An explainable secure federated collaborative credit modelling method based on blockchain

Dengjia Li[a,c], Chaoqun Ma[a,c], Jinglan Yang[a,c] and Yuncheng Qiao[b,c,*]

[a]Business School, Hunan University, Changsha 410082, China
[b]Business School, Shandong University of Technology, Zibo 255000, China
[c]Research Institute of Digital Society and Blockchain, Hunan University, China

ARTICLE INFO

Keywords:
Credit modelling
Federated learning
Blockchain
CKKS homomorphic encryption
SHAP values

ABSTRACT

Faced the privacy, security and credible requirements in collaborative credit modelling, and the inherent deficiencies of blockchain and federated learning, this paper proposes a privacy-preserving federated collaborative credit modeling method (BPFL) through federated learning empowered by Hyperledger Fabric blockchain, as well as the privacy guarantee of CKKS homomorphic encryption technology. Specifically, the decentralized storage and learning architecture of Hyperledger Fabric blockchain and federated learning achieves secure and trustworthy circulation of multi-party credit data. CKKS-based BPFL guarantees compliant modeling of vertical federated logistic regression. Moreover, the deployment of SHAP values-based contributivity calculation contracts enhances the explanation of credit default prediction and simultaneously realizes the effective monitoring of all participants. The convergence analysis and experimental system validates the feasibility and effectiveness of the proposed solution.

1. Introduction

Financial technologies, which are supported and enabled by digital technologies such as big data, the mobile Internet and artificial intelligence, have had a significant impact on the financial system, greatly strengthening the “public" accessibility of financial services and meeting various “short, small, frequent, urgent and scattered" credit demands[1]. However, with more convenient financial credit activities, the number of credit defaults and instances of fraud have also increased. According to statistics from the central bank, credit cards more than half a year overdue reached 917.5 billion yuan in 2022, and according to the Ministry of Public Security, more than 2 trillion yuan was defrauded in the same year. Therefore, a high-quality credit evaluation model is crucial for credit institutions, banks and customers.

A good credit evaluation model must be supported by multisource and high-quality data. With the rapid development of digital applications such as online shopping and mobile payments, massive and diversified data are being provided to credit institutions, offering crucial decision support for a comprehensive and accurate understanding of the credit status of enterprises and individuals. However, these data are scattered across different organizations and information systems[2]. The privacy of the data and compliance restrictions on data transmission create considerable barriers to data sharing among multiple institutions[3]. In addition, abundant false data are mixed in with the massive amounts of data, which has a serious impact on the accuracy of evaluation[4]. Therefore, addressing the issues of data authenticity and privacy protection in sharing are crucial for multi-party collaborative modelling.

Unfortunately, existing privacy protection schemes rely mainly on the data sharing-aggregation mode centred on a third party or central server. Since it is difficult to find institutions with sufficient credibility to assume the role of credible third parties, this approach may have considerable security risks[5]. Blockchain, utilizing block-chain-based data storage with timestamps, digital signatures, multi-node consensus, and a decentralized transaction processing approach, can ensure the authenticity, immutability, and traceability of credit data, eliminating potential threats to the central server and third party. It provides the possibility of secure data sharing and collaborative modelling of multi-party data[6, 7]. However, the public verifiability of the blockchain is a security issue that cannot be ignored, and seriously affects the process of open sharing of data in the blockchain[8]. Therefore, how to solve the “spear" and “shield" problem between data sharing and privacy protection, public verifiability and privacy protection remains an important problem to solve for blockchain technology.

As a popular distributed machine learning framework, federated learning can realize efficient collaborative modelling between different institutions on the premise that the data of all participants are not domain[9, 10, 11, 12]. The combination of blockchain and federated learning technology provides a good approach for solving the problems of privacy protection in data sharing and insufficient compliance in data collaborative modelling[13, 14]. However, federated learning requires shared model parameters (or gradients) in the modelling process, which carries a risk of privacy leakage. Compared to the differential privacy technology, which ensures data privacy at the expense of some data utility, the computability of homomorphic encryption ciphertext realizes the availability of data without affecting the accuracy of the model[15]. Therefore, how to

*Corresponding author
qiaoyc@hnu.edu.cn (Y. Qiao)

use these technologies to safely and accurately realize multi-party data sharing and collaborative modelling, and achieve a wide range of data and value collaboration has become an important problem to solve in current credit modelling and is also the main goal of our research.

Considering the privacy, security and credible requirements in collaborative credit modelling, by leveraging the decentralized storage and learning architecture of Hyperledger Fabric blockchain and federated learning, combined with CKKS homomorphic encryption technology and SHAP (Shapley Additive Explanations) value contribution calculation contracts, we propose an explainable secure federated collaborative credit modelling method based on blockchain. This approach effectively solves the issues of trustworthiness in collaborative modeling, security and compliance inadequacies in data sharing and aggregation, and the lack of interpretability in assessment results. The contributions of this paper are as follows:

- Based on the innovative combination of Hyperledger Fabric blockchain and federated learning, a new architecture (BPFL) is proposed, which is suitable for the secure and trustworthy circulation of multi-party credit data as well as vertical federated logistic regression collaborative modeling. This architecture effectively avoids potential privacy leakage and security threats in traditional centralized model training, while ensuring the consistency of model parameters and the credibility of collaborative modeling.
- Compared to advanced BFV and Paillier homomorphic encryption, CKKS homomorphic encryption-based BPFL not only effectively achieves secure aggregation of shared gradients in federated learning but also enhances computational efficiency while maintaining model accuracy.
- The contributivity calculation contract based on Shapley additive explanations (SHAP) clearly demonstrates the feature contribution of each participant, which enhances the explanations of model decisions but also enables effective monitoring by each participant.

The remainder of this paper is organized as follows. We begin by introducing some related works in Section 2. Section 3 describes the secure and trustworthy collaborative training architecture. The detailed design of BPFL is given in Section 4. Section 5 demonstrates the convergence of SGD-BPFL and MiniSGD-BPFL. The experiment analysis of BPFL is presented in Section 6. Finally, we concludes this work in Section 7.

2. Literature Review

2.1. Blockchain-based credit evaluation

In recent years, financial technology innovation has accelerated the digital and intelligent transformation of finance. Enabled by digital technologies such as big data, artificial intelligence, cloud computing, and blockchain, financial credit institutions are gradually using digital transaction credit as the basis for providing financial services. Ledger data-based credit modeling has become a new paradigm for collaborative credit assessment. Zhang et al.[16] mitigated information asymmetry between lenders and borrowers by establishing a credit data sharing consortium. Chakraborty et al.[17] improved the credit decision-making process using advanced technologies such as blockchain and machine learning, alleviating the fragmentation of information and enhancing the efficiency of current credit scoring in evaluating creditworthiness. Yang et al.[7] enhanced the integrity and traceability of credit data based on ledger data storage mechanisms. To prevent potential data leakage risk of the third-party platform and centralized design in the traditional supply chain finance management scheme, Li et al.[18] achieved secure data storage through the automated execution of smart contracts and attribute-based access control. Zheng et al.[19] combined blockchain and proxy re-encryption technologies to construct a supply chain financial credit system that ensures user security, data security, access security, and sharing security, thereby promoting the secure sharing of corporate credit information and laying the foundation for improving supply chain financing efficiency.

Blockchain, with its traceability, tamper-resistance, and distributed consensus mechanism, provides a trustworthy execution environment for multi-party collaborative credit modeling. However, ensuring data privacy while achieving high data availability is not only a bottleneck of traditional encryption methods but also an inherent problem hindering the open progress of blockchain technology.

2.2. FL and blockchain-based credit evaluation

The distributed machine learning architecture of federated learning aligns well with the distributed network structure of blockchain. Without disclosing raw data, participants achieve collaborative credit modeling through model sharing, which enhances the security of the blockchain system to a certain extent[20, 21, 14]. Additionally, the decentralized execution framework of blockchain effectively mitigates the single point of failure risk associated with the centralized aggregation server in federated learning. Blockchain's immutable, traceable, publicly verifiable, and distributed consensus mechanisms alleviate the trust issues among different participants in federated learning[22, 23]. The dual drive of blockchain and federated learning promotes the secure, trustworthy flow and compliant collaborative modeling of multi-party credit data.

Rückel et al.[24] achieved fairness, integrity, and privacy in multivariate linear regression (LR) modeling by innovatively integrating emerging technologies such as blockchain, local differential privacy, and zero-knowledge proofs with

Table 1
Comparison of our work with other previous works.

Paper	Blockchain	Federated learning	Privacy Computation	Explainable Artificial Intelligence	Detailed design
Yang et al.[7]	✓				✓
Zhang et al.[16]	✓				✓
Chakraborty et al.[17]	✓				
Li et al.[18]	✓				✓
Zheng et al.[19]	✓				✓
Rückel et al.[24]	✓	✓	✓		✓
Zhang et al.[25]	✓	✓	✓		✓
Jia et al.[26]	✓	✓	✓		✓
Jovanovic et al.[27]	✓	✓	✓	✓	
Dumitrescu et al.[28]				✓	✓
Gunnarsson et al.[29]				✓	✓
Our work	✓	✓	✓	✓	✓

the federated learning (FL) architecture, thereby enhancing the practical applicability of FL. Based on on-chain model parameter updates, Zhang et al.[25] established a credit-based economic model among participants to ensure fair collaboration in FL. Moreover, the combination of homomorphic encryption and proxy re-encryption technologies forms a key-secure sharing participation permission strategy, ensuring the privacy of participants' identities and information. Addressing security vulnerabilities commonly faced in federated learning, such as model extraction and model inversion attacks, Jia et al.[26] proposed a secure federated aggregation scheme adapted to Kmeans clustering, random forest and AdaBoost, which combines differential privacy, homomorphic encryption, and blockchain, achieves secure and trustworthy sharing of data and models.

2.3. Summary

Despite the significant advantages of federated learning empowered by blockchain and privacy computing in secure, trustworthy data flow and compliant collaborative modeling, current federated learning lacks explainability in its evaluation results, making it difficult to provide reliable credit evaluation services[27]. It is essential to understand that an explainable federated learning model not only enhances the transparency of credit evaluation services and strengthens users' trust in credit decisions, but also helps identify and understand potential risk factors, thereby further optimizing risk management strategies during decision-making processes. Given the frequent data breaches, credit fraud, and increasing compliance restrictions, the explainability of credit assessment models has attracted considerable attention[28, 29]. However, this area of research is still in its infancy. Although Jovanovic et al.[27] proposed an automatic credit scoring decision framework with model verifiability, explainable and reliable decision-making by the integration of blockchain and federated learning, it remains a conceptual framework without detailed design or performance evaluation.

