An Enhanced Data-sharing Method Based on Block chain And Random Forest

Qidi Zheng, Haiyan Kang

Abstract— **Federated learning can break the data silos while ensuring the participants' control of the data, thus effectively protecting the data privacy of intelligent edge nodes. Existing federated learning techniques usually upload the intermediate parameters of model training to the parameter server to realize model aggregation, but there are two problems in this process: on the one hand, the federated learning process generates a large amount of communication overhead due to multiple learning. On the other hand, malicious nodes may upload false parameters or low-quality models, thus affecting the aggregation process and model quality. To address the above two problems, an enhanced data sharing method based on blockchain and random forest is proposed. The method consists of three parts, which are the tree algorithm based on CART tree, the federated forest model based on CART tree, and the learning algorithm based on CART tree and Bagging. Firstly, a blockchain is constructed and the data is stored on the blockchain. Secondly, the data is distributed to the nodes involved in learning, each node stores the corresponding data, and multiple nodes are constructed into a tree. Then the central server is responsible for collecting all the trees constructed by the nodes to generate a complete tree. Finally, the model is used to make predictions. Experiments using real datasets show that this method can not only enhance the mutual trust between nodes participating in federated learning, but also reduce communication overhead, and finally obtain a trusted federated learning model with enhanced privacy protection.**

Index Terms—**Privacy protection; Blockchain; Federated learning; Random forest.**

I. INTRODUCTION

In recent years, with the significant improvement of computing power, the emergence of relevant innovative machine learning algorithms, the significant improvement of computing power and the rapid development of big data, artificial intelligence technology has reached a new height. However, a significant amount of computation determines the success of a model. The development of big data is accompanied by some success stories. With the further development of big Data, privacy protection has become a global trend today. At present, the national level is emphasizing the privacy of citizens. This brings great challenges to the field of artificial intelligence[1]. With the premise that data privacy, security and regulation need to be satisfied, how to design and train a good machine learning model so that AI systems can access the required data is crucial. One possible solution is federated learning.

However, the traditional federated learning technique does not give particularly satisfactory results when dealing with data that is not independently and identically distributed. After a number of experiments show that when the data of some nodes have particularly obvious differences in distribution, the accuracy of the trained model will become very poor, and can not be applied. In addition, the data of each node in the actual generation process, there is a great possibility that other nodes or their own environment will be affected. At the same time, most of the data of each node is non-independently and identically distributed, which also makes the federated learning in the practical application of some difficulties, that is, how to reduce the non-independent and equally distributed data to the training accuracy of users[2].

Meanwhile, existing federated learning mainly uses a parameter server to generate or update the global model parameters, which is a typical centralized architecture. This also has some problems with federated learning. The specific problems are as follows: (1) In the process of learning the clients involved in learning need to pass through the central server in terms of synchronizing the model at the end of each training session, and in this process there needs to be a large degree of communication overhead. (2) Federated learning requires the participation of the central server, and it is difficult to determine whether the central server is safe during the learning process. (3) Most of the commonly used data now has the characteristics of high overlap of data examples and less feature overlap. If such data cannot be used, there will be a large loss. (4) The quality of federated learning models trained locally by users varies greatly, making it difficult to guarantee the quality of the final global model after aggregation.

Blockchain is a data ledger technology maintained by a peer-to-peer network. It establishes a new decentralized trust scenario in the absence of a trusted third party, which can solve the security problem of central servers. Through the consensus protocols, data ledgers on the blockchain network have a tamper-proof, non-repudiation and non-forgeable way of conducting business. There have been many scholars who have studied in detail the application of blockchain in the context of IoT and have considered very important factors such as security issues, efficiency issues, and consensus algorithms in selecting applications. To sum up, blockchain can be used to solve the centralized services with high security risks or high service charges.

To solve the above deficiencies, an enhanced data sharing method based on blockchain and random forest is proposed, and on this basis, a federated forest model based on

Manuscript received March 17, 2026
Qidi Zheng, School of Computer Science, Beijing Information Science and Technology University, Beijing, China
Haiyan Kang, School of Computer Science, Beijing Information Science and Technology University, Beijing, China

blockchain and random forest (Federated Forest, FF) is proposed. Based on the proposed model, a secure machine learning system with a distributed structure is constructed, which can solve the above shortcomings. The main contributions of this paper are as follows.

(1) For the security problem of the central server, a tree algorithm (Tree Algorithm based on CART, TACART) based on CART tree is proposed to improve the security of the central server. This paper reconstructs the construction algorithm, applies the encryption method and establishes the third-party trusted server, so that the data privacy is fully protected, the content and quantity of information exchange are limited to the minimum, and each participant is blind to others' data, thus guaranteeing the privacy of the data.

(2) In view of the problem of low data utilization in the federated learning, the federated forest model (Federated Forest Model base on CART, DTMCART) based on CART tree is proposed, which improves the scalability of the federated learning. The model supports both classification and regression tasks, and it has high utility and scalability in practice. Experiments on real-world datasets demonstrate high accuracy, efficiency and robustness.

(3) According to the model quality problem in the process of federated learning, a learning algorithm (Leaning Algorithm based on CART and Bagging, LACARTB) based on CART tree and Bagging is proposed and designed to ensure the accuracy of federated learning. The mechanism is based on the CART tree and the Bagging, and is suitable for longitudinal federated learning. This mechanism is experimentally demonstrated to achieve the same accuracy as a non-federated learning method for pooling datasets into the same location.

