Collaborative Modeling of Medical Image Segmentation Based on Blockchain Network

Yang Luo[1], Jing Peng[1], Hong Su[2], Tao Wu[1], and Xi Wu[1*]
[1] School of Computer Science, Chengdu University of Information Technology
Chengdu 610225, China
[e-mail: 3200607007@stu.cuit.edu.cn]
[2] School of Computer Science, Sichuan University
Chengdu 610065, China
[e-mail: wuxi@cuit.edu.cn]
[*]Corresponding author: Xi Wu

Received October 14, 2022; revised November 26, 2022; accepted February 19, 2023; published March 31, 2023

Abstract

Due to laws, regulations, privacy, etc., between 70-90 percent of providers do not share medical data, forming a "data island". It is essential to collaborate across multiple institutions without sharing patient data. Most existing methods adopt distributed learning and centralized federal architecture to solve this problem, but there are problems of resource heterogeneity and data heterogeneity in the practical application process. This paper proposes a collaborative deep learning modelling method based on the blockchain network. The training process uses encryption parameters to replace the original remote source data transmission to protect privacy. Hyperledger Fabric blockchain is adopted to realize that the parties are not restricted by the third-party authoritative verification end. To a certain extent, the distrust and single point of failure caused by the centralized system are avoided. The aggregation algorithm uses the FedProx algorithm to solve the problem of device heterogeneity and data heterogeneity. The experiments show that the maximum improvement of segmentation accuracy in the collaborative training mode proposed in this paper is 11.179% compared to local training. In the sequential training mode, the average accuracy improvement is greater than 7%. In the parallel training mode, the average accuracy improvement is greater than 8%. The experimental results show that the model proposed in this paper can solve the current problem of centralized modelling of multicenter data. In particular, it provides ideas to solve privacy protection and break "data silos", and protects all data.

Keywords: privacy protection, blockchain, deep learning, collaborative model, data security.

This research was supported by Sichuan Science and Technology Program 2020JDTD0020 and 2019ZDZX0007.

http://doi.org/10.3837/tiis.2023.03.015 **ISSN : 1976-7277**

1. Introduction

Deep learning is growing in popularity worldwide, as shown in our earlier paper [1]. According to the white paper of China Arithmetic Power Development Index [2], the scale of China's arithmetic power continues to expand and the demand for applications continues to rise. With the massive increase in computing power, many ideas in deep learning have been realized. We know that to train a high-precision (98%+model precision) and robust deep learning model, the core element is the need for high-quality (large data scale, accurate labels, balanced categories, etc.) training data. Because if there is insufficient data, the model training results will not fit well. There is no doubt that raw data is crucial for deep learning modelling. Obtaining data from various aspects has always been a challenge for deep learning applications.

Privacy concerns are also a significant impediment to data collection. Some data, such as medical data, are peculiar and cannot be safely shared for further research. As countries around the world propose a series of laws and regulations (e.g. GDPR in the EU, CCPA in the US) to protect the privacy and security of data. This requires that data cannot be interacted with out of local or across domains. The sensitivity of personal data greatly hinders traditional centralized machine learning approaches. Traditional approaches to data sharing will face new legal and regulatory challenges. In addition to laws and regulations privacy and other reasons, the reasons for the non-interoperability of medical information include technical difficulties and the game of interests between hospitals. Most hospitals currently have their own local area networks. The permission setting for the network hinders the access to information from outside. Interoperability of medical information across domains is difficult to achieve between two separate domains [3]. Many medical data sit idle for privacy reasons or local laws and regulations. Many AI companies or AI teams in hospitals can only use their limited data to carry out deep learning or machine learning and other research. This makes the use of medical data for AI algorithm learning a great bottleneck. However, it is hoped to break this bottleneck by using all parties' data in a certain way. The existing phenomenon is that most medical institutions are unwilling to share their data. The amount of data held locally by all parties is not enough to support deep learning modelling, but their modelling demands are necessary.

In order to break the data sharing barriers between healthcare organizations and to ensure the privacy of data between individual healthcare organizations, federated learning (FL) and distributed learning have been the main solutions in the past 10 years of research. **Table 1** compares their advantages and disadvantages.

