_1+2}, \ldots, x_{i,n_1+n_2}, 0, 0, \ldots, 0]}_{l} \quad i = 0, 1, \ldots, m-1 \right\},$$

$$\left\{ y_i = \underbrace{[y_i, 0, 0, \ldots, 0]}_{l} \quad i = 0, 1, \ldots, m-1 \right\},$$

sets data set S_b into $2v$ message vectors

$\{x_{b,i} = \underbrace{[x_{i\cdot u}'', x_{i\cdot u+1}'', \cdots, x_{(i+1)\cdot u-1}'']}_{N/2}$ $i = 0, 1, \ldots, v-2\}$,

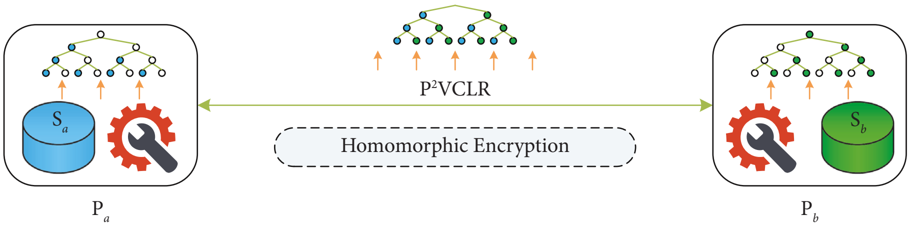

Figure 1: System architecture.

```
$x_{b,v-1} = \underbrace{[x''_{(v-1)\cdot u}, x''_{(v-1)\cdot u+1}, \ldots, x''_m, 0, 0, \ldots, 0]}_{N/2}$ ,
$\{y_i = \underbrace{[y_{i\cdot u}, y_{i\cdot u+1}, \ldots, y_{(i+1)\cdot u-1}]}_{N/2} \quad i = 0, 1, \ldots, v-2\}$,
$y_{v-1} = \underbrace{[y_{(v-1)\cdot u}, y_{(v-1)\cdot u+1}, \ldots, y_m, 0, 0, \ldots, 0]}_{N/2}$,
lets
$\xi''^{(0)} = \underbrace{[\overbrace{0, 0, \ldots, 0}^{n_1+1}, \xi^{(0)}_{n_1+1}, \xi^{(0)}_{n_1+2}, \ldots, \xi^{(0)}_{n_1+n_2}, 0, 0, \ldots, 0]}_{l}$,
sets the initial weight $\xi''^{(0)}$ into one message vector
$\xi^{(0)}_b = [\underbrace{\xi''^{(0)}, \xi''^{(0)}, \ldots, \xi''^{(0)}}_{N/2}]$,
sets the learning rate $\alpha$ into one message vector
$\alpha/m = \underbrace{[\overbrace{\alpha/m, \alpha/m, \ldots, \alpha/m}^{n_1+n_2}, 0, 0, \ldots, 0]}_{N/2}$,
lets
$\left\{ w_i = \underbrace{[w_i, 0, 0, \ldots, 0]}_{l} \quad i = 0, 1, 3, 5, 7 \right\}$,
sets the message vectors
$\left\{ \omega_i = \underbrace{[w_i, w_i, \ldots, w_i]}_{N/2} \quad i = 0, 1, 3, 5, 7 \right\}$,
sets the lists
$X_a = \{x_{a,0}, x_{a,1}, \ldots, x_{a,v-1}\}$,
$X_b = \{x_{b,0}, x_{b,1}, \ldots, x_{b,v-1}\}$,
$Y = \{y_0, y_1, \ldots, y_{v-1}\}$.
**Training**:
$P_b$ computes $\xi^{(0)} = \text{Add\_Plain}(\xi^{(0)}_a, \xi^{(0)}_b)$
**for** $(i = 0$ to $v-1)$ **do**
    $P_b$ computes $X_{[i]} = \text{Add\_Plain}(X_{a,[i]}, X_{b,[i]})$
**end for**
**for** $(j = 0$ to $s-1)$ **do**
    **for** $(i = 0$ to $v-1)$ **do**
        $P_b$ computes $D_{[i]} = \text{Mul}(\xi^{(j)}, \text{X}_{[i]}, rk_a)$
        $P_b$ computes $E_{[i]} = \text{Rot\_Sum\_1}(D_{[i]}, l, gk_a)$
        $P_b$ computes $F_{[i]} = \text{Approx\_Sigmoid}(E_{[i]}, \omega_0, \omega_1, \omega_3, \omega_5, \omega_7, rk_a)$
        $P_b$ computes $\text{G}_{[i]} = \text{Sub\_Plain}(F_{[i]}, Y_{[i]})$
        $P_b$ computes $H_{[i]} = \text{Rot\_Sum\_2}(G_{[i]}, l, gk_a)$
        $P_b$ computes $I_{[i]} = \text{Mul}(H_{[i]}, X_{[i]}, rk_a)$
    **end for**
    $P_b$ computes $a = \text{Add\_Many}(I)$
    $P_b$ computes $b = \text{Rot\_Sum\_3}(a, l, gk_a)$
    $P_b$ computes $c = \text{Mul\_Plain}(b, \alpha/m, rk_a)$
    $P_b$ computes $\widehat{\xi}^{(j+1)} = \text{Sub}(\xi^{(j)}, c)$
    $P_b$ chooses random message vector $\delta^{(j+1)} = [\delta^{(j+1)}_0, \delta^{(j+1)}_1, \ldots, \delta^{(j+1)}_{N/2-1}]$
    $P_b$ computes $\varphi^{(j+1)} = \text{Sub\_Plain}(\widehat{\xi}^{(j+1)}, \delta^{(j+1)})$
    $P_b$ sends $\varphi^{(j+1)}$ to $P_a$
    $P_a$ computes $\varphi^{(j+1)} = \text{Dec}(\varphi^{(j+1)}, sk_a)$ to $P_a$
    $P_a$ sets $\widehat{\varphi}^{(j+1)} = [\varphi^{(j+1)}_{[0]}, \varphi^{(j+1)}_{[1]}, \ldots, \varphi^{(j+1)}_{[n_1+n_2]}, 0, 0, \ldots, 0]$
    $P_a$ sets $\widetilde{\varphi}^{(j+1)} = \underbrace{[\widehat{\varphi}^{(j+1)}, \widehat{\varphi}^{(j+1)}, \ldots, \widehat{\varphi}^{(j+1)}]}_{N/2l}$
    $P_a$ sets $\widetilde{\varphi}^{(j+1)} = \text{Enc}(\widetilde{\varphi}^{(j+1)}, pk_a)$
    $P_a$ sends $\widetilde{\varphi}^{(j+1)}$ to $P_b$
    $P_b$ sets $\widehat{\delta}^{(j+1)} = \underbrace{[\delta^{(j+1)}_{[0]}, \delta^{(j+1)}_{[1]}, \ldots, \delta^{(j+1)}_{[n_1+n_2]}, 0, 0, \ldots, 0]}_{l}$
    $P_b$ sets $\widetilde{\delta}^{(j+1)} = [\underbrace{\widehat{\delta}^{(j+1)}, \widehat{\delta}^{(j+1)}, \ldots, \widehat{\delta}^{(j+1)}}_{N/2}]$
    $P_b$ computes $\xi^{(j+1)} = \text{Add\_Plain}(\widetilde{\varphi}^{(j+1)}, \widetilde{\delta}^{(j+1)})$
**end for**
**Reconstructing**:
$P_a$ sends $[\varphi^{(s)}_{[n_1+1]}, \varphi^{(s)}_{[n_1+2]}, \ldots, \varphi^{(s)}_{[n_1+n_2]}]$ to $P_b$
$P_b$ computes $[\xi^{(s)}_{n_1+1}, \xi^{(s)}_{n_1+2}, \ldots, \xi^{(s)}_{n_1+n_2}] = [\varphi^{(s)}_{[n_1+1]}, \varphi^{(s)}_{[n_1+2]}, \ldots, \varphi^{(s)}_{[n_1+n_2]}] + [\delta^{(s)}_{[n_1+1]}, \delta^{(s)}_{[n_1+2]}, \ldots, \delta^{(s)}_{[n_1+n_2]}]$
$P_b$ sends $[\delta^{(s)}_{[0]}, \delta^{(s)}_{[1]}, \ldots, \delta^{(s)}_{[n_1]}]$ to $P_a$
$P_a$ computes $[\xi^{(s)}_0, \xi^{(s)}_1, \ldots, \xi^{(s)}_{n_1}] = [\varphi^{(s)}_{[0]}, \varphi^{(s)}_{[1]}, \ldots, \varphi^{(s)}_{[n_1]}] + [\delta^{(s)}_{[0]}, \delta^{(s)}_{[1]}, \ldots, \delta^{(s)}_{[n_1]}]$
**return**: $[\xi^{(s)}_0, \xi^{(s)}_1, \ldots, \xi^{(s)}_{n_1}]$ and $[\xi^{(s)}_{n_1+1}, \xi^{(s)}_{n_1+2}, \ldots, \xi^{(s)}_{n_1+n_2}]$ for $P_a$ and $P_b$ respectively
```