II. Related Work1

This section will introduce the studies on related areas in three aspects. Firstly, in order to ensure that the value of data can be highly utilized during the flow of large-scale data, relevant machine learning techniques, including federated learning, are introduced. Secondly, in order to solve the problem of data privacy leakage during the training process of machine learning models, relevant privacy-preserving methods, including homomorphic encryption, are introduced. Finally, relevant research on blockchain is introduced in terms of the distributed as well as trustworthy environment problem of federated learning.

A. Data Sharing Based on Federated Learning Technology

Federated Learning (FL) enables multiple decentralized clients to collaboratively train machine learning models while retaining data locally, thereby mitigating privacy risks and regulatory concerns. However, FL faces several fundamental challenges, particularly in terms of high communication overhead and system heterogeneity. To address these issues, recent efforts have focused on communication-efficient optimization techniques. For instance, FedBAT introduces learnable binarization techniques to reduce parameter sizes while maintaining model performance under non-IID data distributions [3]. Similarly, AdapComFL dynamically adjusts compression ratios based on bandwidth availability, achieving communication cost reductions without compromising accuracy [4].

Advanced frameworks such as FedACG incorporate global momentum to accelerate local training convergence and reduce divergence across client models [5]. In another line of work, FedALS and FedScalar optimize aggregation frequency and scalar-level parameter transmission respectively, reducing communication burdens by up to 80% [6-7]. Moreover, hardware-level optimization such as FedRDMA leverages Remote Direct Memory Access (RDMA) to enable low-latency cross-device communication, accelerating federated learning by over 3.8 times [8].

Despite these advances, federated learning systems remain susceptible to instability arising from device heterogeneity, straggler effects, and non-uniform participation rates. These challenges necessitate further research into adaptive learning schedules, incentive-compatible participation schemes, and fairness-aware aggregation mechanisms.

B. Data Sharing Based on Homomorphic Encryption Technology

To strengthen privacy protection during model updates and aggregation, Homomorphic Encryption (HE) has been integrated into FL to support secure computation over encrypted data. Conventional Fully Homomorphic Encryption (FHE) schemes, although theoretically secure, often suffer from computational inefficiencies that limit their practical deployment. Recent work has focused on improving the efficiency of HE-based FL systems. For example, QuanCrypt-FL employs quantized encryption and model pruning, yielding a 3times acceleration in training and up to 16times improvements in encryption/decryption time [9].

Selective encryption approaches such as MASKCRYPT encrypt only sensitive parts of model updates to balance communication and security overhead [10]. FedNIC offloads homomorphic operations to SmartNIC hardware, alleviating computational burdens on client devices while preserving privacy [11]. Furthermore, FedPHE combines packed encryption with sketch-based client selection, achieving 1.85–4.4times training speed-up and communication reductions of up to 22.6times compared to conventional methods [12].

While these methods significantly improve HE efficiency, they still require careful handling of key distribution, ciphertext accumulation, and computation latency, especially in resource-constrained and real-time learning environments. Recent theoretical advances also explore the integration of quantum-compatible encryption schemes to enhance long-term security guarantees in FL [13].

C. Data Sharing Based on Blockchain

Blockchain technologies offer transparent, tamper-proof, and decentralized infrastructures, making them well-suited to address the trust and traceability challenges in FL. Smart contracts and consensus protocols can be employed to automate client registration, update verification, and contribution-based incentive distribution. For example, the BlockFL framework leverages on-chain consistency checking and weighted aggregation to ensure fairness and transparency among participants [14].

Systems such as DeepChain further extend this paradigm

by combining blockchain with homomorphic encryption and verifiable decryption protocols. These systems utilize digital tokens (e.g., DeepCoin) to encourage honest participation while punishing malicious actors. Despite these advancements, existing blockchain-enhanced FL frameworks often suffer from high consensus latency and energy consumption, especially when using Proof-of-Work (PoW) or Byzantine Fault Tolerant (BFT) protocols.

Emerging research points to the need for lightweight consensus mechanisms (e.g., Proof-of-Stake, DPoS) and reputation-aware client selection protocols that reduce communication complexity while preserving the trustworthiness of the learning environment [15]. Additionally, cross-chain and off-chain computation strategies are being explored to improve scalability and interoperability of blockchain-based FL systems

In summary, the convergence of federated learning, homomorphic encryption, and blockchain lays the foundation for a new generation of secure, collaborative, and decentralized learning systems. While substantial progress has been made in each domain, their integration into a unified and scalable framework remains an open research challenge, particularly in balancing privacy, performance, and trust.

III. Related Knowledge

A. Federated Learning

Federated Learning (FL) enables collaborative model training across multiple clients without sharing raw data, preserving privacy and complying with data regulations [16-17]. Challenges such as communication overhead and statistical heterogeneity have driven research into communication-efficient algorithms, including learnable binarization and adaptive compression techniques [18-19]. Hardware-accelerated communication protocols like RDMA further improve training efficiency in large-scale FL. Additionally, federated large language models (FedLLM) focus on efficient fine-tuning and prompt learning for personalization [20].

B. Homomorphic Encryption

Homomorphic Encryption (HE) supports computation on encrypted data, allowing privacy-preserving aggregation in FL. Fully Homomorphic Encryption (FHE) offers strong security but with high computational costs. Recent approaches optimize this trade-off by employing quantized HE schemes and selective encryption of sensitive model components [21]. Multi-key HE (MKHE) schemes allow secure computation with distinct keys across parties, suitable for both horizontal and vertical FL scenarios. Hardware-accelerated implementations further enhance HE efficiency [22].