Table 1. Federated Learning vs. Distributed Learning

Method	**Node Control**	**Data Distribution Type**	**Data volume level**	**Node stability**	**Centralization**
Distributed Learning	no	Independent homogeneous distribution	Evenly distributed	Stability	yes
FL	yes	Non-independent identical distribution	Equipment-related	Instability	yes

The comparison is mainly from five aspects. They are node control, data distribution type, data magnitude, node stability, centralization or not. These two approaches are discussed next.

In the healthcare context, federated learning is used as a common solution to comply with existing privacy laws to protect patient anonymity [4]. FL was first used by McMahan [5] and others. It describes a distributed, privacy-preserving way of training machine learning models. FL relies on sharing model parameters rather than directly sharing source data between untrusted parties. FL system follows a client-server architecture with one server, who is responsible for facilitating the training, building the model, and making it available to all clients who are training the model on their local datasets[6]. Federated learning controls local devices, data does not need to be independently distributed identically, and node loads are often unbalanced, but many designs are centralized. In addition, the training process of the global deep learning model is not discussed in detail in many articles that use decentralized architecture for joint deep learning modeling.

Distributed learning addresses legal and ethical privacy concerns. It also improves the computational performance of machine learning and deep learning models when training them. On distributed machine learning, the euroCAT[7] and ukCAT[8] projects are a proof of distributed learning being successfully implemented into clinical settings to overcome data access restrictions. On distributed deep learning, an example of distributed deep learning in the medical domain is that of Chang et al,[9] who deployed a deep learning model across four medical institutions for image classification purposes using three distinct datasets: retinal fundus, mammography, and ImageNet. In these distributed machine learning strategies, the equipment has no control, and most of the data distributed on the nodes is required to be independent and identically distributed. These requirements are challenging to meet in practical applications.

In addition, the literature on security and privacy has grown in recent years.[10] proposed a hybrid cloud platform with attribute-based encryption strategy. Advanced Encryption Standard (AES) and Attribute-Based Encryption (ABE) algorithms are proposed to ensure the security of data and robustness of quality of service on the cloud platform. A generic IoT blockchain terminal system architecture is proposed in [11]. The system can ensure the security, privacy and data confidentiality of data access control. [12] proposed an LEDA framework to enhance data security and privacy. It can broadly address the known privacy issues in educational environments. Most of them are implemented by cryptographic means. With the development of blockchain technology, it has outstanding application advantages in data management, collaboration enhancement, privacy security and regulation of healthcare. [13-14] are combining IoT and blockchain for securing medical data. They all define the behavior of the health care system through smart contracts. This paper is also based on two key technologies, cryptography and smart contracts, to ensure the security and privacy of data. There are also many scholars conducting research on healthcare data management. [15] describes a platform for managing medical images. It enables secure, efficient and rational storage of medical data. [16] gives a democratic, easy-to-use, and low-cost solution to promote cooperation among organizations. This also provides ideas for the current construction of an efficient healthcare information security sharing model. As a result, technical and non-technical factors such as solutions, cryptography, and blockchain become important enablers for application implementation.

1.1 Motivation

Although federated learning can achieve data availability invisibility compared to traditional data encryption sharing methods. It is a collaborative model training by aggregating all users' encrypted model parameters without data out of local. So it is better able to face new problems and legal constraints that arise in the field of data sharing. However, it is designed to be centralized. If the aggregation server is attacked, the whole system will go down. This is a security risk.

Distributed learning shares only parameters (or metadata) and does not share any instance data to ensure the security and privacy of the data [17]. However, usually distributed learning nodes have no control. It requires data to be distributed evenly among the nodes. In fact, the size of data storage varies from device to device.

To address the strengths and weaknesses of the above techniques, we ask the question. Can we combine federated learning and distributed learning to solve the privacy protection problem of data in the case of non-independent and homogeneous data distribution? The answer should exist in combining the two technologies to eliminate each other's limitations. We use the decentralized, tamper-proof and distributed ledger features of blockchain to create a decentralized platform. We use key technologies such as cryptography and smart contracts to provide a secure framework to accommodate healthcare data sharing.