5. Performance Evaluation

We execute the performance comparisons among our P^2 VCLR scheme and existing schemes [9, 21]. We perform all experiments on a 64-bits Linux system machine with i7 CPU and 16 GB memory. For all experiments, we choose the initial weights $[\xi^{(0)}_0, \xi^{(0)}_1, \ldots, \xi^{(0)}_{n_1}] = [0, 0, \ldots, 0]$, $[\xi^{(s)}_{n_1+1}, \xi^{(s)}_{n_1+2}, \ldots, \xi^{(s)}_{n_1+n_2}] = [0, 0, \ldots, 0]$, the learning rate $\alpha = 0.15$, and the maximum number of iterations $s = 20$. The schemes [9, 21] choose the Paillier cryptosystem [25] to provide the additive HE operations, the proposed scheme

Table 2: Performance comparisons.

S_a	S_b	scheme	Training time	Accuracy	F1-score	AUC	No TTP
Θ_1: $\{x_1 - x_4\}$	Θ_1: $\{x_5 - x_8, y\}$	[9]	1.27 min	74.1 %	85.1 %	0.57	×
		[21]	2.18 min	74.1 %	85.1 %	0.56	√
		Ours	0.86 min	74.4 %	85.2 %	0.58	√
Θ_2: $\{x_1 - x_5\}$	Θ_1: $\{x_6 - x_9, y\}$	[9]	2.20 min	91.3 %	77.5 %	0.96	×
		[21]	3.41 min	90.9 %	75.3 %	0.96	√
		Ours	1.49 min	92.3 %	78.0 %	0.96	√
Θ_3: $\{x_1 - x_8\}$	Θ_1: $\{x_9 - x_{15}, y\}$	[9]	11.13 min	82.7 %	60.1 %	0.88	×
		[21]	21.32 min	82.7 %	60.1 %	0.89	√
		Ours	5.87 min	85.7 %	61.9 %	0.91	√