C. Blockchain

Blockchain provides decentralized, tamper-proof ledgers, enhancing trust, auditability, and incentive mechanisms in FL [23]. Smart contracts are used to authenticate participants and manage contribution rewards [24]. However, blockchain integration incurs overheads in communication and consensus latency. Lightweight consensus protocols and hybrid architectures have been proposed to improve scalability, especially in IoT and edge computing environments.

D. Random Forest

Random Forest (RF) is an ensemble of decision trees, valued for accuracy, robustness, and interpretability [25]. Recent advances in RF include feature selection methods like Regularized Random Forest (RRF) and Variable Selection Using Random Forests (VSURF), improving generalization. Techniques addressing class imbalance, such as Improved Balanced Random Forest (iBRF), enhance performance metrics like MCC and F1-score. Pruning techniques reduce model complexity while preserving predictive power.

IV. Design of Data Sharing Method Based on Blockchain and Random Forest

In view of the difficulty of data training, the training efficiency and the security of the central server caused by the uneven distribution of data in traditional machine learning, in this paper, with the blockchain as the carrier, a Data Share Method Based on Blockchain and Random Forest (DS-BRF) are proposed. And as a result, a federated forest model based on CART trees and bagging is proposed, which can handle classification problems and regression problems simultaneously. At the same time, a Learning Algorithm Based on CART Tree and Bagging (LACARTB) is designed in the model, and a Tree Algorithm Based on CART Tree (TACART) is proposed. The model framework is shown in Figure 1. Specifically, the blue node in the graph is the tree node that stores the split threshold, while the node without marked color is ordinary nodes, which do not store any information, and only represents the tree structure in the block, with no practical significance.

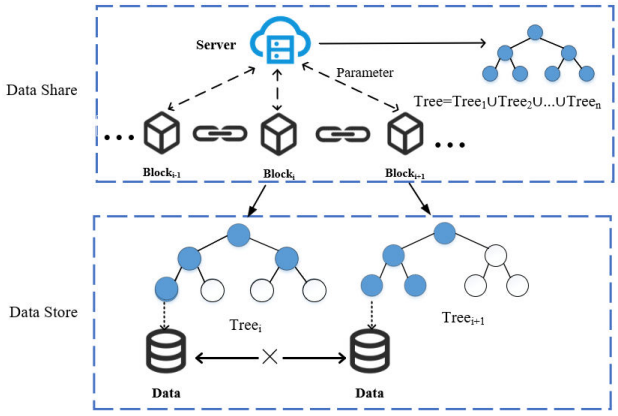


Fig1. System architecture diagram

The core idea of DS-BRF is to store data on the tree node, so as to solve the problem of data in the training process. The DS-BRF method consists of two parts, namely, the parameter exchange part and the data storage part. The parameter exchange part consists of the server and the blockchain nodes. The blockchain nodes store the data of the tree nodes, while the server stores the data of all the tree nodes. Each blockchain node (client) is responsible for training its own data and uploading the relevant parameters to the server. The parameters include number, sub rate and depth. On the one hand, the server integrates the relevant parameters to obtain the final model. On the other hand, the server is responsible for collecting information from all nodes and storing all nodes as a CART tree. The final server coordinates the client to complete the model prediction. A detailed description of the federated forest model construction, its structural storage, the federated forest model

prediction, and the DS-BRF method analysis in the DS-BRF method is provided below.

The main symbols used in this paper are shown in Table 1.

Table 1. Main symbols in the algorithm

symbol	*definition*
M	Area number
D_a	Dataset owned by the client a
D	All datasets
F_a	The sample space of the dataset D_a
F	The sample space of D
L	label
T_a	Partial decision tree stored on the a-th client

A. Federated Forest Model Structure

Any tree is created by the learners involved in federated learning. The tree structure is stored on the server and each client, while each tree only stores segmentation information about their own characteristics. In algorithm 1 and in algorithm 2 it describes how the server coordinates the modeling process.

The server first randomly selects a subset of features and samples from the entire dataset, and then notifies each customer of the selected features and sample ID. For the selected feature, the server notifies each customer privately. All clients enter the split state and do not stop splitting if the termination condition is not triggered. The best split features of the current tree node were selected by comparing improved improvements. The specific approach is as follows: Firstly, each client a finds the local optimal segmentation feature Fa. The server then collects all the local optimal features for integration to find the globally optimal features. Secondly, the server notifies the clients that provide the global best features. The corresponding client splits the samples and sends the results of the data division (sample IDs that fall into the left subtree left_tree and the right subtree right_tree) to the server for distribution. For the current tree node, only the client providing the best split will save the corresponding information of the split. Other customers only know that the selected function is not their own contribution, and they do not know the relevant information such as the threshold and segmentation features. Finally, it recursively creates the subtree and returns the current tree node. During the tree creation, if the subtree node is created successfully created, the parent node does not need to save the sample ID of the subtree. Otherwise, the modeling can be easily resumed from the breakpoint. The client-side construction forest algorithm is shown in algorithm 1, and the server construction forest algorithm is shown in algorithm 2.

Algorithm 1: Model construction algorithm of DS-BRF (part of blockchain nodes) TACART (D, F, L)

Input: Dataset D of the client aa, Local feature Fa, Encryption label La.

Output: Subfederal forest model on client a.