1.2 Contributions

Based on the background mentioned above, this paper proposes a "blockchain + deep learning" framework for the problem that medical data is challenging to collect centrally for deep learning modelling. This is a decentralized architecture. This paper has novel contributions in the following aspects:

- We combine blockchain and deep learning to help healthcare organizations collaboratively train a deep learning model without exposing their own raw data. Simply exchange weight files with other organizations on a blockchain-based platform. It solves the problem of centralized modelling of multi-center data and effectively solves "data silos".
- We deploy the aggregation function in federated learning to the blockchain as a smart contract. The aggregation of local weights is implemented in a decentralized approach to generate global weights.
- We discuss two approaches (sequential training and parallel training) for collaborative modelling based on blockchain platforms. The effects of the global models produced by these two training methods are compared horizontally. The segmentation accuracy of the global and local models in the same node is compared vertically.

Its implementation is able to help medical institutions jointly train a deep learning model without exposing their data. The decentralized architecture ensures that all parties are not under the jurisdiction of third-party central institutions during the entire training process [18]. All participants have equal rights, the whole data exchange process is transparent. The decentralized mechanism primarily ensures the honest behaviour of the participants. The mode of realizing multi-party cooperation based on such a blockchain platform provides an idea for solving the problem that multi-centre data is challenging to conduct joint modelling in a centralized manner. The data is always saved by the device that generates the data. It will not be accessed by other devices. It has specific protection for data privacy. After experimental verification, the "blockchain + deep learning" proposed in this paper can effectively solve the dilemma of data islands and data privacy protection. In the sequential training mode, the average accuracy improvement is greater than 7%. In parallel training mode, the average accuracy improvement is greater than 8%.

1.3 Organization

The remainder of this paper is organized as follows: Section II presents the design of our framework. This section will be broken down into three subsections. Section 2.1 provides a brief overview of the blockchain. The process of data exchange and the security measures of network transaction are discussed. Section 2.1.1 introduces key algorithms for data upload and transaction verification. This algorithm will be used throughout the trading process. Section 2.2 introduces the strategy of sequential training and the algorithm design of sequential training for blockchain-based networks (Section 2.2.1). Section 2.3 introduces strategies for parallel training of blockchain-based networks. The algorithm design for parallel training (Section 2.3.1) and the selection of aggregation functions in smart contracts (Section 2.3.2) are also covered. Section III describes the design of the experiments and analysis of the results. Section 3.1 describes the experimental background, including the source of the data set, data resolution, data volume, lesion types, evaluation metrics, and classical network segmentation models. Section 3.2 presents information about the hardware and software configuration in the experimental environment. Section 3.3 describes the details in the experiments, including node configuration, data set partitioning (training set, validation set, and test set), and the number of communications in parallel training. Section 3.4 presents the results of the sequential training and analyzes them. Section 3.5 presents the experimental results of parallel training and analyzes them. Section 3.6 discusses the advantages and disadvantages of the two training modes. The fourth section is the conclusion of the article.

2. Proposed model

The blockchain itself has the characteristics of decentralization, tamper resistance, and collective maintenance [19]. It can be used as a data platform for decentralized and secure interaction. Therefore, we propose a novel training of deep learning models based on blockchain to ensure the information security of data interaction between medical organizations. This model can protect data privacy and ensure effective machine learning and deep learning under legal compliance as described in the introduction section. This section will describe how blockchain, as the underlying platform for deep learning, supports the exchange of information among various organizations. The overall system architecture of collaborative training is shown in **Fig. 1**. All medical institutions(nodes) that join the blockchain network share their data in the training process through the blockchain platform. The two main key modules in a blockchain network consist of smart contracts and consensus. Aggregation functions are deployed for use in smart contracts. They will use the medical data they hold to train the network model. The files generated by each institution only use their local data training, and we call it local weights. As in **Fig. 1**, W_N^t denotes the local weights. This local weight file will be added to the blockchain network as a transaction after the medical institution signs it. Nodes in the blockchain network with permission can download. Other medical institutions repeat the above operations. Finally, a certain amount of local weight data is aggregated to generate global weights. As in **Fig. 1**, W_N^{t+1}.This global weight, caused after a finite number of iterative loops, is the best we want in our theory. After several rounds of iterations, this global model will reach its best state.