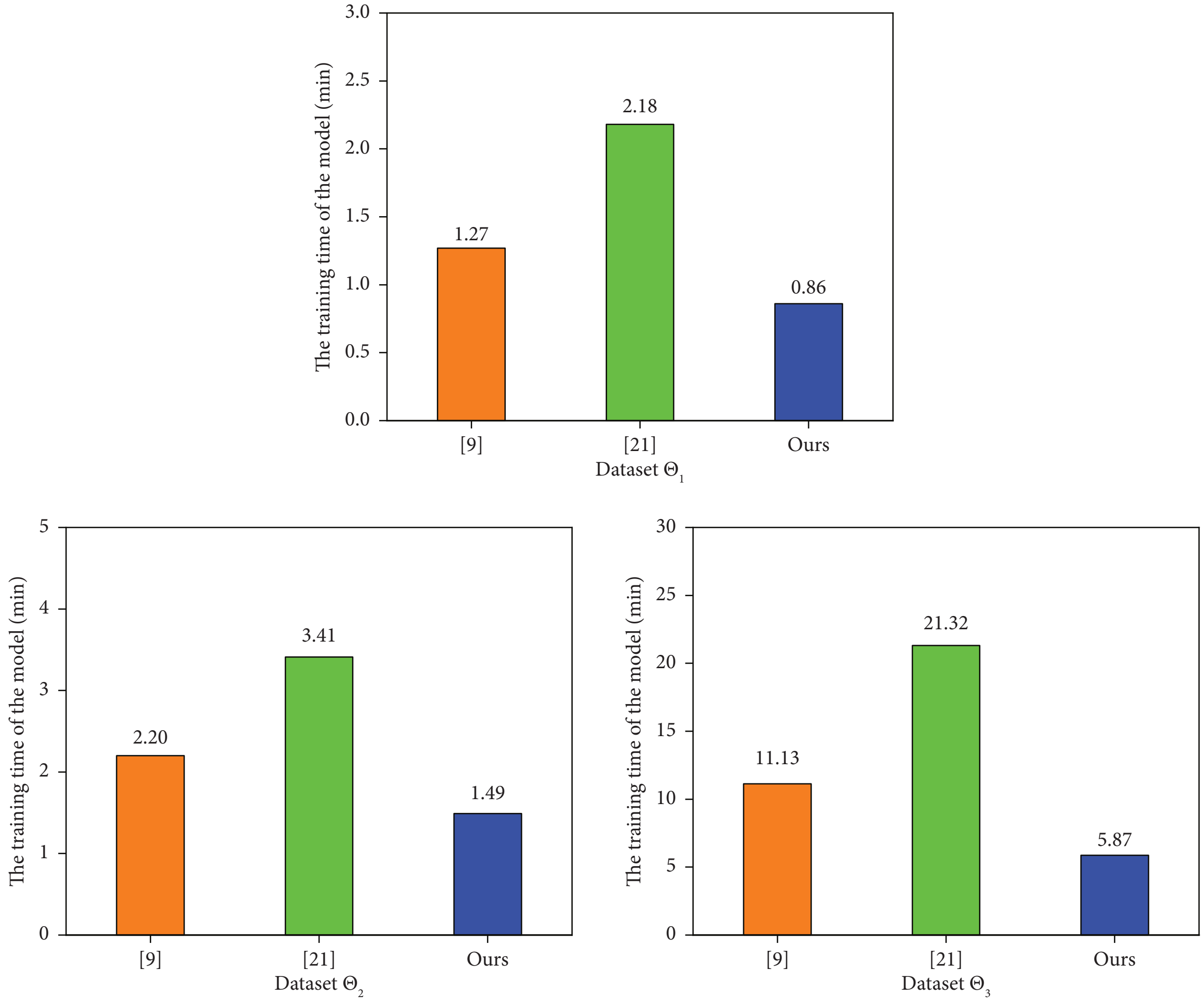

Figure 2: The training time of the model.

uses the Microsoft SEAL library [41] to instantiate the HE operations [29]. To achieve $\kappa = 80$ bits security, for the schemes [9, 21], we set the prime number $p, q = 512$ bits and $n = 1024$ bits; for the proposed scheme, we choose the polynomial-degree $N = 2^{15}$, the coefficient-modulus $Q = 520$, and the scaling factor $\Delta = 2^{40}$. On three publicly available datasets [30]: Θ_1 - Umaru Impact Study, Θ_2 - Myocardial Infarction from Edinburgh, and Θ_3 - Nhanes III, we compare the proposed scheme and schemes [9, 21] in terms of training time, accuracy, F1-score, AUC. P_a has the first 4 features $\{x_1 - x_4\}$ of all samples of Θ_1, P_b has the last 4 features and labels $\{x_5 - x_8, y\}$ of all samples of Θ_1; P_a has the first 5 features $\{x_1 - x_5\}$ of all samples of Θ_2, P_b has the last 4 features and labels $\{x_6 - x_9, y\}$ of all samples of Θ_2; P_a has the first 8 features $\{x_1 - x_8\}$ of all samples of Θ_3, P_b has the last 7 features and labels $\{x_9 - x_{15}, y\}$ of all samples of

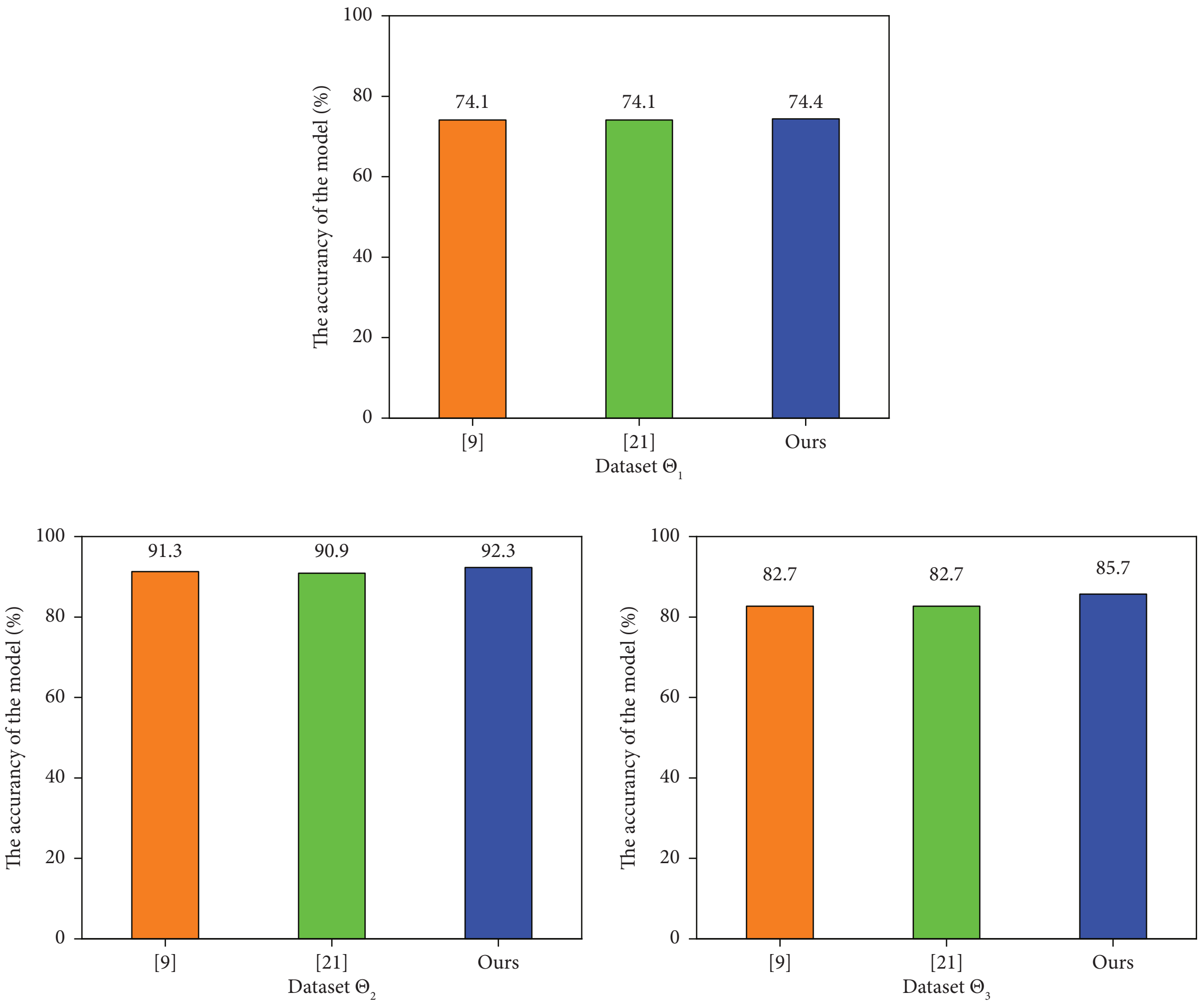

Figure 3: The accuracy of the model.