1. Create an empty tree node
2. The if satisfies the pre-pruning condition then
3. Set the current node to be the leaf node;
4. return Leaf node
5. if Fa`non-empty then
6. Calculate the improvement value p of any f and find the local maximum value pmax;
7. Record the locally optimal segmentation feature f* and the segmentation threshold;
8. Send the encrypted pmax to the server.
9. if split is true then
10. Splitting the sample
11. Send the sample index of the left_tree and right_tree to the server;
12. else
13. The server receives the sample index of the left_tree, right_tree.
14. left_tree ← TreeCreate(D`a_left,F`a,yleft) // Left _ tree is the left subtree.
15. right_tree ← TreeCreate(D`a_right,F`a,yright) // Right _ tree is the right subtree.
16. if is _ selected then // tree node is selected
17. Save f and upload the threshold to the tree_node
18. Save the subtree to the tree_node
19. return tree_node
20. return partial federal forest model on client a

Algorithm 2: Model constructor of DS-BRF (server part) TACART (D, F, y)

Input: Index D, encryption feature F, encryption feature y.

Output: The complete federated forest.

Function TACART (D, F, y)

1. Receive and related information from all clients
2. Make j=max
3. The server broadcasts the split index received from the client j
4. left_tree ← TreeCreate(Dleft, F, yleft)
5. right_tree ← TreeCreate(Dright, F, yright)
6. Save the subtree and split information to the tree_node
7. return tree_node
8. Add the current trees to the forest
9. Return to the full federated forest

B. Federal Forest Model Storage

Since the forest is constructed by all clients, the structure of each tree above each client is the same. However, for a given tree node, the client may or may not store the details.

Only the master server can store the complete model. However, for any tree node, the client will store the corresponding resolution threshold only if the resolution function is provided. Otherwise, the client does not store anything at the current node and only retains the node structure. In the text, T is used to represent the complete tree nodes, that is, the nodes stored on the server, and the tree nodes without the complete details stored by the a-th client is represented as T_a. Since the structure of each tree is consistent, the intersection of all Ta is set to L in the text, where L is the set of leaf nodes, then the complete tree T is the union of all subtrees.

C. Federal Forest Model Prediction

In vertical federated learning, classical prediction methods involve multiple communications between the server and the client. Even only one sample requires multiple communications. When the number of trees, the depth of maximum trees and the sample size are sufficient, the

predicted communication requirements will become a serious burden. To solve this problem, a novel prediction method is designed. The method utilizes a distributed model storage strategy and only requires a round of collective communication on this tree or even the entire forest. The client prediction algorithm is shown in algorithm 3, and the server coordination client implementation prediction algorithm is shown in algorithm 4. The detailed steps of the algorithm prediction are as follows.

In the first step, each client predicts the sample using its locally stored model. For the a-th client T_a, each sample enters T_a from the root node and finally enters a few leaf nodes through the binary tree. When the sample passes through each node, it is necessary to judge whether the node stores the split information. If the model stores split information at that node, it is determined that the sample entered the left subtree left_tree or the right subtree right_tree by checking the split threshold. If the model does not split at this node, the sample enters both left_tree and right_tree.

In the second step, one recursively confirms the path of the leaf nodes until each sample falls into one or several leaf nodes. When this process is complete, five samples are kept for each leaf node of the tree T_a on client a. Denote by Sa the samples that fall into leaf node a of the tree model T_a, where a is a subset of L and L is the set of leaf nodes of the tree T_a.

In the third step, for each leaf a∈L, the server will take the intersection on $\{S_a\}M_a=1$, resulting in S_a. A formal proposition is given here for the proposed method, so it can be defined mathematically.

Definition. For the sample S falling into one or more leaves on the tree T_a, then for any leaf A of the complete tree T, the sample ID S_l in the leaf A can be obtained from the intersection, that is, $S_l = S_1 l S_2 l ... \{S_i^l\}_{i=1}^M \bigcap\bigcap\bigcap S_m l$.

Algorithm 3: Function LACARTB (T_a, D_a^{test}, F_a): The blockchain part

Input: Partial federated forest model T_a saved on the a-th client, coded features F_a on the a-th client, test set D_a^{test} on the a-th client.

Output: T_a Upper sample ID of leaf l, where l∈L.

Function TACART (D, F, y)
1. The if leaf node then
2. return Sample ID, leaf labels, and S^l_a
// S^l_a for the sample of the leaf node a that fell into the tree model
3. else
4. if T_a saves the split information of the current node, then
5. Samples were split into subtrees
6. left_tree ← TreePredict(T_{a_left}, F_a, $D_{a,left}^{test}$)
// left _ tree is the left subtree
7. right_tree←TreePredict(T_{a_right})
// right _ tree is the right subtree
8. else
9. left_tree ← TreePredict(T_{a_left}, F_a, $D_{a\_left}^{test}$)
10. right_tree ← TreePredict(T_{a_right}, F_a, T_{a_right})
11. return left_tree and right_tree

Algorithm 4: LACARTB--The Server section

Input: Test set D_{test} of the sample ID, S.

Output: Projected value of federal forests.

1. while TreePredict is the true do
2. collect $\{S_1, S_2, \ldots, S_n, \ldots\}$
3. gain $\{S_1, S_2, \ldots, S_n, \ldots\}$ among $S^n = S_1^n \cap S_2^n \cap \ldots \cap S_m^n$
4. return S_m^l the label of the leaf l of the middle sample, where l∈n
5. Calculate the predicted value of the forest by voting on the results of the tree
6. Return the final predicted value