Table 1 summarizes previous efforts to develop secure, trustworthy, and privacy-preserving credit assessment systems, showing the gap in the literature addressed in the current study. For few works, trustworthy, privacy, and explainability tradeoff are null. BPFL achieves trustworthy data flow, compliant collaborative modeling compared to existing works and guarantees explainability with detailed design and performance evaluation.

3. Preliminaries

3.1. CKKS Homomorphic encryption

Unlike the BFV scheme[30], which only supports integer addition and multiplication homomorphic operations, and the Paillier scheme[31], which only supports addition homomorphism, CKKS (Cheon-Kim-Kim-Song)[32] is a scheme that supports homomorphic operations on encrypted real and complex numbers. This makes it very useful in applications that need to handle floating-point numbers. Through approximate computation, CKKS can significantly reduce computational complexity and improve the efficiency of homomorphic operations. The CKKS scheme consists of three basic algorithms: Key Generation (KeyGen), Encryption (Enc), and Decryption (Dec)[33].

- **Key Generation:** $KeyGen(1^{\lambda}) \rightarrow (pk, sk, evk)$
  where, λ is the security parameter, q_i is the ciphertext modulus, and $q_i = p^i \cdot q_0, i = 0, 1, \cdots, L$, p acts as a foundational base, scaling each operation, $sk = s \in \mathcal{R}_q, pk = (\natural, \sharp), \natural \in \mathcal{R}_q, \sharp = -\natural \cdot s + e$, $\mathcal{R}_q$ refers to the polynomial ring modulo q, $\mathcal{R}_q = \mathbb{Z}_q[x]/ < f(x) >$, e is an error polynomial sampled from the error distribution χ.

- **Encryption:** $Enc(pk, m) \rightarrow (c_0, c_1)$
  where, $c_0 = p \cdot \triangle \cdot m + \natural \cdot r + e_1$, $c_1 = \sharp \cdot r + e_2$, $\triangle$ is a pre-selected constant used to scale real or complex messages into integer form, e_1 and e_1 are error polynomials.

- **Decryption:** $Dec(s, c) \rightarrow (m')$
  where, $c = (c_0, c_1)$, $m' = \frac{1}{\triangle} \cdot (Round(c_0 + c_1 \cdot s))$, $Round(\cdot)$ is used to round the result to the nearest integer, reducing errors caused by floating-point representation.

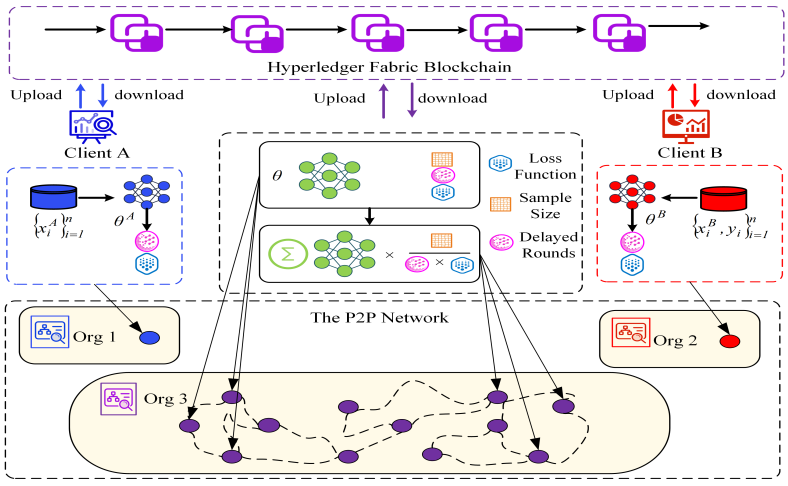


Figure 1: A secure and trustworthy collaborative training architecture.

3.2. CKKS Homomorphic Operations

- **Homomorphic Addition:** $c_{Add} = (c_{0,1} + c_{0,2}, c_{1,1} + c_{1,2})$
- **Homomorphic Multiplication:** $c_{mult} = (Rescale(c_{0,1} \cdot c_{0,2}), Rescale(c_{1,1} \cdot c_{1,2})$
  where, $Rescale(\cdot)$ is used to reduce the modulus of the ciphertext after homomorphic multiplication, while also reducing the scale factor of the ciphertext. $Rescale(c) = ([\frac{c_0}{p}], [\frac{c_1}{p}])$, which helps control the growth of ciphertext noise and maintains an appropriate ciphertext size.
- **Rotation:** $c_{rot} = (Rotate(c_0, k), Rotate(c_1, k))$
  where, the rotation operation $Rotate(\mathbf{c}, k)$ cyclically shifts the elements of the vector by k positions. Here, k can be positive (shift to the right) or negative (shift to the left), i.e., $Rotate(\mathbf{c}, k) = (c_{n-k}, c_{n-k+1}, \cdots, c_{n-1}, c_0, c_1, \cdots, c_{n-k-1}), \mathbf{c} = (c_0, c_1, \cdots, c_{n-1})$.

4. System architecture

4.1. Architecture of BPFL

To address the issues of traditional coordinator-centered federated collaborative training security and compliance deficiencies, as shown in Figure 1, this paper provides a secure and trustworthy collaborative training architecture (BPFL) for interpretable logistic model utilizing decentralized execution architecture, immutability, and chaincode automatic execution of Hyperledger Fabric blockchain. BPFL is composed of three components: Peer-to-Peer (P2P) network layer, Ledger layer, and Application layer.

4.1.1. P2P network layer

P2P network is a decentralized network architecture that operates without a central authority or server, making it more resilient to failures and attacks. Each node is autonomous and can independently initiate or complete tasks, rendering the network dynamic and self-organizing. Nodes in different geographical regions can directly share information with each other, enabling credit institutions to quickly perform cross-organizational collaborative modeling. The network layer provides essential functional modules for the secure and compliant operation of a vertical federated logistic regression model. These modules include Fabric CA, which is responsible for issuing, managing, and revoking certificates for entities within the network, endorsement policies, ordering services, distributed consensus, and chaincode (also known as smart contracts) related to the operation of the blockchain network.

In the network, multiple computing nodes are randomly selected to replace the centralized coordinator, combined with CKKS homomorphic encryption technology, to achieve the compliance, efficient interaction and aggregation of gradient information between party A and party B. The privacy-preserving Taylor loss function and the corresponding gradients for interpretability logistic model are as follow.

$$\begin{aligned}[[\mathfrak{L}]] &= \frac{1}{n}\sum_{i=1}^{N}\{-y_i([[\varpi_i^A]]+[[\varpi_i^B]])+\frac{1}{2}([[\varpi_i^A]]+[[\varpi_i^B]]) \\ &+ \frac{1}{8}[[(\varpi_i^A)^2]]+\frac{1}{8}[[(\varpi_i^B)^2]]+\frac{1}{4}[[\varpi_i^A\times\varpi_i^B]]\}, \end{aligned} \tag{1}$$

$$\begin{aligned}[[\frac{\partial\mathfrak{L}}{\partial\theta_A}]] &= \frac{1}{n}\{\sum_{i=1}^{N}-[[y_i x_i^A]]+\frac{1}{2}[[x_i^A]]+\frac{1}{4}[[\varpi_i^A x_i^A]] \\ &+ \frac{1}{4}[[\varpi_i^B]]x_i^A\}, \end{aligned} \tag{2}$$

$$\begin{aligned}[[\frac{\partial\mathfrak{L}}{\partial\theta_B}]] &= \frac{1}{n}\{\sum_{i=1}^{N}-[[y_i x_i^B]]+\frac{1}{2}[[x_i^B]]+\frac{1}{4}[[\varpi_i^B x_i^B]] \\ &+ \frac{1}{4}[[\varpi_i^A]]x_i^B\}, \end{aligned} \tag{3}$$

where, θ are coefficients of features to be learned by the model, $\theta = [\theta^A, \theta^B]$, $\varpi_i^A = \theta^A x_i^A, \varpi_i^B = \theta^B x_i^B$, $[[\cdot]]$ is

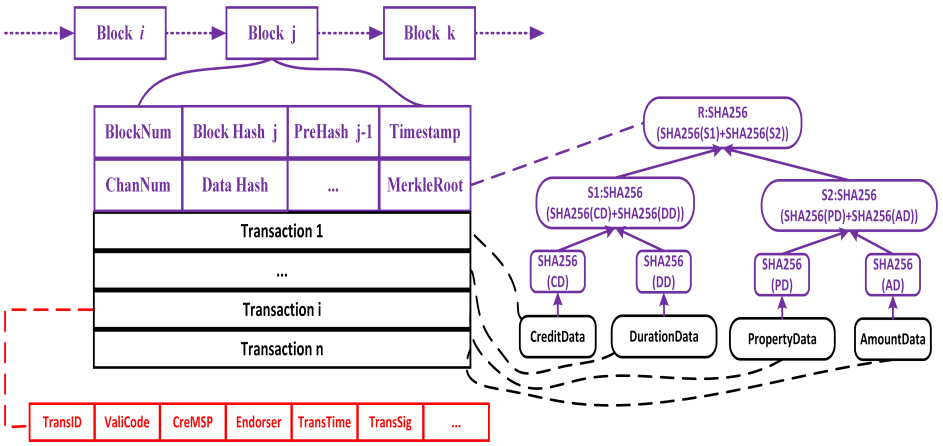


Figure 2: Blockchain Structure.

CKKS homomorphic encryption operator. $[[\cdot]]_A$, $[[\cdot]]_B$, and $[[\cdot]]_C$ are CKKS homomorphic encryption operators with public keys from party A, B, and C.

If most of the computing nodes (generally more than two-thirds) can obtain the same calculation result, then the aggregation results reach consensus, and the aggregation results are subsequently uploaded to the blockchain.

4.1.2. Ledger layer

The ledger layer in Hyperledger Fabric is the core component responsible for storing blockchain data, ensuring data integrity and immutability. The ledger layer primarily consists of the Blockchain Log, which records data in a sequential block-chain, and the World State, which is managed by the LevelDB storage engine for key-value storage, querying, and updating (though CouchDB, a document-oriented NoSQL database suitable for storing and querying complex JSON documents, is also commonly used as a storage engine; this article focuses on the LevelDB storage engine as an example).

In the blockchain, only the hash values related to shared data during federated collaborative credit modelling (e.g., R:SHA256) and identifying information such as the hash values of the original data (e.g., SHA256(CD), SHA256(DD), SHA256(PD), SHA256(AD)) are stored. The Blockchain Log structure of the blockchain-based federated collaborative credit evaluation system mainly comprises two parts: block header and block body. As shown in Figure 2, the block header records the identifying information of the current block and its contained transactions, while the block data and block metadata together form the block body to store all transactions and their related verification information.