Θ_3. We get the validity of the experimental results by using 5-fold cross-validation. All experiment results are shown as the average of 10 experiments. The performance comparisons between the proposed scheme and schemes [9,21} are described in Table 2, in which ” √” denotes ”satisfied” and ” ×” means ”unsatisfied”. From Table 2, we can see that our P^2 VCLR scheme outperforms existing schemes [9, 21] in both training time and model performance, and does not need a TTP coordinator.

From Figure 2, we can get that, for dataset Θ_1, in our scheme, the training time of the model is 0.86 min, which is decreased by nearly 32.3% and 60.6% compared with that of [9, 21], respectively; for dataset Θ_2, in our scheme, the training time of the model is 1.49 min, which is reduced by almost 32.3% and 56.3% in comparison to that of [9, 21], respectively; for dataset Θ_3, in our scheme, the training time of the model is 5.87 min, which is nearly 47.3% and 72.5% less than that of [9, 21], respectively.

From Figure 3, we can get that, for dataset Θ_1, in our scheme, the accuracy of the model is 74.4%, which has nearly 0.3% and 0.3% improvement compared with that of [9, 21], respectively; for dataset Θ_2, in our scheme, the accuracy of the model is 92.3%, which has an increase of almost 1.0% and 1.4% in comparison to that of [9, 21], respectively; for dataset Θ_3, in our scheme, the accuracy of the model is 85.7%, which is nearly 3.0% and 3.0% higher than that of [9, 21], respectively.

From Figure 4, we can get that, for dataset Θ_1, in our scheme, the F1-score of the model is 85.2%, which has nearly 0.1% and 0.1% improvement compared with that of [9, 21], respectively; for dataset Θ_2, in our scheme, the F1-score of the model is 78.0%, which has an increase of almost 0.5% and 2.7% in comparison to that of [9, 21], respectively; for dataset Θ_3, in our scheme, the F1-score of the model is 61.9%, which is nearly 1.8% and 1.8% higher than that of [9, 21], respectively.

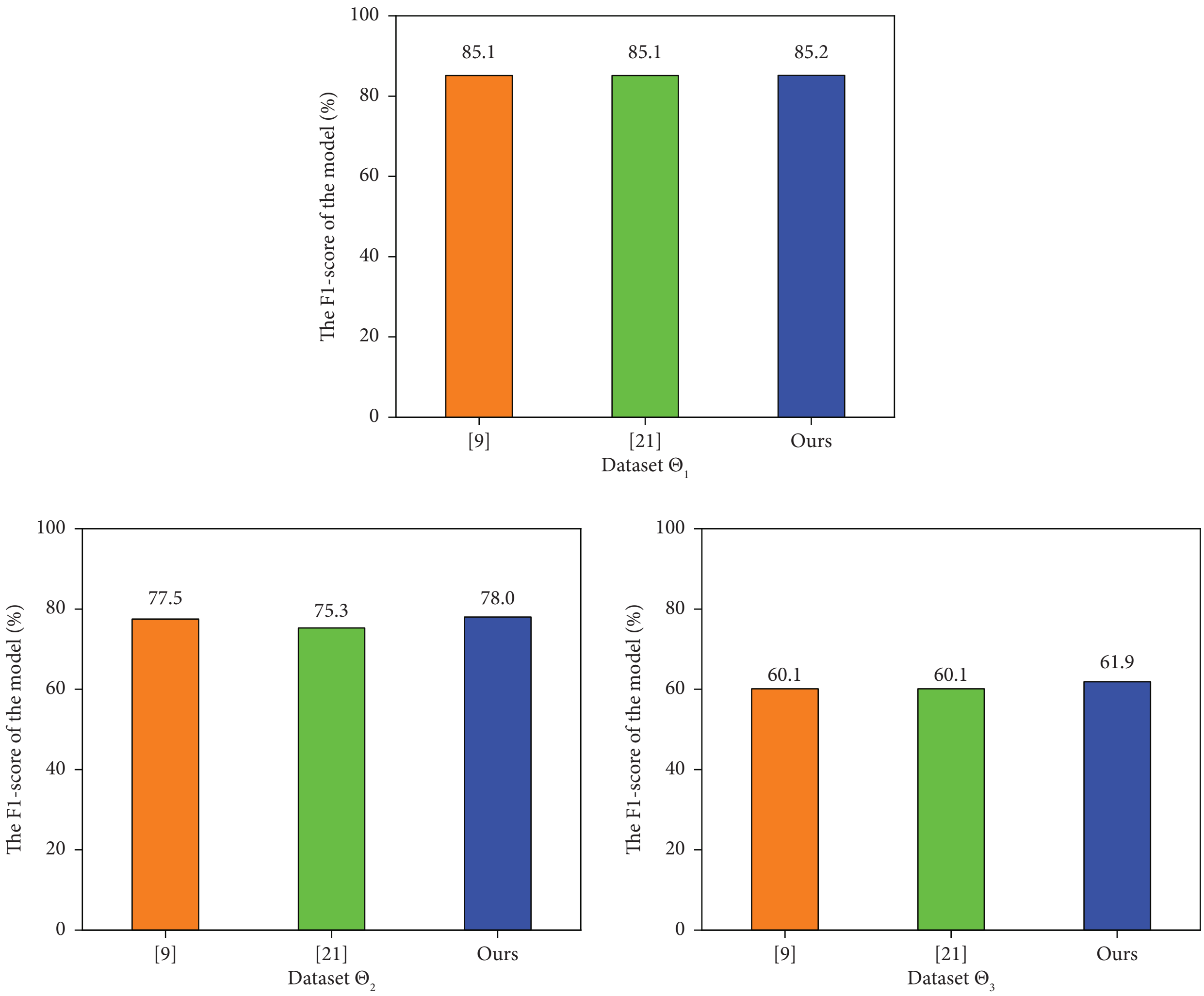

Figure 4: The F1-score of the model.