D. DS-BRF Method Analysis

1) Privacy Analysis

Suppose At is the attack initiated by the attacker in order to obtain the relevant privacy information of the t-th user, X_t is the number of malicious users in layer t, N_t is the number of users in layer t, and N is the total number of users. The relevant information in the DS-BRF algorithm proposed in this paper is integrated from the lower layer and then uploaded to the upper layer. This also indicates that an attacker below layer t-1 cannot receive any relevant information in layer t, while users above layer t+1 can only obtain their model s of local integration. According to paper[26], when malicious users in layer t are more than half of the total number of users in layer t, malicious users can obtain the privacy information of users in layer t-1, that is, $P[A_t] = P[X_t > N_t/2]$. To calculate $P[A_t]$, it is assumed that the number of malicious users in layer t is no more than T, while Xi follows a hypergeometric distribution of parameters N, T, N_i, with the probability of defining X_t as

$$P(X_t \geq \frac{N_t}{2}) \leq \exp(-c_T N_t) \tag{1}$$

Where $C_T = D(0.5 \| T/N)$. When $T \leq (0.5-\varepsilon)N$ and $\varepsilon > 0$, is then always a positive constant. Therefore,

$$\begin{aligned} &P(privacy\ \ leakage) \\ &= P(\underset{l\in[N/N_t]}{U} E_l) \\ &\leq \sum_{l\in[N/N_t]} P(E_t) \end{aligned} \tag{2}$$

$$\leq \frac{N}{N_t}\exp(-C_T N_t) \tag{3}$$

$$= X_{privacy} \tag{4}$$

Where $\frac{N}{N_t}$ is the number of subgroups and $X_{privacy}$ is the privacy risk. Taking the limit on $X_{privacy}$ yields:

$$\begin{aligned} &\lim_{N\to\infty} X_{privacy} \\ &= \exp(\lim_{N\to\infty} \log X_{privacy}) \\ &= \exp(\lim(\log N - \log N_t - C_T N_t)) \\ &= \exp(-\infty) = 0 \end{aligned} \tag{5}$$

To sum up, when N is large enough, the DS-BRF method proposed in this paper can prevent users' privacy information from being obtained by other malicious nodes, and the risk of privacy leakage is 0.

2) Algorithm Complexity Analysis

Let the n trees be constructed, and each tree has m nodes. The number of iterations for the model construction algorithm for DS-BRF is t and k for the model prediction algorithm. The implementation of the DS-BRF algorithm has the following two stages.

(1) Model construction algorithm. In the model construction algorithm, as multiple trees are constructed and

the time complexity of constructing a tree is O(mlog(m)). The algorithmic complexity of constructing a tree is O(nm(log(m))). Therefore, the time complexity of the model construction algorithm is O(tnm(log(m))).

(2) Model prediction algorithm. Where the time complexity of constructing the binary tree is O(mlog(m)) and the time complexity of the prediction algorithm in the server part is O(mn), the time complexity of the model prediction algorithm is O(k(mn+mlog(m))).

Therefore, the time complexity of DS-BRF algorithm is O(tnm(log(m)) + k(mn+mlog(m))). It has lower time complexity compared to the traditional random forest algorithm.

V. Experiment and Analysis

A. Experimental Environment and Dataset

Experimental environment: This experiment is conducted under the windows 11 system, the hardware configuration is intel i5-6300HQ CPU, 16GB RAM, and the operating environment is Python 3.8.

Experimental data: Three publicly available datasets were used in this experiment: wine data set, Iris data set, and tea data set. Relevant information for the dataset is shown in Table 2. The purpose of the experiment is to verify the performance of the proposed method (DS-BRF) with the non-federated method. Two experiments are conducted, including multi-user comparison experiment and prediction efficiency experiment. The ratio of test set to training set used in the experiment is 1:3. This experiment is performed in a trusted environment.

Table 2 Dataset information

dataset	big or small	feature
wine	12kb	15
Iris	5kb	5
tea	5kb	15

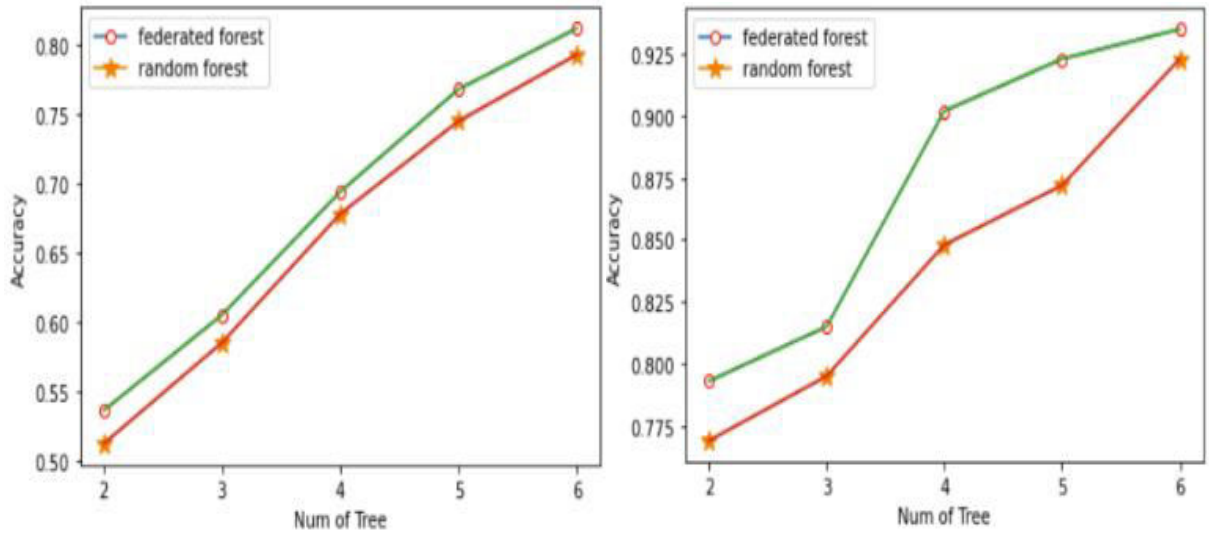
B. Experimental Procedure and Analysis of Experimental Results

1) Test 1: Multi-user Comparison Experiment (Model Prediction Accuracy Experiment)

In this paper, we compare the accuracy aspects of federated forests and commonly used random forests. Tested on the wine and Iris datasets to verify whether the federated forest can reasonably improve the accuracy. To simulate data from multiple users, set up each user uses the wine, Iris and tea datasets in the test. In the test, a tree was added to the federated model each time, and the experimental steps were as follows.