In the block header, BlockNum is the sequence number of the block, Block Hash is the hash value of the current block, PreHash is the hash value of the previous block, Timestamp is the timestamp of block creation, ChanNum is the channel number to which the current block belongs, Data Hash represents the hash value of all transaction data in the current block, and MerkleRoot is the Merkle tree root of all transaction data in the block, enabling fast transaction verification. The block body contains the transactions and their related validation information, as well as the creator's validation information. Within this, TransID represents the transaction ID, ValiCode is the transaction validation code, CreMSP is the MSP name of the transaction creator, Endorser is the endorser of the transaction, TransTime is the creation time of the transaction, and TransSig is the transaction signature.

The World State is a representation of the current state of the blockchain, storing the latest key-value pair data. It provides a method for quickly accessing the current valid data without needing to traverse the entire Blockchain Log. As shown in Figure 3, by setting parameters such as block number, block hash, and transaction ID as keys, and setting the block path as the value in a key-value storage, a corresponding index list for the blocks is generated and stored in LevelDB. Using these key-value indexes, data requesters can quickly query the off-chain stored information corresponding to them. Simultaneously, by verifying the hash value of the off-chain data with the hash value of the corresponding on-chain data, the validity and integrity of the off-chain data can be ensured.

4.1.3. Application layer

The physical application layer of the blockchain is an application client. In this layer, participants can use mobile phones, computers, and other terminal devices to interact with the relevant chaincode via the Fabric Software Development Kit (SDK), facilitating interactions and collaborative modeling with other participants without needing to understand the underlying blockchain technology. Application Programming Interfaces (APIs) serve as a good practice to abstract this process, making it easier to connect applications and various participating entities.

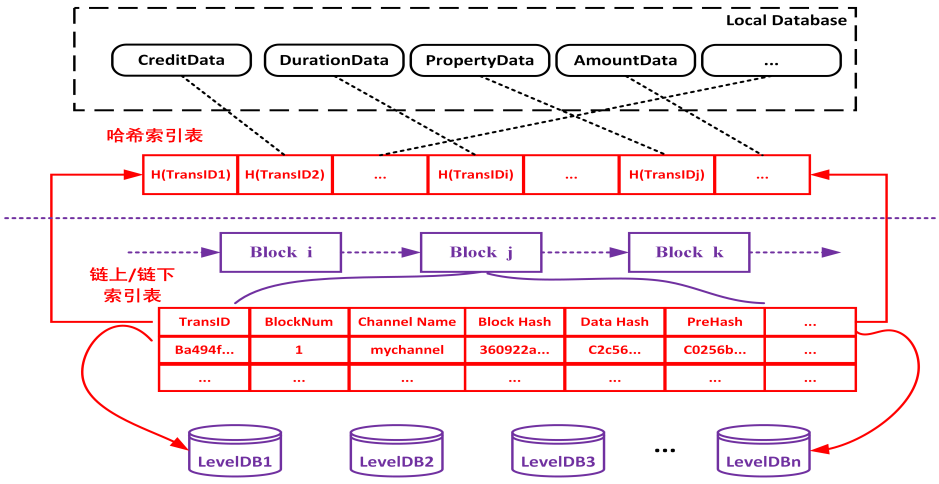


Figure 3: Blockchain encrypted storage structure.

5. Detailed design of the BPFL system

5.1. Blockchain-based private set intersection(BC-PSI)

Considering credit datasets with vertical distribution characteristics, i.e., datasets with the same sample space but different feature spaces, and to avoid the risk of privacy leakage brought by traditional centralized ID comparison, a smart contract-based decentralized private set intersection (BC-PSI) method is proposed. This method completes the common sample ID comparison of all participants by uploading encrypted ID information on the blockchain. Assuming N participants, the sample ID set for each participant is$\{U_1, U_2, U_3, \cdots, U_N\}$. The detailed BC-PSI method as follows.

5.1.1. System initialization

Each participant P_i generates its own public key pk_i and private key pair sk_i on the blockchain. Specifically, Fabric-CA generates the global initialization system parameter $(\mathbb{G}, G)$ under given security parameters λ. Each participant P_i selects a random exponent $\alpha \in \mathbb{G}_p^*$ and uses this random exponent α as the private key sk_i. Using the generator G, they generate the public key pk_{P_i}, i.e., $\mathcal{G}(\alpha, G) \rightarrow pk_i$, where $sk_{DP} = \alpha, pk_{DP} = \alpha G$. Each participant then broadcasts their public key to the blockchain, and upon reaching consensus, the $Block_t$ is updated. Algorithm 1 outlines the whole process.

Algorithm 1 Initialization

Input: Participants $P = \{P_i\}_{i=1}^N$
1: $(\mathbb{G}, G) \longleftarrow Setup(1^\lambda)$
2: **for** Each participant P_i **do**
3: $(pk_i, sk_i) \longleftarrow \mathcal{G}(\alpha, G)$
4: **end for**
5: $Block_t \leftarrow \{pk_i\}$
6: **broadcast** $Block_t$ to all participants
7: **for** Each participant i, $i = 1, 2, \cdots, n$ **in parallel do**
8: **update** $Block_t$ to the global ledger.
9: **end for**

5.1.2. Encrypted data uploading and storage

To protect the privacy of participant identities and sample ID, each participant i hashes their own sample ID, SHA_{U_i}, and then encrypts it to obtain the encrypted dataset Y_i using their own public key. Uploading the encrypted dataset Y_i to blockchain, and updating $\{Y_i, pk_i\}$ to the global ledger. Y_i is associated with each participant's identity and public key through the use of smart contracts. Algorithm 2 delineates the detailed encrypted data upload and storage(EUS) procedure.

Algorithm 2 EUS Algorithm

Input: Sample $ID = \{U_i\}_{i=1}^N$
1: **for** Each sample ID in $\{U_i\}$ **do**
2: $SHA_{U_i} = Hash(U_i)$
3: $Y_i = Encrypt(SHA_{U_i}, pk_i)$
4: **end for**
5: $Block_{t+1} \leftarrow \{Y_i, pk_i\}$
6: **broadcast** $Block_{t+1}$ to all participants
7: **for** Each participant i, $i = 1, 2, \cdots, n$ **in parallel do**
8: **update** $Block_{t+1}$ to the global ledger.
9: **end for**

5.1.3. Private set intersection

Each participant i retrieves the encrypted dataset $Y_j (i \neq j)$ from blockchain and encrypts its own dataset using the public key of the other $N-1$ participant. The encrypted $N-1$ dataset is compared with the encrypted dataset of other nodes to find the common data sample U_{common_k} of participant i. The results are uploaded to the blockchain, and the smart contract is invoked to determine the minimum intersection $\{U_{common_1}, U_{common_2}, \cdots, U_{common_k}\}$. The Private Set Intersection(PSI) algorithm is implemented in Algorithm 3.

Algorithm 3 PSI Algorithm

Input: Participants $P = \{P_i\}_{i=1}^N$, and Sample $ID = \{U_i\}_{i=1}^N$
1: **for** Each sample ID in $\{U_i\}$ **do**
2: **if** $j! = i$ **then**
3: $Y_j = Download(SHA_{U_j})$

$Y_{i,j} = Encrypt(SHA_{U_i}, pk_j)$
end if
$U_{common_k} = common_{ID}.intersection\{Y_{i,j}, Y_j\}$
end for
$Block_{t+k} \leftarrow \{U_{common_1}, U_{common_2}, \cdots, U_{common_k}, pk_i\}$
broadcast $Block_{t+k}$ to all participants
for Each participant $i,\ i = 1, 2, \cdots, n$ **in parallel do**
update $Block_{t+k}$ to the global ledger.
end for

5.2. Blockchain-based privacy-federated collaborative logistic regression modelling(BPFL)

The entire process of BPFL is summarized into four main stages that interact involve interactions among different entities, namely, data encryption and sharing(DES), secure collaborative modeling(SCM), model aggregation and decryption(MAD), and federated model training(FMT). Taking the logistic regression modelling by participants A and B as an example, the complete workflow of BPFL as follows.

5.2.1. Data encryption and sharing(DES)

① Party A calculates ϖ_i^A and $(\varpi_i^A)^2$, and then encrypts them as $[[\varpi_i^A]]_A$ and $[[(\varpi_i^A)^2]]_A$ by CKKS homomorphic encryption, respectively. Uploading these encrypted coefficients to blockchain.

② Similarity, party B calculates ϖ_i^B and $(\varpi_i^B)^2$ by CKKS homomorphic encryption, and then uploads its encrypted $[[\varpi_i^B]]_B$, $[[(\varpi_i^B)^2]]_B$ to blockchain.

Algorithm 4 DES Algorithm

Input: θ^π, $\mathfrak{D} = (x_i^\pi, y_i^\pi)$, $\pi = A, B$
for Party π, $\pi = A, B$ **in parallel do**
$(\varpi_i^\pi, (\varpi_i^\pi)^2) \longleftarrow (\theta^\pi, x_i^\pi)$.
$([[\varpi_i^\pi]]_\pi, [[(\varpi_i^\pi)^2]]_\pi) \longleftarrow (\varpi_i^\pi, (\varpi_i^\pi)^2)$.
end for
$Block_{t+l} \leftarrow \{[[\varpi_i^\pi]]_\pi, [[(\varpi_i^\pi)^2]]_\pi, pk_\pi\}$
broadcast $Block_{t+l}$ to all participants
for Each participant $i,\ i = 1, 2, \cdots, n$ **in parallel do**
update $Block_{t+l}$ to the global ledger.
end for

5.2.2. Secure collaborative modelling(SCM)

③ Downloaded $[[\varpi_i^B]]_B$ and $[[(\varpi_i^B)^2]]_B$ from blockchain, party A computes $[[\mathfrak{L}]]_B$ and $[[\frac{\partial \mathfrak{L}}{\partial \theta_A}]]_B$ via (1) and (2). Then party A creates random mask τ_A and adds it to $[[\frac{\partial \mathfrak{L}}{\partial \theta_A}]]_B$ to obtain $[[\frac{\partial \mathfrak{L}}{\partial \theta_A} + \tau_A]]_B$. Uploading $[[\mathfrak{L}]]_B$ and $[[\frac{\partial \mathfrak{L}}{\partial \theta_A} + \tau_A]]_B$ to blockchain.