From Figure 5, we can get that, for dataset Θ_1, in our scheme, the AUC of the model is 0.58, which has nearly 0.01 and 0.02 improvement compared with that of [9, 21], respectively; for dataset Θ_2, in our scheme, the AUC of the model is 0.96, which is the same as that of [9, 21]; for dataset Θ_3, in our scheme, the AUC of the model is 0.91, which is nearly 0.03 and 0.02 higher than that of [9, 21], respectively.

6. Security Analysis

In the semi-honest model [42], we let the parties P_a and P_b know pk_a, rk_a, gk_a, and only P_a has sk_a. The proposed P^2 VCLR scheme belongs to secure two-party computation, which denotes an objective functionality $\mathcal{F} \cdot \{\mathcal{F}_a, \mathcal{F}_b\}$. For the inputs $\{x_a, x_b\}$, where x_a is from party P_a and x_b is from party P_b, the outputs $\{\mathcal{F}_a(x_a, x_b), \mathcal{F}_b(x_a, x_b)\}$ are random. $\mathcal{F}_a(x_a, x_b)$ is the output for P_a, and $\mathcal{F}_b(x_a, x_b)$ is for P_b, and neither party can know more private information than its output. According to the simulation-based security [43], we perform a security analysis of our P^2 VCLR scheme.

Definition 1. Let $\mathcal{F}$ be a deterministic functionality and Π be a secure two-party computation protocol to compute $\mathcal{F}$. Given P_a's input x_a, P_b's input x_b, and security level κ, the views of P_a and P_b in the protocol Π are denoted as $\mathcal{V}_a = \{\kappa, x_a, x_b\} = \{sk_a, pk_a, rk_a, gk_a, x_a, x_b, y_a\}$ and $\mathcal{V}_b = \{\kappa, x_a, x_b\} = \{pk_a, rk_a, gk_a, x_b, y_b\}$, where y_a and y_b are the messages received by P_a and P_b. We think that, in semi-honest model, Π can securely calculate $\mathcal{F}$ if there are the probabilistic polynomial-time (PPT) simulators $\mathcal{S}_a$ and $\mathcal{S}_b$, such that

$$\begin{aligned}\{\mathcal{S}_a(1^\kappa, x_a, \mathcal{F}_a(x_a, x_b))\}_{\kappa, x_a, x_b} &\cong \{\mathcal{V}_a(\kappa, x_a, x_b)\}_{\kappa, x_a, x_b}\\ \{\mathcal{S}_b(1^\kappa, x_b, \mathcal{F}_b(x_a, x_b))\}_{\kappa, x_a, x_b} &\cong \{\mathcal{V}_b(\kappa, x_a, x_b)\}_{\kappa, x_a, x_b}.\end{aligned} \tag{2}$$

Theorem 1. *Assuming that the P_a and P_b do not collude with each other, and the HE scheme (CKKS) [29] satisfies the semantic security, our P^2 VCLR scheme can ensure the security in semi-honest model.*

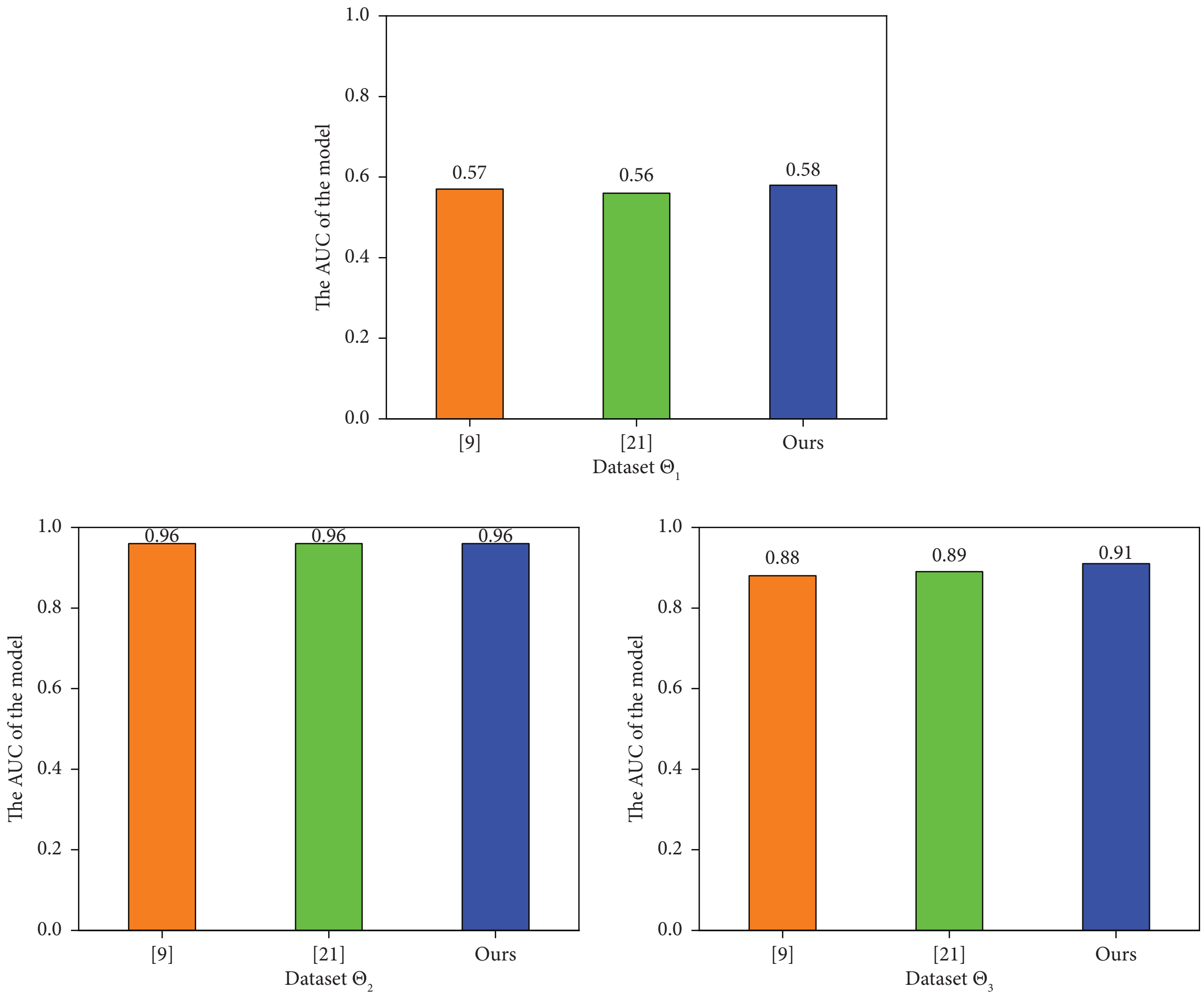

FIGURE 5: The AUC of the model.