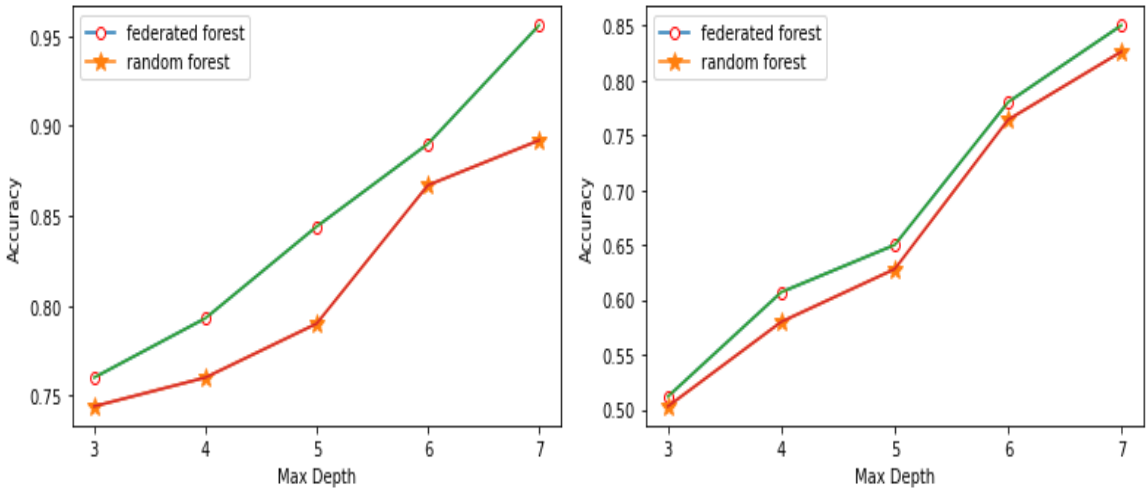
Firstly, the depth of the largest tree was set to 5 and the number of trees was set from 2 to 6. Experiments were performed on the wine, Iris datasets, and the results are shown in (a) and (b) in Figure 2. Where (a) the wine dataset and (b) the Iris dataset.

Secondly, the number of trees was set to 6 and the depth of the largest trees was adjusted from 3 to 7 for experiments on the wine and Iris datasets. The results are shown in (a) and (b) in Figure 3. Where (a) the wine dataset and (b) the Iris dataset.



(a) The wine dataset (b) The Iris data set

Fig 2. Accuracy of federated and random forests (Max Length = 5)



(a)The wine dataset (b) The Iris data set

Fig 3. Accuracy of federated forests and random forests with tree depth (Num of Tree = 6)

The experimental analysis of test 1 yields the following conclusions.

(1) As can be seen from (a) and (b) in Figure 2, the DS-BRF method proposed in this paper has a high prediction efficiency. The reason is that as the number of trees increases, the DS-BRF method produces a great improvement in the prediction efficiency.

(2) From (a) and (b) in Figure 3, the proposed DS-BRF method is relatively robust. Due to the strong linear correlation with the sample size, the results were found to show a linear increasing trend.

(3) As can be seen from Figures 2 to 3, the accuracy of federated forests is constantly improved compared with traditional random forests. It is expected that the training execution time is almost linear with the number of domains. Because all of the features were examined in the tree structure. For prediction time, although more domains and features were added, the difference in execution time was negligible. The results show that the proposed new prediction method (DS-BRF) is very effective in handling many kinds of data.

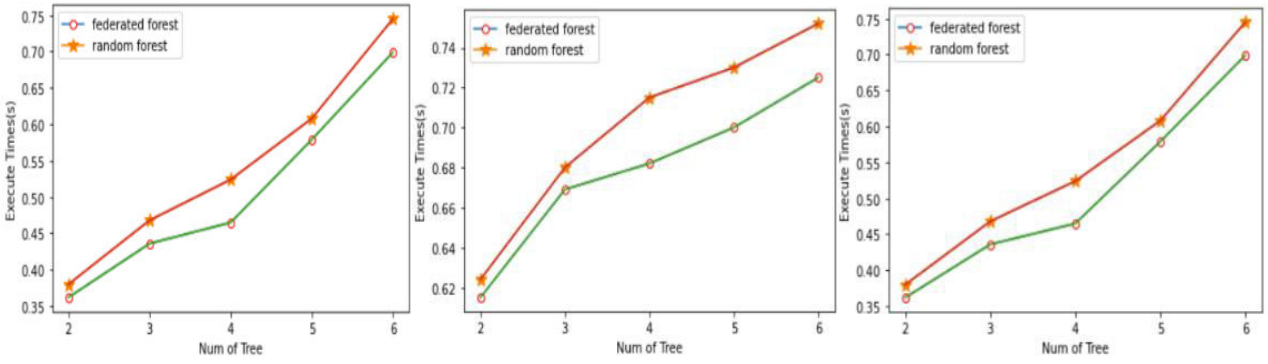
2) Test 2: Prediction Efficiency Experiment

In this part, the DS-BRF method is compared with the classical prediction method regarding the time spent predicting. Experiments were performed on the wine, Iris, and tea datasets.