④ Downloaded $[[\mathfrak{L}]]_B$, $[[\frac{\partial \mathfrak{L}}{\partial \theta_A} + \tau_A]]_B$, and $[[\varpi_i^A]]_A$ from blockchain, party B decrypts $[[\mathfrak{L}]]_B$, $[[\frac{\partial \mathfrak{L}}{\partial \theta_A} + \tau_A]]_B$, and computes $[[\frac{\partial \mathfrak{L}}{\partial \theta_B}]]_A$ via (3). Then party B creates random mask τ_B and adds it to $[[\frac{\partial \mathfrak{L}}{\partial \theta_B}]]_A$ to obtain $[[\frac{\partial \mathfrak{L}}{\partial \theta_B} + \tau_B]]_A$. Uploading $[[\frac{\partial \mathfrak{L}}{\partial \theta_B} + \tau_B]]_A$ and $[[\frac{\partial \mathfrak{L}}{\partial \theta_A} + \tau_A]]_C$ to blockchain.

⑤ Party A decrypts $[[\frac{\partial \mathfrak{L}}{\partial \theta_B} + \tau_B]]_A$ downloaded from blockchain, and uploads the encrypted $[[\frac{\partial \mathfrak{L}}{\partial \theta_B} + \tau_B]]_C$ to blockchain.

Algorithm 5 SCM Algorithm

Input: θ^π, $\mathfrak{D} = (x_i^\pi, y_i^\pi)$, $\pi = A, B$
for Party A **do**
$([[\varpi_i^B]]_B, [[(\varpi_i^B)^2]]_B) \longleftarrow Download(Block_{t+l}, pk_B)$.
$[[\mathfrak{L}]]_B \longleftarrow ([[\varpi_i^A]]_A, [[(\varpi_i^A)^2]]_A, [[\varpi_i^B]]_B, [[(\varpi_i^B)^2]]_B)$.
$[[\frac{\partial \mathfrak{L}}{\partial \theta_A}]]_B \longleftarrow ([[y_i x_i^A]], [[x_i^A]], [[\varpi_i^A x_i^A]], [[\varpi_i^B]]x_i^A)$.
$[[\frac{\partial \mathfrak{L}}{\partial \theta_A} + \tau_A]]_B \longleftarrow (\frac{\partial \mathfrak{L}}{\partial \theta_A}, \tau_A)$.
end for
$Block_{t+m} \leftarrow \{[[\mathfrak{L}]]_B, [[\frac{\partial \mathfrak{L}}{\partial \theta_A} + \tau_A]]_B\}$
broadcast $Block_{t+m}$ to all participants
for Each participant $i,\ i = 1, 2, \cdots, n$ **in parallel do**
update $Block_{t+m}$ to the global ledger.
end for
for Party B **do**
$([[\mathfrak{L}]]_B, [[\frac{\partial \mathfrak{L}}{\partial \theta_A} + \tau_A]]_B) \longleftarrow Download(Block_{t+m}, pk_A)$.
$[[\frac{\partial \mathfrak{L}}{\partial \theta_B}]]_A \longleftarrow ([[y_i x_i^B]], [[x_i^B]], [[\varpi_i^B x_i^B]], [[\varpi_i^A]]x_i^B)$.
$[[\frac{\partial \mathfrak{L}}{\partial \theta_B} + \tau_B]]_A \longleftarrow (\frac{\partial \mathfrak{L}}{\partial \theta_B}, \tau_B)$.
$[[\frac{\partial \mathfrak{L}}{\partial \theta_A} + \tau_A]] \longleftarrow Dec([[\frac{\partial \mathfrak{L}}{\partial \theta_A} + \tau_A]]_B, sk_B)$.
$[[\frac{\partial \mathfrak{L}}{\partial \theta_A} + \tau_A]]_C \longleftarrow Enc(\frac{\partial \mathfrak{L}}{\partial \theta_A} + \tau_A, pk_C)$.
end for
$Block_{t+n} \leftarrow \{[[\frac{\partial \mathfrak{L}}{\partial \theta_B} + \tau_B]]_A, [[\frac{\partial \mathfrak{L}}{\partial \theta_A} + \tau_A]]_C\}$
broadcast $Block_{t+n}$ to all participants
for Each participant $i,\ i = 1, 2, \cdots, n$ **in parallel do**
update $Block_{t+n}$ to the global ledger.
end for
for Party A **do**
$([[\frac{\partial \mathfrak{L}}{\partial \theta_B} + \tau_B]]_A) \longleftarrow Download(Block_{t+l}, pk_C)$.
$[[\frac{\partial \mathfrak{L}}{\partial \theta_B} + \tau_B]] \longleftarrow Dec([[\frac{\partial \mathfrak{L}}{\partial \theta_B} + \tau_B]]_A, sk_A)$.
$[[\frac{\partial \mathfrak{L}}{\partial \theta_B} + \tau_B]]_C \longleftarrow Enc(\frac{\partial \mathfrak{L}}{\partial \theta_B} + \tau_B, pk_C)$.
end for
$Block_{t+r} \leftarrow \{[[\frac{\partial \mathfrak{L}}{\partial \theta_B} + \tau_B]]_C\}$
broadcast $Block_{t+r}$ to all participants
for Each participant $i,\ i = 1, 2, \cdots, n$ **in parallel do**
update $Block_{t+r}$ to the global ledger.
end for

5.2.3. Model aggregation and decryption(MAD)

⑥ Computing node C downloaded $[[\frac{\partial \mathfrak{L}}{\partial \theta_A} + \tau_A]]_C$, $[[\frac{\partial \mathfrak{L}}{\partial \theta_B} + \tau_B]]_C$ from blockchain, and then decrypts $[[\frac{\partial \mathfrak{L}}{\partial \theta_A} + \tau_A]]_C$, $[[\frac{\partial \mathfrak{L}}{\partial \theta_B} + \tau_B]]_C$ as $\frac{\partial \mathfrak{L}}{\partial \theta_A} + \tau_A$, $\frac{\partial \mathfrak{L}}{\partial \theta_B} + \tau_B$, respectively. Uploading them to blockchain.

Algorithm 6 MAD Algorithm

Input: θ^π, $\mathfrak{D} = (x_i^\pi, y_i^\pi)$, $\pi = A, B$
for Computing node C **do**
$([[\frac{\partial \mathfrak{L}}{\partial \theta_\pi} + \tau_\pi]]_C,) \longleftarrow Download(Block_{t+n}, Block_{t+r})$.
$\frac{\partial \mathfrak{L}}{\partial \theta_A} + \tau_A \longleftarrow Dec([[\frac{\partial \mathfrak{L}}{\partial \theta_A} + \tau_A]]_C, sk_C)$.
$\frac{\partial \mathfrak{L}}{\partial \theta_B} + \tau_B \longleftarrow Dec([[\frac{\partial \mathfrak{L}}{\partial \theta_B} + \tau_B]]_C, sk_C)$.
end for
$Block_{t+s} \leftarrow \{\frac{\partial \mathfrak{L}}{\partial \theta_A} + \tau_A, \frac{\partial \mathfrak{L}}{\partial \theta_B} + \tau_B\}$
broadcast $Block_{t+s}$ to all participants

8: **for** Each participant $i,\ i = 1, 2, \cdots, n$ **in parallel do**
9: **update** $Block_{t+s}$ to the global ledger.
10: **end for**

5.2.4. Federated model training(FMT)

⑦ Party A and B download $\frac{\partial \mathfrak{L}}{\partial \theta_A} + \tau_A$, $\frac{\partial \mathfrak{L}}{\partial \theta_B} + \tau_B$, respectively. Then, party A obtains gradient $\frac{\partial \mathfrak{L}}{\partial \theta_A}$ by subtracting τ_A; party B obtains gradient $\frac{\partial \mathfrak{L}}{\partial \theta_B}$ by subtracting τ_B.

⑧ Using Mini-batch Stochastic Gradient Descent (abbreviated as MiniSGD), Party A and B update their respective model parameters according to 4 and 5 once the loss $\mathfrak{L}$ converges. Otherwise, goes to step 1 to continue the collaborative training process.

$$\theta^A \doteq \theta^A - \eta \frac{\partial \mathfrak{L}}{\partial \theta_A}, \tag{4}$$

$$\theta^B \doteq \theta^B - \eta \frac{\partial \mathfrak{L}}{\partial \theta_B}, \tag{5}$$

where, η is learning rate.

Algorithm 7 FMT Algorithm

Input: θ^π, $\mathfrak{D} = (x_i^\pi, y_i^\pi)$, $\pi = A, B$
1: **for** Party π, $\pi = A, B$ **in parallel do**
2: $\frac{\partial \mathfrak{L}}{\partial \theta_\pi} + \tau_\pi \longleftarrow Download(Block_{t+s})$.
3: $\frac{\partial \mathfrak{L}}{\partial \theta_\pi} = (\frac{\partial \mathfrak{L}}{\partial \theta_\pi} + \tau_\pi) - \tau_\pi$.
4: allocate $\mathfrak{D}_i \sim \mathfrak{D}$ to π
5: $\mathfrak{B} \longleftarrow$ split $\mathfrak{D}_i$ into batches of size $\mathfrak{B}$
6: **for** each epoch i from 1 to $\mathfrak{E}$ **do**
7: **for** each epoch $b \in \mathfrak{E}$ **do**
8: $\theta^\pi \longleftarrow \theta^\pi - \frac{1}{|\mathfrak{B}|}\eta \frac{\partial \mathfrak{L}}{\partial \theta_\pi}$.
9: **end for**
10: **end for**
11: **end for**
12: $Block_{t+v} \leftarrow \{\partial \mathfrak{L}, \theta^\pi\}$
13: **broadcast** $Block_{t+v}$ to all participants
14: **for** Each participant $i,\ i = 1, 2, \cdots, n$ **in parallel do**
15: **update** $Block_{t+v}$ to the global ledger.
16: **end for**

5.3. SHAP values-based contributivity calculation(SHAP-C)

To enhance the explanations of credit default prediction, the SHAP values-based contributivity calculation contract(SHAP-C) is deployed. Specifically, it calculates the contribution of each feature subset to the model prediction output, and uses these contributions to calculate the SHAP values of each feature to quantify the contribution of each feature to the final output.