Security Proof. Security proof of our P^2 VCLR scheme follows the simulation-based security [43]. We prove that we are able to build $\mathscr{S}_a$ and $\mathscr{S}_b$, such that

$$\begin{aligned}\left\{\mathscr{S}_{\mathscr{A}_a}(1^\kappa, y_a, sk_a)\right\}_{\kappa,z,sk_a} &\cong \left\{\mathscr{V}_{\mathscr{A}_a}(\kappa, z, sk_a)\right\}_{\kappa,z,sk_a} \\ \left\{\mathscr{S}_{\mathscr{A}_b}(1^\kappa, z, y_b)\right\}_{\kappa,z,sk_a} &\cong \left\{\mathscr{V}_{\mathscr{A}_b}(\kappa, z, sk_a)\right\}_{\kappa,z,sk_a},\end{aligned} \tag{3}$$

where $\mathscr{V}_{\mathscr{A}_a}$ and $\mathscr{V}_{\mathscr{A}_b}$ denote the views of $\mathscr{A}_a$ and $\mathscr{A}_b$, respectively. Next, we show that the above two equations are indistinguishable for the corrupted parties $\mathscr{A}_a$ and $\mathscr{A}_b$, respectively.

Against corrupted $\mathscr{A}_a$: We build $\mathscr{S}_a$ that, when given κ, $\mathscr{A}_a$'s input sk_a and $\mathscr{A}_a$'s output y_a, is able to simulate $\mathscr{A}_a$'s view in the execution of the protocol. In this respect, we then analyze $\mathscr{A}_a$'s view $\mathscr{V}_{\mathscr{A}_a}(\kappa, z, sk_a)$ in the execution of the protocol. The only message $\mathscr{A}_a$ gets is the ciphertext z. Therefore, $\mathscr{V}_{\mathscr{A}_a}(\kappa, z, sk_a)$ consists of $\mathscr{A}_a$'s secret key sk_a, random message vector r_a and ciphertext y_a. Given κ, sk_a, and y_a, $\mathscr{S}_{\mathscr{A}_a}$ generates a simulation of $\mathscr{V}_{\mathscr{A}_a}(\kappa, z, sk_a)$. $\mathscr{S}_{\mathscr{A}_a}$ encrypts $\mathscr{S}_{\mathscr{A}_a}$ with pk_a into y'_a, and generates the output (sk_a, r_a, y'_a). Therefore, we can obtain two equations as follows:

$$\begin{aligned}\mathscr{V}_{\mathscr{A}_a}(\kappa, z, sk_a) &= (sk_a, r_a, y'_a) \\ \mathscr{S}_{\mathscr{A}_a}(1^\kappa, y_a, sk_a) &= (sk_a, r_a, y'_a).\end{aligned} \tag{4}$$

Through the above analysis, we are able to get that probability distribution of $\mathscr{A}_a$'s view and $\mathscr{S}_{\mathscr{A}_a}$'s output is indistinguishable. Therefore, the proposed P^2 VCLR scheme is secure against the corrupted $\mathscr{A}_a$ in semi-honest model.

Against corrupted $\mathscr{A}_b$: We build $\mathscr{S}_b$ that, when given κ, $\mathscr{A}_b$'s input z and $\mathscr{A}_b$'s output y_b, is able to simulate $\mathscr{A}_b$'s view in the execution of the protocol. For this reason, we analyze $\mathscr{A}_b$'s view $\mathscr{V}_{\mathscr{A}_b}(\kappa, z, sk_a)$ in the execution of the protocol. $\mathscr{A}_b$ does not receive any message vectors from $\mathscr{A}_a$. Therefore, $\mathscr{V}_{\mathscr{A}_b}(\kappa, z, sk_a)$ consists of $\mathscr{A}_b$'s input z and random message vector r_b. Given κ, z, and y_b, $\mathscr{S}_{\mathscr{A}_b}$ generates a simulation of $\mathscr{V}_{\mathscr{A}_b}(\kappa, z, sk_a)$ by outputting (z, r_b). Therefore, we have the following two equations:

$$\begin{aligned}\mathscr{V}_{\mathscr{A}_b}(\kappa, z, sk_a) &= (z, r_b) \\ \mathscr{S}_{\mathscr{A}_b}(1^\kappa, z, y_b) &= (z, r_b).\end{aligned} \tag{5}$$

Through the above analysis, we are able to get that probability distributions of $\mathscr{A}_b$'s view and $\mathcal{S}_{\mathscr{A}_b}$'s output are indistinguishable. Therefore, the proposed P^2 VCLR scheme is secure against the corrupted $\mathscr{A}_b$ in the semi-honest model.

7. Conclusion

In this paper, to improve the efficiency of the collaborative LR, based on an approximate HE algorithm, we propose a P^2 VCLR over vertically distributed data while realizing the security of training data and the privacy of model parameters for all parties. We then evaluate the proposed scheme on the public datasets. The evaluation results show that our P^2 VCLR scheme achieves a better performance in terms of joint training time and model performance in comparison to that of existing schemes [9, 21]. Specifically, the training time of the model is decreased by almost 32.3%-72.5%; the accuracy, F1-score, and AUC of the model have nearly 0.3% - 3.0%, 0.1% - 2.7% and 0 - 0.03 improvement, respectively. In the future, we will extend our method for supporting more complex ML, and deploy our scheme for practical applications.