Firstly, the depth of the largest tree was set to 5, and the number of trees was changed from 2 to 6. Experiments were performed on the wine, Iris and tea datasets and the results are shown in (a), (b) and (c) in Figure 4. Where (a) the wine dataset, (b) the Iris dataset, and (c) the tea dataset.

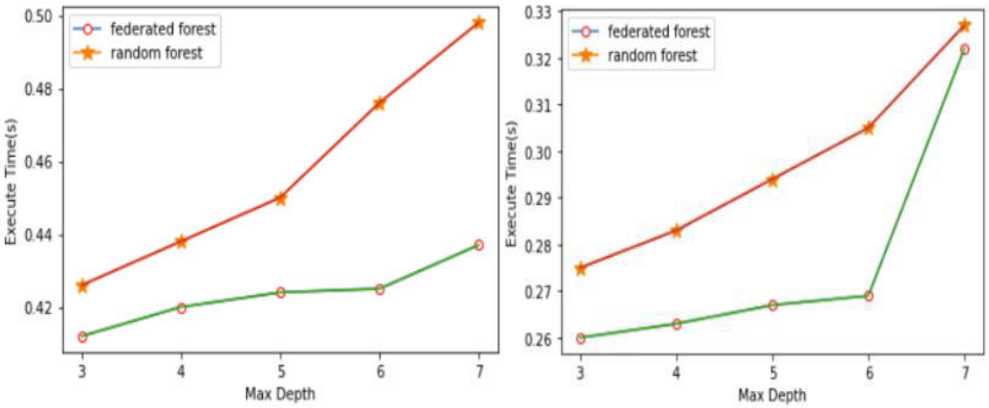
Secondly, the number of trees was set to 6 and the depth of the largest trees was adjusted from 3 to 7 for experiments on the wine and Iris datasets. The results are shown in (a) and (b) in Figure 5. Where (a) the wine dataset and (b) the Iris dataset.

Finally, the number of trees and the depth of maximum large trees were fixed, and the test sampling rate was changed from 0.1 to 0.6. Experiments were performed on the wine, Iris and tea datasets and the results are shown in (a) and (b) in Figure 6. Where (a) the wine dataset, (b) the Iris dataset, and (c) the tea dataset.



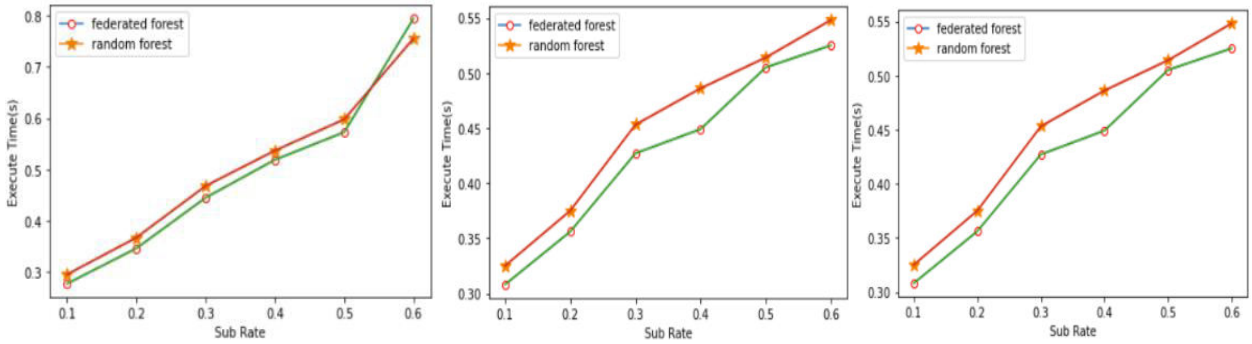
(a) The wine dataset (b) The Iris dataset (c) The tea dataset

Fig 4. Execution time of Federated Forest and Random Forest with tree number (Max Depth = 5)



(a) The wine dataset (b) The Iris dataset

Fig 5. Federated forest and random forest execution time with tree depth (Num of Tree = 6)



(a) The wine dataset (b) The Iris dataset (c) The tea dataset

Fig 6. Federated forest and Random forest execution time with tree depth (Max Depth = 5, Num of Tree = 6)

The experimental analysis of test 2 shows the following conclusions. It can be seen from Figure 4 to Figure 6 that the proposed DS-BRF method has the same effect on different data sets, which also indicates that the DS-BRF method has strong applicability.

Considering the theoretical and experimental analysis, the conclusions are obtained: (1) the proposed DS-BRF method has significant results in terms of prediction accuracy and efficiency, (2) the method is lossless in terms of model accuracy and can protect data privacy, and (3) the method performs well in terms of robustness.

VI. Conclusion

This paper proposes an enhanced data sharing method (DS-BRF) based on blockchain and random forest, which can greatly reduce communication overhead and improve prediction efficiency. Data privacy is protected by redesigning tree algorithms, deploying encryption methods, and building third-party trusted servers. The raw data is never exchanged directly, with only a limited number of intermediate values between the parties. Experiments by wine, Iris as well as the tea dataset show the excellent performance of the proposed models in the classification and regression experiments. At the same time, a new machine learning model based on blockchain and random forest (called the federated forest model) is proposed. The model is lossless in terms of model accuracy and can protect data privacy, based on which a secure cross-regional machine learning system is developed, which allows a learning model to be jointly trained between different clients with the same user sample but different properties. The raw data above each client is not exposed and exchanged to other clients during the modeling process.