The calculation formula of SHAP is as follows:

$$v_i^k = \sum_{T \subseteq I} (\phi_i^k(T)), \tag{6}$$

where v_i^k represents the output of the feature set I at the kth sample point. Specifically, the SHAP value $\phi_i^k(T)$ of the ith feature x_i at the kth sample point can be expressed as follows:

$$\phi_i^k(T) = \frac{|T|!(|I| - |T| - 1)!}{|I|!}[v^k_{T \bigcup \{i\}} - v_T^k], \tag{7}$$

where $\phi_i^k(T)$ represents the contribution of the ith feature to the output when considering subset T, T is a subset of the feature set I, $|T|$ represents the number of elements in set T, $v^k_{T \bigcup \{i\}}$ represents the model output including feature i, and v_T^k represents the model output without including feature i. Algorithm 8 delineates the detailed explainable credit default prediction procedure.

Algorithm 8 SHAP-C Algorithm

Input: model, X, feature set I, k-th sample point
1: **for** each feature i in I **do**
2: $\phi_i^k \leftarrow 0$
3: **for** each subset $T \subseteq I \setminus \{i\}$ **do**
4: $T_{\bigcup\{i\}} \leftarrow T \cup \{i\}$
5: $v_T^k \leftarrow model.predict(X_T)$
6: $v^k_{T_{\bigcup\{i\}}} \leftarrow model.predict(X_{T_{\bigcup\{i\}}})$
7: $weight \leftarrow \frac{|T|!(|I|-|T|-1)!}{|I|!}$
8: $\phi_i^k \leftarrow \phi_i^k + weight \times (v^k_{T_{\bigcup\{i\}}} - v_T^k)$
9: **end for**
10: $v_i^k \leftarrow \sum_{T \subseteq I} (\phi_i^k(T))$
11: **end for**

6. Convergence analysis of BPFL

Compared to using the entire dataset, MiniSGD in Section 5.2.4 allows for rapid iteration and accelerated computation, helping the model converge more smoothly. Under the assumptions of convex functions and bounded gradient variance, this section provides a detailed proof of the convergence of SGD-BPFL and MiniSGD-BPFL.

6.1. Convergence of SGD-BPFL

Assumption 1 (Lipschitz continuity). The global loss function $f(\cdot)$ is L-Lipschitz continuous, i.e., for any parameter vector θ_1, θ_2, there is

$$\|f(\theta_1) - f(\theta_2)\| \leq L\|\theta_1 - \theta_2\|. \tag{8}$$

Assumption 2 (Bounded gradient). The gradient of the local loss function in each client is bounded, i.e., for an arbitrary client k and the parameter vector θ_1, θ_2, there is

$$\|\nabla_k f(\theta)\| \leq G. \tag{9}$$

Assumption 3 (Bounded variance). The variance of stochastic gradients in each client is bounded, i.e., for any client k and the parameter vector θ, there is

$$E[\|\nabla_k f(\theta) - \nabla f(\theta)\|^2] \leq \sigma^2. \tag{10}$$

Theorem 1. Let $f(\cdot)$ be a convex function that satisfies Assumptions 1 to 3, $\theta^* \in argmin_{\theta:\|\theta\|\leq B} f(\theta)$. If the SGD-BPFL runs for T iterations with a step size $\eta = \sqrt{\frac{G^2}{\sigma^2 T}}$, then the following holds,

$$E[f(\theta)] - f(\theta^*) \leq \frac{G\sigma}{T}. \tag{11}$$

Moreover, for any $\epsilon > 0$, to satisfy $E[f(\bar{\theta})] - f(\theta^*) \leq \epsilon$, the number of iterations required by the SGD-BPFL must satisfy $T \geq \frac{G^2\sigma^2}{\epsilon^2}$.

Proof. Assume

$$\begin{aligned}
& E[f(\overline{\theta_A + \theta_B})] - f(\theta^*) \\
= \; & E[f(\overline{\theta_A + \theta_B}) - f(\theta^*)] \\
= \; & E[\frac{1}{T}\sum_{T+1}^{t=1}(f(\theta_A + \theta_B)^{(t+1)}) - f(\theta^*)] \\
\leq \; & E[\frac{1}{T}\sum_{T+1}^{t=1} < (\theta_A + \theta_B)^{(t+1)} - \theta^*, \nabla f(\theta_A + \theta_B) >].
\end{aligned}$$

Expanding the inner product,

$$\begin{aligned}
& \sum_{t=1}^{T+1} < (\theta_A + \theta_B)^{(t)} - \theta^*, \nabla f(\theta_A + \theta_B) > \\
= \; & \sum_{t=1}^{T+1} \frac{1}{2\eta}(\|(\theta_A + \theta_B)^{(t+1)} - \theta^*\|^2 - \|(\theta_A + \theta_B)^{(t)} - \theta^*\|^2 \\
& + \eta^2 \|\nabla f(\theta_A + \theta_B)\|^2) \\
= \; & \frac{1}{2\eta}(\|(\theta_A + \theta_B)^{(T+1)} - \theta^*\|^2 - \|(\theta_A + \theta_B)^{(1)} - \theta^*\|^2) \\
& + \frac{\eta}{2}\sum_{t=1}^{T} \|\nabla f(\theta_A + \theta_B)\|^2 \\
\leq \; & \frac{1}{2\eta}(\|(\theta_A + \theta_B)^{(1)} - \theta^*\|^2) + \frac{\eta}{2}\sum_{t=1}^{T} \|\nabla f(\theta_A + \theta_B)\|^2 \\
\leq \; & \frac{1}{2\eta}\sigma^2 + \frac{\eta}{2}TG^2.
\end{aligned}$$

Applying the basic inequality,

$$\frac{1}{2\eta}\sigma^2 + \frac{\eta}{2}TG^2 \leq G\sigma.$$

Thus,

$$E[f(\overline{\theta_A + \theta_B})] - f(\theta^*) \leq \frac{G\sigma}{\sqrt{T}}.$$

Let $E[f(\overline{\theta_A + \theta_B})] - f(\theta^*) \leq \epsilon$,

$$E[f(\overline{\theta_A + \theta_B})] - f(\theta^*) \leq \frac{G\sigma}{\sqrt{T}} \leq \epsilon.$$

Then,

$$T \leq \frac{G^2\sigma^2}{\epsilon^2}.$$

6.2. Convergence of MiniSGD-BPFL

Definition 1. Let $b \in \{1, 2, \cdots, n\}$, define the gradient variance of MiniSGD as

$$\sigma_b^* \overset{def}{=\!=} \inf_{x^* \in argminf} \nu[\nabla f_B(x^*)]. \tag{12}$$

Definition 2. Let $b \in \{1, 2, \cdots, n\}$. $\forall \theta_1, \theta_2 \in \mathbb{R}$, if function $f(\cdot)$ satisfies (13), then $f(\cdot)$ is said to be L_b-smooth under the expected definition.

$$\begin{aligned}
& \frac{1}{2L_b} E[\|\nabla f_B(\theta_2) - \nabla f_B(\theta_1)\|^2] \\
& \leq f(\theta_2) - f(\theta_1) - < \nabla f(\theta_1), \theta_2 - \theta_1 > .
\end{aligned} \tag{13}$$

Lemma 1. Assume that $f(\cdot)$ satisfies convexity and L-smoothness, then the L_b under the expectation definition of $f(\cdot)$ is

$$L_b = \frac{n(b-1)}{b(n-1)}L + \frac{n-b}{b(n-1)}L_{max}. \tag{14}$$

The gradient of PBFL-MiniSGD can be calculated by

$$\sigma_b^* = \frac{n-b}{b(n-1)}\sigma_f^*. \tag{15}$$

The following discusses the expected smoothing constant L_b and mini-batch gradient noise σ_b^*, when $b = 1$, MiniSGD degenerate into SGD ($L_b = L_{max}, \sigma_b^* = \sigma_f^*$). In this case, the convergence proof of MiniSGD equivalent to the proof of SGD.

Lemma 2. Assume that $f(\cdot)$ satisfies convexity and L-smoothness, we have

$$E[\|\nabla f_B(\theta)\|^2] \leq 4L_b(f(\theta) - \inf f) + 2\sigma_b^*. \tag{16}$$

Theorem 2. Assume that $f(\cdot)$ satisfies convexity and L-smoothness, let $(\theta^t)_{t\in\mathbb{N}}$ is the sequence produced by MiniSGD, if the step size of each θ satisfies $0 < \gamma_t \leq \frac{1}{4L_b}$, then for $T \geq 1, \theta^* \in argminf$ and $\overline{\theta_A + \theta_B}^T = \frac{1}{\sum_{t=0}^{T-1}\gamma_t}\sum_{t=0}^{T-1}\gamma_t\theta_A + \theta_A^t$,

$$E[f(\overline{\theta_A + \theta_B}^T) - \inf f] \leq \frac{\|\theta_A + \theta_B^0 - \theta^*\|}{\sum_{t=0}^{T-1}\gamma_t} + \frac{2\sigma_b^*\sum_{t=0}^{T-1}\gamma_t^2}{\sum_{t=0}^{T-1}\gamma_t}. \tag{17}$$

Proof. Let $\theta^* \in argminf$, then, $\sigma_b^* = \nu[\nabla f_B(x^*)]$. The L_2 norm of the sequence $\|\theta_A + \theta_B^{t+1} - \theta^*\|$ is obtained as follows

$$\begin{aligned}
\|\theta_A + \theta_B^{t+1} - \theta^*\|^2 &= \|\theta_A + \theta_B^t - \gamma_t \nabla f_B(x^t) - \theta^*\|^2 \\
&= \|\theta_A + \theta_B^t - \theta^*\|^2 \\
&\quad -2\gamma_t < \nabla f_B(\theta_A + \theta_B^t), \theta_A + \theta_B^t - \theta^* > \\
&\quad +\gamma_t^2 \|\nabla f_B(\theta_A + \theta_B^t)\|^2
\end{aligned}$$

Using Lemma 1 and Lemma 2,

$$E[\|\theta_A + \theta_B^{t+1} - \theta^*\|^2 | x^t]$$

$$
\begin{aligned}
&= \|\theta_A + \theta_B^t - \theta^*\|^2 - 2\gamma_t < \nabla f_B(\theta_A + \theta_B^t), \theta_A + \theta_B^t - \theta^* > \\
&\quad +\gamma_t^2 E[\|\nabla f_B(\theta_A + \theta_B^t)\|^2 | \theta_A + \theta_B^t] \\
&\leq \|\theta_A + \theta_B^t - \theta^*\|^2 - 2\gamma_t (f(\theta_A + \theta_B^t - \inf f)) \\
&\quad +\gamma_t^2 E[\|\nabla f_B(\theta_A + \theta_B^t)\|^2 | \theta_A + \theta_B^t] \\
&\leq \|\theta_A + \theta_B^t - \theta^*\|^2 - 2\gamma_t (2\gamma_t L_b - 1)(f(\theta_A + \theta_B^t - \inf f)) + 2\gamma_t^2 \sigma_b^* \\
&\leq \|\theta_A + \theta_B^t - \theta^*\|^2 - \gamma_t (f(\theta_A + \theta_B^t) - \inf f) + 2\gamma_t^2 \sigma_b^*
\end{aligned}
$$