Appendix

```
Input: $x, \omega_0, \omega_1, \omega_3, \omega_5, \omega_7, rk_i$
Output: $f(x)$
$x^2 = \mathrm{Mul}(x, x, rk_i)$
$x^4 = \mathrm{Mul}(x^2, x^2, rk_i)$
$x^6 = \mathrm{Mul}(x^2, x^4, rk_i)$
$\omega_7 x = \mathrm{Mul\_Plain}(x, \omega_7, rk_i)$
$\omega_7 x^7 = \mathrm{Mul}(\omega_7 x, x^6, rk_i)$
$\omega_5 x = \mathrm{Mul\_Plain}(x, \omega_5, rk_i)$
$\omega_5 x^5 = \mathrm{Mul}(\omega_5 x, x^4, rk_i)$
$\omega_3 x = \mathrm{Mul\_Plain}(x, \omega_3, rk_i)$
$\omega_3 x^3 = \mathrm{Mul}(\omega_3 x, x^2, rk_i)$
$\omega_1 x = \mathrm{Mul\_Plain}(x, \omega_1, rk_i)$
$\omega_0 + \omega_1 x = \mathrm{Add\_Plain}(\omega_1 x, \omega_0)$
$\omega_0 + \omega_1 x - \omega_3 x^3 = \mathrm{Sub}(\omega_0 + \omega_1 x, \omega_3 x^3)$
$\omega_0 + \omega_1 x - \omega_3 x^3 + \omega_5 x^5 = \mathrm{Add}(\omega_0 + \omega_1 x - \omega_3 x^3, \omega_5 x^5)$
$\omega_0 + \omega_1 x - \omega_3 x^3 + \omega_5 x^5 - \omega_7 x^7 = \mathrm{Sub}(\omega_0 + \omega_1 x - \omega_3 x^3 + \omega_5 x^5, \omega_7 x^7)$
return: $f(x) = \omega_0 + \omega_1 x - \omega_3 x^3 + \omega_5 x^5 - \omega_7 x^7$
```

```
Input: $x = [[[x_0, x_1, \ldots, x_{l-1}, x_l, x_{l+1}, \ldots, x_{2l-1}, \ldots, x_{(u-1)l}, x_{(u-1)l+1}, \ldots, x_{N/2-1}]]], l, gk_i$
Output: $y = [[[\sum_{i=} 0^{l-1} x_i, \circ, \ldots, \circ\_l, \sum_{i=l}^{2l-1} x_i, \circ, \ldots, \circ, \ldots, \sum_{i=(u-1)l}^{N/2-1} x_i, \circ, \ldots, \circ]]]$
$y = x_{\bar{l}}$        $\bar{l}$
for $(k = l/2;\ k \geq 1;\ k = k/2)$ do
    $z = \mathrm{Rot\_Vector}(y, k, gk_i)$
    $y = \mathrm{Add}(y, z)$
end for
return: $y$
```

```
Input:
$x = [[[x_0, 0, \ldots, 0, x_l, 0, \ldots, 0, \ldots, x_{(u-1)l}, 0, \ldots, 0]]], l, gk_i$
Output: $\bar{l}$        $\bar{l}$        $\bar{l}$
$y = [[[x_0, x_0, \ldots, x_0, x_l, x_l, \ldots, x_l, \ldots, x_{(u-1)l}, x_{(u-1)l}, \ldots, x_{(u-1)l\_}l]]]_{\bar{l}}$        $\bar{l}$
$y = x$
for $(k = l/2;\ k \geq 1;\ k = k/2)$ do
    $z = \mathrm{Rot\_Vector}(y, -k, gk_i)$
    $y = \mathrm{Add}(y, z)$
end for
return: $y$
```

```
Input:
$x = [[[x_0, x_1, \ldots, x_{l-1}, x_l, x_{l+1}, \ldots, x_{2l-1}, \ldots, x_{(u-1)l}, x_{(u-1)l+1}, \ldots, x_{N/2-1}]]], l, gk_i$
Output:
$y = [[[\sum_{i=0}^{u-1} x_{il}, \ldots, \sum_{i=0}^{u-1} x_{(i+1)l-1}, \ldots, \sum_{i=0}^{u-1} x_{il}, \ldots, \sum_{i=0}^{u-1} x_{(i+1)l-1\_}l]]]$        $\bar{l}$
$y = x$
for $(k = N/2;\ k \geq l/2 + 1;\ k = k/2)$ do
    $z = \mathrm{Rot\_Vector}(y, k, gk_i)$
    $y = \mathrm{Add}(y, z)$
end for
return: $y$
```

Data Availability

Previously reported datasets were used to support this study and are available at https://doi.org/10.1186/s12920-018-0401-7. These prior studies (and datasets) are cited at relevant places within the text as references [30].

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work is supported by the National Key R\&D Program of China (Grant No. 2019YFE0113200) the National Natural Science Foundation of China (Grant No. U19B2021, No. 61901317), and the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2020A1515110496, No. 2020A1515110079).

References

[1] P. Mohassel and Y. Zhang, “SecureML: a system for scalable privacy-preserving machine learning,” in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 19–38, San Jose, CA, USA, May, 2017.

[2] J. M. Cort, A. Tchernykh, M. Babenko, B. Pulido-Gayt, and G. Radchenko, “Multi-cloud privacy-preserving logistic

regression," in *Proceedings of the 7th Russian Supercomputing Days*, pp. 457–471, Moscow, Russia, September, 2021.

[3] Y. Guan, "Application of logistic regression algorithm in the diagnosis of expression disorder in Parkinson's disease," in *Proceedings of the 2nd International Conference on Information Technology, Big Data and Artificial Intelligence*, pp. 1117–1120, Chongqing, China, December, 2021.

[4] E. Dumitrescu, S. Hué, C. Hurlin, and S. Tokpavi, "Machine learning for credit scoring: improving logistic regression with non-linear decision-tree effects," *European Journal of Operational Research*, vol. 297, no. 3, pp. 1178–1192, 2022.

[5] J. Feng, W. Zhang, Q. Pei, J. Wu, and X. Lin, "Heterogeneous computation and resource allocation for wireless powered federated edge learning systems," *IEEE Transactions on Communications*, vol. 70, no. 5, pp. 3220–3233, 2022.

[6] J. Feng, L. Liu, Q. Pei, and K. Li, "Min-max cost optimization for efficient hierarchical federated learning in wireless edge networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 11, pp. 1–2700, 2022.

[7] J. Du, F. R. Yu, X. Chu, J. Feng, and G. Lu, "Computation offloading and resource allocation in vehicular networks based on dual-side cost minimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1079–1092, 2019.

[8] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019.

[9] S. Hardy, W. Henecka, H. Ivey-Law et al., "Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption," 2017, https://arxiv.org/abs/1711.10677.

[10] K. Yang, T. Fan, T. Chen, Y. Shi, and Q. Yang, "A quasi-Newton method based vertical federated learning framework for logistic regression," 2019, https://arxiv.org/abs/1912.00513.

[11] K. Mandal and G. Gong, "PrivFL: practical privacy-preserving federated regressions on high-dimensional data over mobile networks," in *Proceedings of the 10th ACM SIGSAC Conference on Cloud Computing Security Workshop*, pp. 57–68, London, England, November 2019.

[12] Y. Zhang, G. Bai, X. Li, C. Curtis, and R. K. L. Ko, "PrivColl: practical privacy-preserving collaborative machine learning," in *Proceedings of the 25th European Symposium on Research in Computer Security*, pp. 399–418, Guildford, UK, September 2020.