In addition, the efficiency and robustness of the proposed DS-BRF method were also verified. All in all, federated forest overcomes the problem of data isolation and privacy challenges in a new way and can be deployed in practical applications.

References

[1] KANG Haiyan, JI Yuanrui, ZHANG Shuxuan. Enhanced Privacy Preserving for Social Networks Relational Data Based on Personalized Differential Privacy [J]. Chinese Journal of Electronics. 2022, 31(4): 741-751.

[2] Kang Haiyan, Wang Xiaoshi. A Deep Learning Method Based on Data Feature Correlation and Adaptive Differential Privacy[J]. Acta Electronica Sinica, 2024, 52(6): 1963–1976.

[3] Li S W, Zhuansun Y, Huang X, et al. FedBAT: Communication-Efficient Federated Learning via Learnable Binarization[J]. Journal of Privacy-Preserving Computation, 2024, 15(2): 213–229.

[4] Zhuansun Y, Li D, Huang X, et al. Communication-Efficient Federated Learning with Adaptive Compression under Dynamic Bandwidth[J]. International Journal of Distributed AI, 2024, 28(1): 45–61.

[5] Kim G H. Communication-Efficient Federated Learning with Accelerated Client Gradient[C]. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024: 1123–1132.

[6] Pillutla K, Kakade S M, Harchaoui Z. Robust aggregation for federated learning[J]. IEEE Transactions on Signal Processing, 2022, 70: 1142-1154.

[7] Chen M, Shlezinger N, Poor H V, et al. Communication-efficient federated learning[J]. Proceedings of the National Academy of Sciences, 2021, 118(17): e2024789118.

[8] Zhang Z, Cai D, Zhang Y, et al. FedRDMA: Communication-Efficient Cross-Silo Federated LLM via Chunked RDMA Transmission[J]. ACM Transactions on Distributed Systems, 2024, 39(2): 1–17.

[9] Mia M J, Amini M H. QuanCrypt-FL: Quantized Homomorphic Encryption with Pruning for Secure Federated Learning[J]. arxiv preprint arxiv:2411.05260, 2024.

[10] Hu C, Li B. MASKCRYPT: Federated Learning with Selective Homomorphic Encryption[J]. IEEE Transactions on Dependable Secure Computing, 2024, 21(2): 324–338.

[11] Choi S, Patel D, Tootaghaj D Z, et al. FedNIC: Enhancing Privacy-Preserving Federated Learning via Homomorphic Encryption

Offload on SmartNIC[J]. Frontiers in Computer Science, 2024, 6: 1465352.
[12] Li Y, Yan N, Chen J, et al. FedPHE: A Secure and Efficient Federated Learning via Packed Homomorphic Encryption[J]. IEEE Transactions on Dependable and Secure Computing, 2025.
[13] Dutta S, Karanth P P, Neira B N, et al. Federated Learning with Quantum Computing and Fully Homomorphic Encryption[J]. Quantum Secure Computing Journal, 2024, 5(3): 201–216.
[14] Wu B, Seneviratne O. Blockchain-based Framework for Scalable and Incentivized Federated Learning[C]//Companion Proceedings of the ACM on Web Conference 2025. 2025: 1761-1767.
[15] Issa W, Moustafa N, Turnbull B, et al. Blockchain-based federated learning for securing internet of things: A comprehensive survey[J]. ACM Computing Surveys, 2023, 55(9): 1-43.
[16] Iranzad R, Liu X. A Review of Random Forest-based Feature Selection Methods for Data Science Education and Applications[J]. International Journal of Data Science and Analytics, 2024, 8(1): 23–35.
[17] Mohosheu S. iBRF: Improved Balanced Random Forest Classifier[J]. arXiv preprint, 2024, arXiv:2403.09867.
[18] Kumar P, et al. Random Forest Pruning Techniques: A Recent Review[J]. Springer Journal of AI and Data Mining, 2023, 11(4): 475–488.
[19] Uddin M R, Shankar G, Mukta S H, et al. Evolving Topics in Federated Learning: Trends and Emerging Directions for IS[J]. Information Systems Frontiers, 2024, 26(1): 1–20.
[20] Kim G H. Communication-Efficient Federated Learning with Accelerated Client Gradient[C]. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024: 1123–1132.
[21] Choi S, Patel D, Tootaghaj D Z, et al. FedNIC: Enhancing Privacy-Preserving Federated Learning via Homomorphic Encryption Offload on SmartNIC[J]. Frontiers in Computer Science, 2024, 6: 1465352.
[22] Cai Z, Chen J, Fan Y, et al. Blockchain-empowered Federated Learning: Benefits, Challenges, and Solutions[J]. IEEE Transactions on Network and Service Management, 2024, 21(1): 100–117.
[23] Sameera K M, Nicolazzo S, Arazzi M, et al. Privacy-Preserving in Blockchain-based Federated Learning Systems[J]. IEEE Access, 2024, 12: 45832–45845.
[24] Jiang Y, Ma B, Wang X, et al. Blockchained Federated Learning for Internet of Things: A Comprehensive Survey[J]. IEEE Internet of Things Journal, 2023, 10(6): 4940–4955.
[25] Li Y, Mu Y. Research and performance analysis of random forest-based feature selection algorithm in sports effectiveness evaluation[J]. Scientific Reports, 2024, 14(1): 26275.
[26] Z. Zhu, J. Hong, and J. Zhou, “Data-free knowledge distillation for heterogeneous federated learning,” arXiv preprint arXiv: 2105. 10056, 2021.