The penultimate inequality applies the stride assumption $\gamma_t \leq \frac{1}{4\mathfrak{L}_b}$. Rearranging and moving terms from the above equation yields,

$$
\begin{aligned}
&\gamma_t E[f(\theta_A + \theta_B^t) - \inf f] \\
\leq\ & E[\|\theta_A + \theta_B^t - \theta^*\|^2] - E[\|\theta_A + \theta_B^{t+1} - \theta^*\|^2] + 2\gamma_t^2 \sigma_b^*
\end{aligned}
$$

Sum up the above formula,

$$
\begin{aligned}
&\sum_{t=0}^{T-1} \gamma_t E[f(\theta_A + \theta_B^t) - \inf f] \\
\leq\ & E[\|\theta_A + \theta_B^0 - \theta^*\|^2] - E[\|\theta_A + \theta_B^T - \theta^*\|^2] \\
+\ & 2\sigma_b^* \sum_{t=0}^{T-1} \gamma_t^2
\end{aligned}
$$

Since $E[\|\theta_A + \theta_B^T - \theta^*\|^2] \geq 0$, dividing both sides of the above inequality by $\sum_{t=0}^{T-1} \gamma_t^2$ yields

$$
\begin{aligned}
&\frac{1}{\sum_{t=0}^{T-1} \gamma_t} \sum_{t=0}^{T-1} \gamma_t E[f(\theta_A + \theta_B^t) - \inf f] \\
\leq\ & \frac{\|\theta_A + \theta_B^0 - \theta^*\|^2}{\sum_{t=0}^{T-1} \gamma_t} + \frac{2\sigma_b^* \sum_{t=0}^{T-1} \gamma_t^2}{\sum_{t=0}^{T-1} \gamma_t}
\end{aligned}
$$

Given $\overline{\theta_A + \theta_B}^T = \frac{1}{\sum_{t=0}^{T-1} \gamma_t} \sum_{t=0}^{T-1} \gamma_t \theta^t$, applying Jensen's inequality to f yields

$$
\begin{aligned}
&E[f(\overline{\theta_A + \theta_B}^T) - \inf f] \\
\leq\ & E[\frac{1}{\sum_{t=0}^{T-1} \gamma_t} \sum_{t=0}^{T-1} \gamma_t (f(\theta_A + \theta_B^t) - \inf f)] \\
\leq\ & \frac{\|\theta_A + \theta_B^0 - \theta^*\|}{\sum_{t=0}^{T-1} \gamma_t} + \frac{2\sigma_b^* \sum_{t=0}^{T-1} \gamma_t^2}{\sum_{t=0}^{T-1} \gamma_t}
\end{aligned}
$$

7. Experiment and analysis

7.1. Data and experimental design

This paper uses the German credit dataset to train and validate the proposed BPFL method, which is frequently utilized in credit evaluation research[34, 35, 36]. The dataset contains 1000 samples and 20 features, including status_account, duration, and credit_history, etc (7 continuous and 13 categorical). During the data preprocessing, WOE encoding[37] is used for nonnumerical variables. After encoding and feature selection, as shown in Figure 4(a), only 14 features remain.

To address the dual challenges of high ciphertext computation costs and blockchain communication overhead in vertical federated learning, this paper utilizes the XGBoost model to pre-train a portion of the dataset and obtain feature importance ranking. As shown in Figure 4(a), features such as age, loan duration, and property have a significant impact on the model's accuracy. Based on this, ablation experiments were conducted by sequentially removing the less important features to observe changes in the model's performance. As illustrated in Figure 4(b), with the reduction in the number of features, particularly the removal of less important ones, the model's loss value significantly decreases, and the rate of decline accelerates. This indicates that feature selection can reduce computational overhead and improve the training efficiency and performance of the model. Most importantly, retaining the most critical features (such as age, loan duration, and property) significantly enhances the model's accuracy and efficiency.

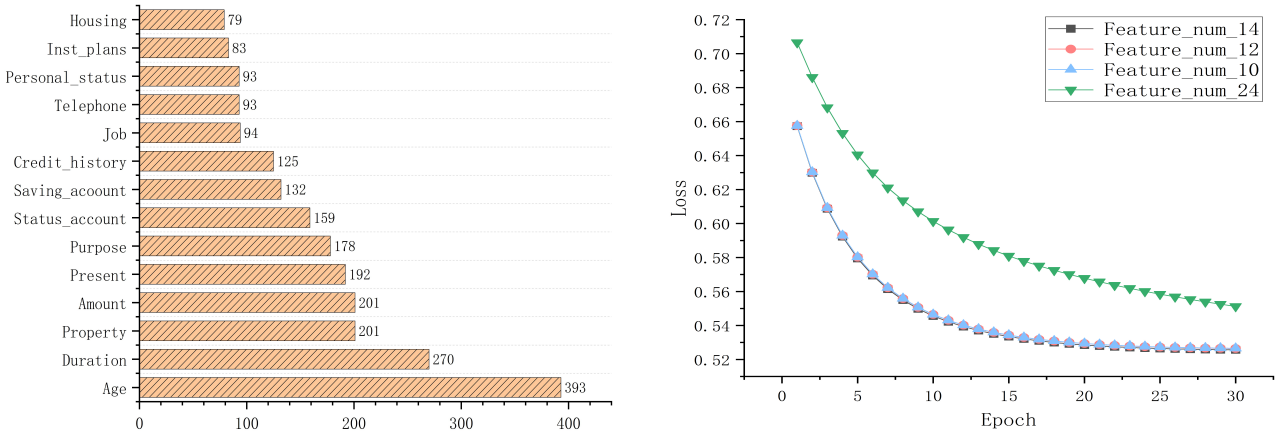
For the experiment, this study randomly split the original dataset vertically based on features into two parts, representing Bank A and Bank B. Each bank holds an equal number of different features. And the hardware and software environments used to implement the prototype system are shown in Table 2.

Table 2
Hardware and software environments.

Hardware	
CPU	Intel 8336C
GPU	GeForce RTX4090(24g)
Memory	256 GB
Software	
OS	Ubuntu 22.04
Golang	v1.90
Python	3.11
Pytorch	2.0.0
cuda	11.8
Cliper	v0.5.0
Hyperledger fabric	2.4.1

7.2. Prototype system

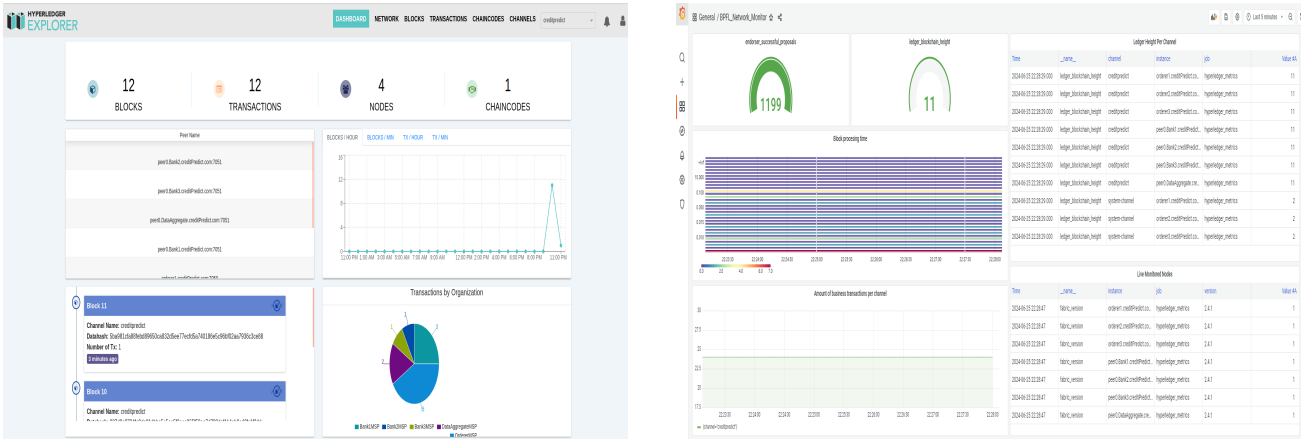
Figure 5(a) shows the activity status of the overall underlying blockchain network by the Hyperledger Expolrer, which includes 12 blocks, 12 transactions, 4 nodes, and 1 chaincode. Figure 5(b) shows the visualization of blockchain node operations monitoring using the Grafana plugin. The specific monitoring metrics include block processing speed,



(a) XGBoost feature selection

(b) Loss performance under different features

Figure 4: XGBoost feature selection and loss performance



(a) Hyperledger Explorer(DASHBOARD)

(b) Operations Monitoring

Figure 5: Blockchain Network Visualization and Operations Monitoring

the number of successful transactions per second, the number of failed transactions, and the total number of transactions in the channel.

7.3. Performance of BC-PSI

To analyse the feasibility of the proposed BC-PSI, the experiment uses Diffie-Hellman-based PSI as a benchmark method and takes the time cost as the evaluation metric to quantify the processing time of completing the co-sample ID comparison among all participants. The experimental results are presented in Figure 6. For different datasets, the execution times of the proposed BC-PSI are slightly better than those of the Diffie-Hellman-based PSI in terms of performing co-sample alignment.

7.4. Performance of BPFL

As previously indicated, the CKKS-based PBFL scheme primarily consists of key generation, plaintext encryption, ciphertext operations, and ciphertext decryption. The BFV[30] and Paillier[31] homomorphic encryption are chosen as the benchmark methods for PBFL. Figure 7(a) illustrates that CKKS significantly outperforms the BFV and Paillier schemes in terms of key generation time. Figures 7(b) and 7(d) show that CKKS and BFV schemes outperformed Paillier schemes for both plaintext encryption and ciphertext decryption at different vector lengths. Compared to the Paillier scheme, Figure 7(c) illustrates that CKKS and BFV ciphertext aggregation costs much less time with vector length increasing. Specifically, CKKS performs slightly better than BFV. Therefore, we can infer that the proposed CKKS-based PBFL methodology is significantly superior in privacy-federated collaborative modelling.