[13] Y. Liu, T. Fan, T. Chen, Q. Xu, and Q. Yang, "FATE: an industrial grade platform for collaborative learning with data protection," *Journal of Machine Learning Research*, vol. 22, no. 226, pp. 1–6, 2021.

[14] M. D. Cock, R. Dowsley, A. C. A. Nascimento, D. Railsback, J. W. Shen, and A. Todoki, "High performance logistic regression for privacy-preserving genome analysis," *BMC Medical Genomics*, vol. 14, no. 1, pp. 1–18, 2021.

[15] C. Wang, J. Xu, and L. Yin, "A secure cloud-edge collaborative logistic regression model," in *Proceedings of the IEEE Congress on Cybermatics/14th IEEE International Conference on Internet of Things/14th IEEE International Conference on Cyber, Physical and Social Computing/17th IEEE International Conference on Green Computing and Communications/7th IEEE International Conference on Smart Data*, pp. 244–253, Electric Network, Melbourne, Australia, December, 2021.

[16] R. Zhu, C. Jiang, X. Wang, S. Wang, H. Zheng, and H. Tang, "Privacy-preserving construction of generalized linear mixed model for biomedical computation," *Bioinformatics*, vol. 36pp. 128–135, supplement_1, 2020.

[17] S. Yang, B. Ren, X. Zhou, and L. Liu, "Parallel distributed logistic regression for vertical federated learning without third-party coordinator," 2019, https://arxiv.org/abs/1911.09824.

[18] C. Chen, B. Wu, L. Wang, C. Chen, and B. Zhang, "Nebula: a scalable privacy-preserving machine learning system in ant financial," in *Proceedings of the 29th ACM International Conference on Information and Knowledge Management, Electr Network*, pp. 3369–3372, Ireland, October, 2020.

[19] Q. Li, Z. Huang, W. J. Lu et al., "HomoPAI: a secure collaborative machine learning platform based on homomorphic encryption," in *Proceedings of the 36th International Conference on Data Engineering*, pp. 1713–1717, Dallas, USA, April, 2020.

[20] Q. J. Wei, Q. Li, Z. P. Zhou, Z. Q. Ge, and Y. G. Zhang, "Privacy-preserving two-parties logistic regression on vertically partitioned data using asynchronous gradient sharing," *Peer-to-Peer Networking and Applications*, vol. 14, no. 3, pp. 1379–1387, 2020.

[21] C. Chen, J. Zhou, L. Wang et al., "When homomorphic encryption marries secret sharing: secure large-scale sparse logistic regression and applications in risk control," in *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2652–2662, Singapore, August, 2021.

[22] A. R. Ghavamipour, F. Turkmen, and X. Jiang, "Privacy-preserving logistic regression with secret sharing," *BMC Medical Informatics and Decision Making*, vol. 22, no. 1, pp. 89–11, 2022.

[23] D. He, R. Du, S. Zhu, M. Zhang, K. Liang, and S. Chan, "Secure logistic regression for vertical federated learning," *IEEE Internet Computing*, vol. 26, no. 2, pp. 61–68, 2022.

[24] H. Sun, Z. Wang, Y. Huang, and J. Ye, "Privacy-preserving vertical federated logistic regression without trusted third-party coordinator," in *Proceedings of the 6th International Conference on Machine Learning and Soft Computing*, pp. 132–138, Singapore, January, 2022.

[25] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceedings of the Advances in Cryptology - EUROCRYPT 1999: International Conference on the Theory and Application of Cryptographic techniques*, pp. 223–238, Prague, Czech Republic, May, 1999.

[26] A. C. Yao, "Protocols for secure computations," in *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, pp. 1–5, Chicago, Illinois, USA, November, 1982.

[27] Z. Li, Z. Huang, C. Chen, and C. Hong, "Quantification of the leakage in federated learning," 2019, https://arxiv.org/abs/1910.05467.

[28] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[29] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Proceedings of the Advances in Cryptology - ASIACRYPT 2017: 23rd International Conference on the Theory and Application of Cryptology and Information Security*, pp. 409–437, Hong Kong, China, December, 2017.

[30] A. Kim, Y. Song, M. Kim, K. Lee, and J. H. Cheon, "Logistic regression model training based on the approximate homomorphic encryption," *BMC Medical Genomics*, vol. 11, no. S4, pp. 83–31, 2018.

[31] M. Joye and B. Libert, "Efficient cryptosystems from 2k-th power residue symbols," in *Proceedings of the Advances in Cryptology - EUROCRYPT 2013: 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 76–92, Athens, Greece, May, 2013.

[32] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, Dallas, Texas, USA, November, 2017.

[33] B. Dan, S. Laur, and J. Willemson, "Sharemind: a framework for fast privacy-preserving computations," in *Proceedings of the 13th European Symposium on Research in Computer Security*, pp. 192–206, Málaga, Spain, October, 2008.

[34] M. De Cock, R. Dowsley, C. Horst et al., "Efficient and private scoring of decision trees, support vector machines and logistic regression models based on pre-computation," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 2, pp. 217–230, 2019.

[35] D. Reich, A. Todoki, R. Dowsley, M. D. Cock, and A. Nascimento, "Privacy-preserving classification of personal text messages with secure multi-party computation: an application to hate-speech detection," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 3757–3769, Vancouver, Canada, December, 2008.

[36] H. Chen, W. Dai, M. Kim, and Y. Song, "Efficient multi-key homomorphic encryption with packed ciphertexts with application to oblivious neural network inference," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pp. 395–412, London, United Kingdom, November, 2019.

[37] W. Fang, C. Chen, J. Tan et al., "A hybrid-domain framework for secure gradient tree boosting," 2020, https://arxiv.org/abs/2005.08479.

[38] E. Boyle, N. Gilboa, and Y. Ishai, "Function secret sharing," in *Proceedings of the Advances in Cryptology – EUROCRYPT: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 337–367, Sofia, Bulgaria, April, 2015.

[39] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*, pp. 265–284, Springer, New York, NY, USA, 2006.

[40] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Proceedings of the Advances in Cryptology - EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 1–23, French Riviera, June, 2010.

[41] Microsoft Research and W. A. Redmond, "Microsoft SEAL (release 4.0)," mar, 2022, https://github.com/Microsoft/SEAL.

[42] O. Goldreich, *Foundations of Cryptography: Volume.I, Basic Applications*, Cambridge University Press, Cambridge , UK, 2006.

[43] A. Datta, J. C. Mitchell, and A. Ramanathan, "On the relationships between notions of simulation-based security," *Journal of Cryptology*, vol. 21, pp. 492–546, 2008.