Figure 8 shows the time costs for each stage of CKKS-based PBFL in Section 5.2. As illustrated, adopting the CKKS scheme results in shorter durations for each stage of federated training. Whether it's participant A and B encrypting data, performing aggregation operations on ciphertext data, or participant C decrypting ciphertexts, the time consumption is relatively low. This is mainly because CKKS supports more complex operations beyond simple additions

Table 3
Performance comparison between the CFL and BPFL methods

	Measures			
Model	Accuracy	Precision	Recall	F1 Score
CFL	0.745	0.60	0.45	0.51
BPFL	0.720	0.56	0.42	0.48

and multiplications, and it can control the growth of ciphertexts during these operations. Additionally, CKKS allows for computations on encrypted data with approximate results, which is beneficial for applications involving real or complex numbers. Therefore, CKKS homomorphic encryption is well-suited for scenarios requiring efficient computations on encrypted data, particularly those involving large-scale numerical operations or complex computations.

To further evaluate the discrimination performance of the proposed BPFL method, accuracy, precision, recall, and F1 score are utilized to compare PBFL method (i.e., the real scenario) and CFL method (i.e., ideal scenario, that is, multi-source data can be easily integrated integrated learning by a central node), which respectively measure the overall accuracy of the model, the accuracy of positive class predictions, the detection ability of positive class samples, and the balance between precision and recall. As shown in Table 3, despite the slight difference in the BPFL performance relative to that of the CFL method, the BPFL approaches the CFL method for each performance metric. Considering the privacy requirements in specific scenarios and the computational performance of homomorphic encryption, the discrimination performance of BPFL is still acceptable.

Moreover, Figure 9 shows that the AUC and Loss curves of the BPFL and CFL methods. AUC curve is primarily used for measuring the performance of classification models, especially evaluating binary classifiers. The larger the AUC value, the stronger the model's discriminative ability. Loss curve is taken as the evaluation metric to evaluate the convergence and performance of BPFL and CFL methods during the training process. The lower the loss is, the better the model's performance on the training data. As illustrated, our proposed BPFL method can perform similarly to the ideal scenario (i.e., CFL). The experimental results also further supported the convergence of BPFL in Section 6.

7.5. System performance

Cliper is used for conducting stress testing on ciphertext queries, write, and delete smart contracts to analyze the relationship between ciphertext length and system performance. As shown in Figure 10(a), the performance of ciphertext query and deletion operations are superior to that of write operation. However, as the ciphertext size increases, the throughput of these operations all decrease to varying degrees. For example, when the ciphertext size reaches 1.0KB, the write operation throughput drops to 387; when the ciphertext size reaches 2.2KB, the throughput decreases to 316; and when the ciphertext size reaches 3.4KB, the throughput decreases to 236. This indicates that the throughput of the blockchain decreases as the ciphertext length increases. Figure 10(b) shows the average latency for ciphertext queries, writes, and deletions on the blockchain network. The results indicate that the average latency increases with the ciphertext size increasing.. When the ciphertext size reaches 2KB, the average latency increases from 0.03 seconds to 0.08 seconds, and then stabilizes around 0.07 seconds. This demonstrates that the increase in ciphertext size also leads to an increase in the average latency of blockchain transactions.

However, as can be seen in Figure 10(a), the throughput of the lowest query operation is 300-tps, and even the write operation has 240-tps with an average delay of about 1 second. This means that the platform can process at least 300 query operations or 240 write transactions in one second under a test environment. Therefore, the platform has reasonable performance in querying data and writing data in terms of results. Indeed, to further alleviate the ledger storage and write pressure on blockchain, ciphertext can be compressed before being added to the chain, with decompression operations performed locally at the federated client nodes. This will help improve the performance of the blockchain network.

7.6. Global explanations for BPFL

Without disrupting the existing training, the model incorporates a smart contract for the automatic computation of SHAP values through a blockchain network, aiming to assess the feature contributions of nodes in each training iteration. In Figure 11(a), the x-axis represents the feature values of the samples, while the y-axis represents different features. The position and colour of each point reflect the contributions of diverse dimensions of the data samples to the model output, with the colour depth intuitively indicating the magnitude of the contribution. Some instances of red points appearing on the negative side of the axis may result from the offsetting effects between negative impacts of certain features and positive impacts of others, leading to an overall negative expected value. Figure 11(b) shows the importance ranking of features, where red denotes contributions to positive class samples (class 0), and blue signifies contributions to negative class samples (class 1). The length of the bars in the chart reflects the contribution of each feature to the overall model. Notably, features such as "purpose," "status_account," and "duration" had significant influences on the model.

Acknowledgment

This research was supported by the National Natural Science Foundation of China (No.71850012, 71790593), National Social Science Fund of China (No. 19AZD014), Major Special Projects of the Department of Science and Technology of Hunan Province (No. 2018GK1020).

References

[1] Giudici, P., 2018. Fintech risk management: A research challenge for artificial intelligence in finance. Frontiers in Artificial Intelligence and Applications 1, 1–6.

[2] Yang, F., Abedin, M.Z., Hajek, P., 2024. An explainable federated learning and blockchain-based secure credit modeling method. European Journal of Operational Research 317, 449–467.

[3] Lan, Q., Ma, J., Yan, Z., Li, G., 2022. Utility-preserving differentially private skyline query. Expert Systems with Applications 187, 115871.

[4] Qiao, Y., Cheng, L., Lan, Q., Wang, Y., 2023a. Lechain:linear credit evaluation system based on hyperledger fabric blockchain, in: Proceedings of the 2022 5th International Conference on Blockchain Technology and Applications, p. 164169.

[5] Qiao, Y., Lan, Q., Wang, Y., Jia, S., Kuang, X., Yang, Z., Ma, C., 2023b. Pevachain: Privacy-preserving ridge regression-based credit evaluation system using hyperledger fabric blockchain. Expert Systems with Applications 223, 119844.

[6] Hassija, V., Bansal, G., Chamola, V., Kumar, N., Guizani, M., 2020. Secure lending: Blockchain and prospect theory-based decentralized credit scoring model. IEEE Transactions on Network Science and Engineering 7, 2566–2575.

[7] Yang, F., Qiao, Y., Qi, Y., Bo, J., Wang, X., 2022. Bacs: Blockchain and automl-based technology for efficient credit scoring classification. Annals of Operations Research , 1–21.

[8] Qiao, Y., Lan, Q., Zhou, Z., Ma, C., 2022. Privacy-preserving credit evaluation system based on blockchain. Expert Systems With Applications 188, 188.

[9] Yang, Q., Liu, Y., Chen, T., Tong, Y., 2019. Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology 10, 1–19.

[10] Abdulrahman, S., Tout, H., Ould-Slimane, H., Mourad, A., Talhi, C., Guizani, M., 2021. A survey on federated learning: The journey from centralized to distributed on-site learning and beyond. IEEE Internet of Things Journal 8, 5476–5497.

[11] Mothukuri, V., Parizi, R.M., Pouriyeh, S., Huang, Y., Dehghantanha, A., Srivastava, G., 2021. A survey on security and privacy of federated learning. Future Generation Computer Systems 115, 619–640.

[12] Li, T., Sahu, A.K., Talwalkar, A., Smith, V., 2020. Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine 37, 50–60.

[13] Kim, H., Park, J., Bennis, M., Kim, S.L., 2019. Blockchained on-device federated learning. IEEE Communications Letters 24, 1279–1283.

[14] Yang, F., Qiao, Y., Abedin, M.Z., Huang, C., 2022. Privacy-preserved credit data sharing integrating blockchain and federated learning for industrial 4.0. IEEE Transactions on Industrial Informatics 18, 8755–8764.

[15] Rieyan, S.A., News, M.R.K., Rahman, A.M., Khan, S.A., Zaarif, S.T.J., Alam, M.G.R., Hassan, M.M., Ianni, M., Fortino, G., 2024. An advanced data fabric architecture leveraging homomorphic encryption and federated learning. Information Fusion 102, 102004.

[16] Zhang, J., Tan, R., Su, C., Si, W., 2020. Design and application of a personal credit information sharing platform based on consortium blockchain. Journal of Information Security and Applications 55, 102659.

[17] Chakraborty, S., Aich, S., Seong, S.J., Kim, H.C., 2019. A blockchain based credit analysis framework for efficient financial systems, in: 2019 21st International Conference on Advanced Communication Technology (ICACT), pp. 56–60.

[18] Li, D., Han, D., Crespi, N., Minerva, R., Li, K.C., 2023. A blockchain-based secure storage and access control scheme for supply chain finance. Journal of Supercomputing 79, 109–138.

[19] Zheng, K., Zheng, L.J., Gauthier, J., Zhou, L., Xu, Y., Behl, A., Zhang, J.Z., 2022. Blockchain technology for enterprise credit information sharing in supply chain finance. Journal of Innovation & Knowledge 7, 100256.

[20] Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., Gao, Y., 2021. A survey on federated learning. Knowledge-Based Systems 216, 106775.

[21] Banabilah, S., Aloqaily, M., Alsayed, E., Malik, N., Jararweh, Y., 2022. Federated learning review: Fundamentals, enabling technologies, and future applications. Information Processing & Management 59, 103061.

[22] Zhu, J., Cao, J., Saxena, D., Jiang, S., Ferradi, H., 2023. Blockchain-empowered federated learning: Challenges, solutions, and future directions. ACM Computing Surveys 55, 1–31.

[23] Zhang, H., Fan, W., Wang, J., 2024. Bidirectional utilization of blockchain and privacy computing: Issues, progress, and challenges, p. 103795.

[24] Rückel, T., Sedlmeir, J., Hofmann, P., 2022. Fairness, integrity, and privacy in a scalable blockchain-based federated learning system. Computer Networks 202, 108621.

[25] Zhang, J., Ye, A., Chen, J., Zhang, Y., Yang, W., 2023. Csfl: Cooperative security aware federated learning model using the blockchain. The Computer Journal 67, 1298–1308.

[26] Jia, B., Zhang, X., Liu, J., Zhang, Y., Huang, K., Liang, Y., 2022. Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in iiot. IEEE Transactions on Industrial Informatics 18, 4049–4058.

[27] Jovanovic, Z., Hou, Z., Biswas, K., Muthukkumarasamy, V., 2024. Robust integration of blockchain and explainable federated learning for automated credit scoring. Computer Networks 243, 110303.

[28] Dumitrescu, E., Hué, S., Hurlin, C., Tokpavi, S., 2022. Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. European Journal of Operational Research 297, 1178–1192.

[29] Gunnarsson, B.R., vanden Broucke, S., Baesens, B., Óskarsdóttir, M., Lemahieu, W., 2021. Deep learning for credit scoring: Do or dont? European Journal of Operational Research 295, 292–305.

[30] Brakerski, Z., Gentry, C., Vaikuntanathan, V., 2014. (leveled) fully homomorphic encryption without bootstrapping. ACM Transactions on Computation Theory 6.

[31] Paillier, P., 1999. Public-key cryptosystems based on composite degree residuosity classes, in: Advances in Cryptology– EUROCRYPT '99, pp. 223–238.

[32] Cheon, J.H., Kim, A., Kim, M., Song, Y., 2017. Homomorphic encryption for arithmetic of approximate numbers, in: Advances in Cryptology – ASIACRYPT 2017, pp. 409–437.

[33] Lin, X., Li, Y., Xie, X., Ding, Y., Wu, X., Ge, C., 2024. Sf-cabd: Secure byzantine fault tolerance federated learning on non-iid data. Knowledge-Based Systems 296, 111851.

[34] Yang, F., Qiao, Y., Huang, C., Wang, S., Wang, X., 2021. An automatic credit scoring strategy (acss) using memetic evolutionary algorithm and neural architecture search. Applied Soft Computing 113, 107871.

[35] Patel, S.B., Bhattacharya, P., Tanwar, S., Kumar, N., 2021. Kirti: A blockchain-based credit recommender system for financial institutions. IEEE Transactions on Network Science and Engineering 8, 1044–1054.

[36] Khalili, N., Rastegar, M.A., 2023. Optimal cost-sensitive credit scoring using a new hybrid performance metric. Expert Systems with Applications 213, 119232.

[37] Lyn Thomas, J.C., Edelman, D., 2017. Common Methods for Building Scorecards. pp. 25–47.

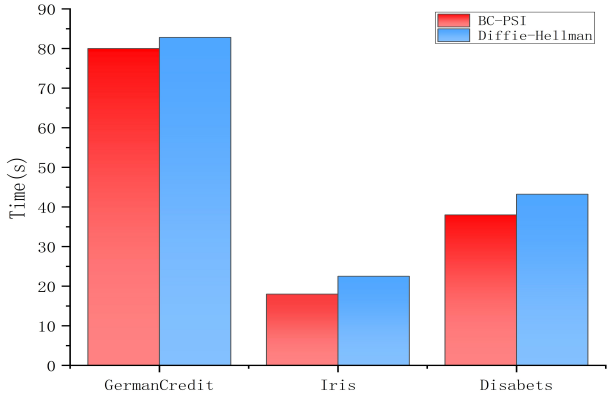
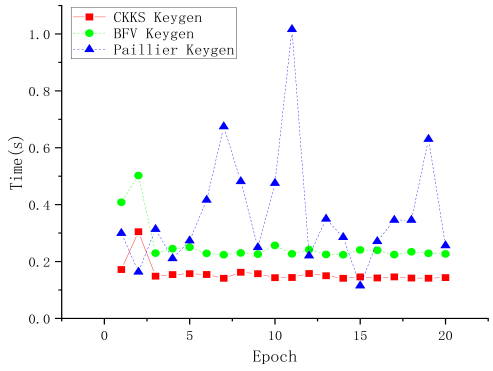
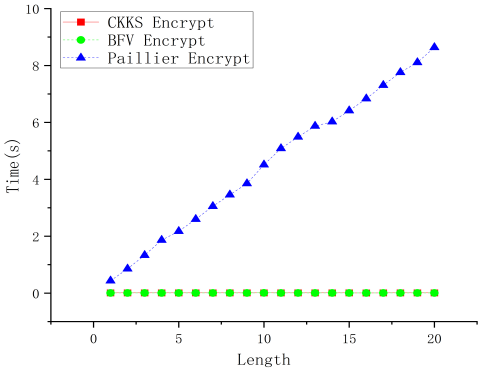


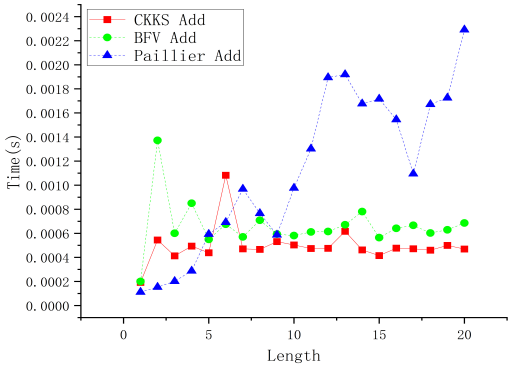
Figure 6: Performance comparison between BC-PSI and Diffie-Hellman-based PSI.



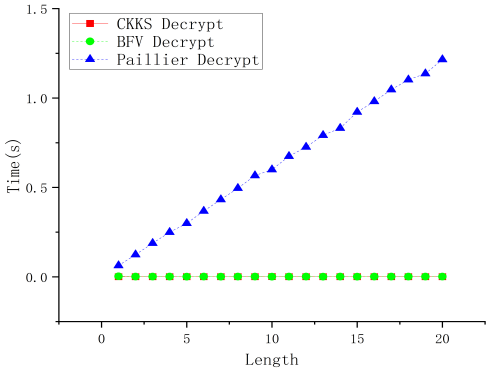
(a) Key-generation costs under different homomorphic encryption



(b) Encryption costs under different homomorphic encryption



(c) Aggregation costs under different homomorphic encryption



(d) Decryption costs under different homomorphic encryption

Figure 7: Performance comparison between CKKS, BFV and Paillier homomorphic encryption.

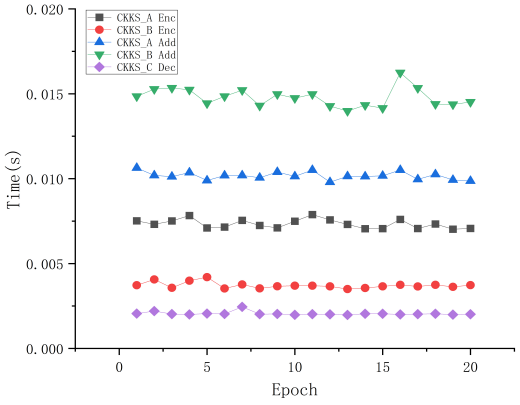
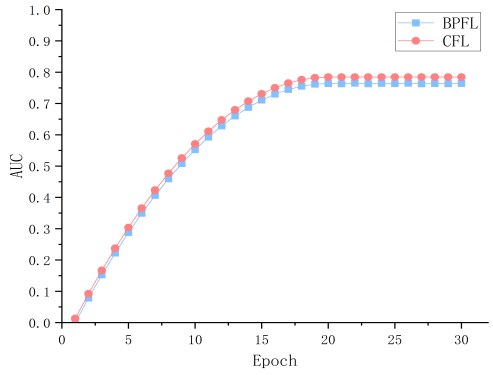
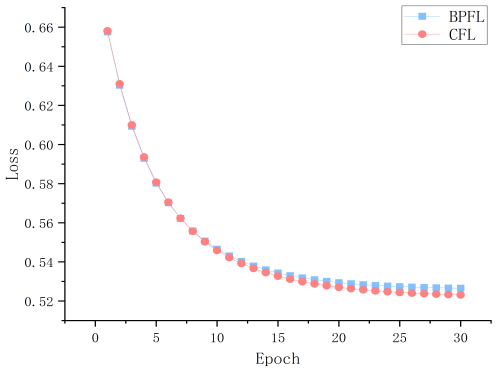


Figure 8: Time costs of CKKS-based BPFL.

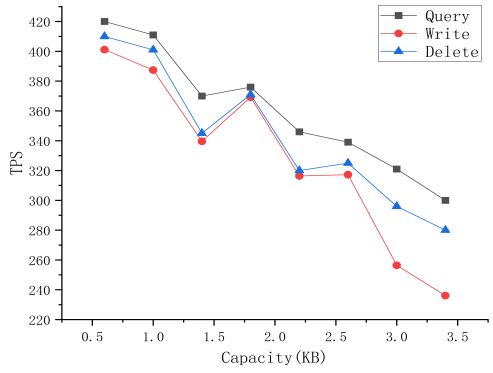


(a) The AUC curves of the BPFL and CFL methods

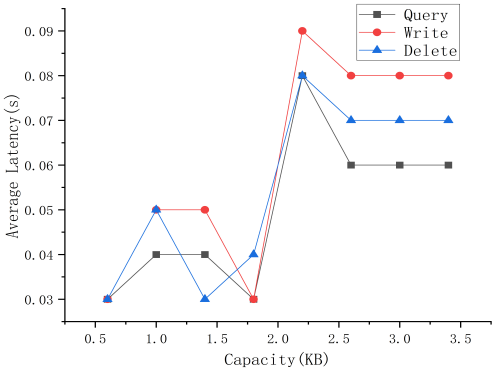


(b) The Precision-Recall of the BPFL and CFL methods

Figure 9: AUC and Loss curves

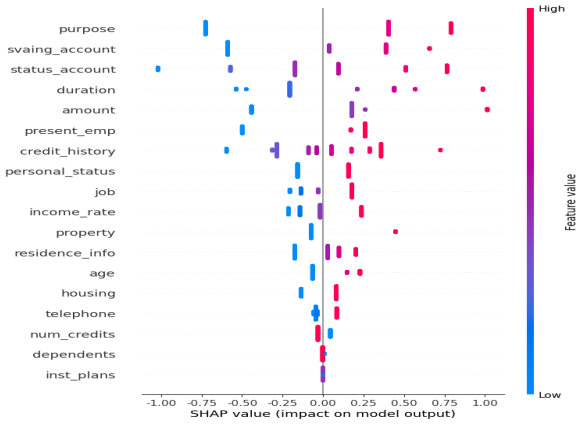


(a) Throughput for Query-Write-Delete

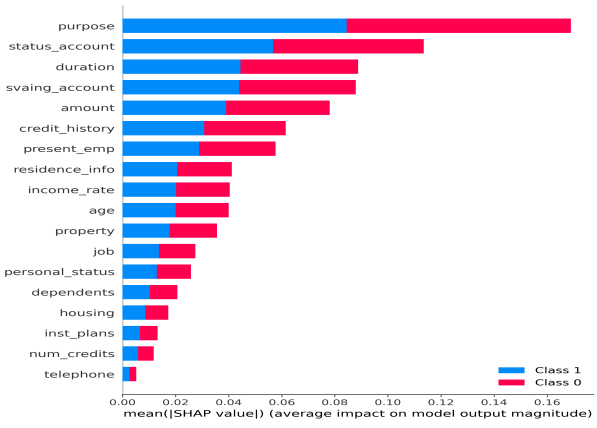


(b) Average Latencey for Query-Write-Delete

Figure 10: Caliper Pressure test



(a) Total SHAP values



(b) Average impact on model output magnitude

Figure 11: Global model feature contribution