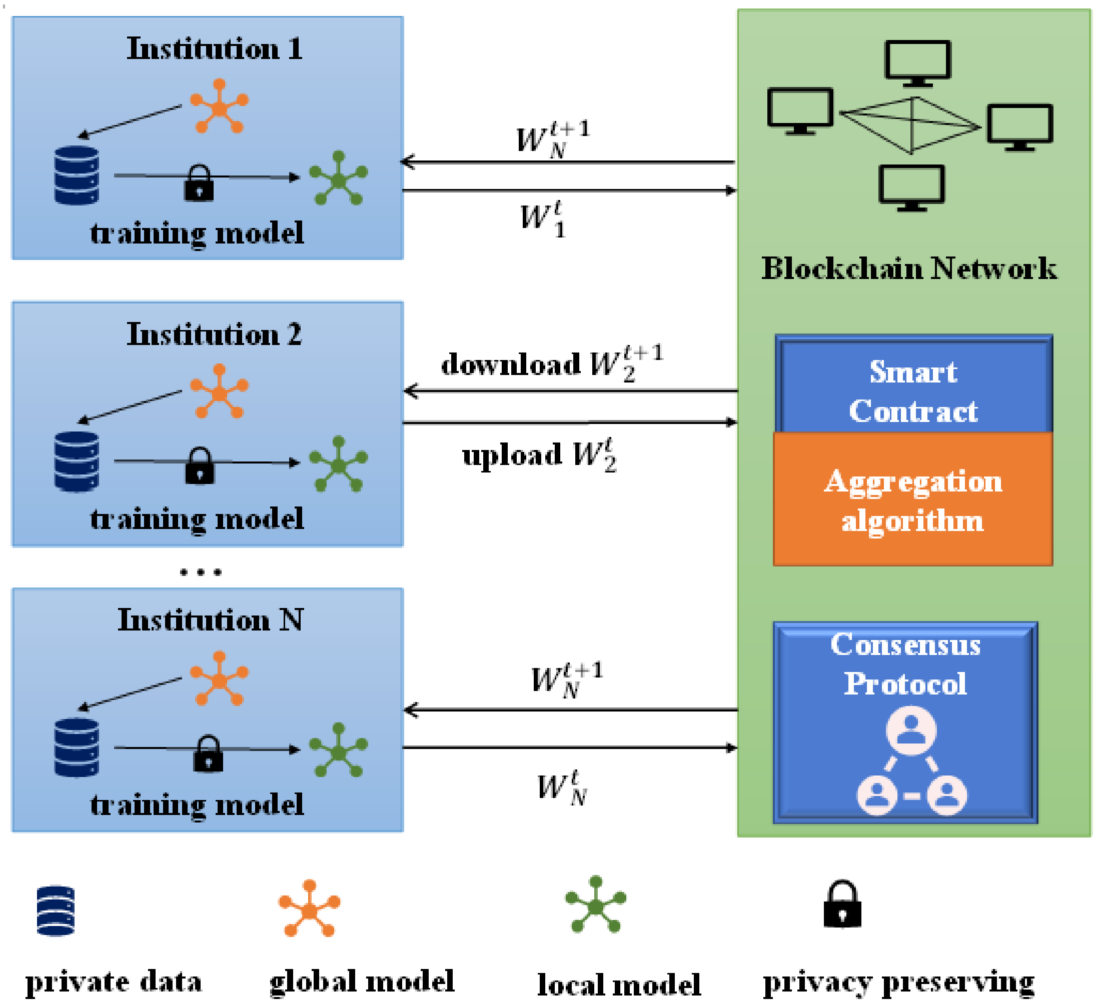


Fig. 1. Overall architecture of the system

We do not expose local data during any of the training mentioned above. Assuming that there are M institutions and k∈M.

$$OM = \{data_1, data_2, \dots, data_k\} \tag{1}$$

$$NM = \{W_1, W_2, \dots, W_k\} \tag{2}$$

$$W_k = train(data_k), where\ data_k \in OM \tag{3}$$

$$W_g = F_{agg}(\,)\ where\ F_{agg} \begin{cases} SequentialTrain(W_k) \\ ParallelTrain(W_k) \end{cases} \tag{4}$$

Eq. (1) represents the original data set, $data_k$ in the set denotes the amount of source data held by the kth node. Eq. (2) represents the weight data set. In Eq. (3), W_k denotes the kth node weight value. In Eq. (4), $W_g$is the global weight parameter. F_{agg} indicates two types of training. Training can be divided into sequential training and parallel training. Each node of the blockchain propagates the weight data in (2) instead of the original data in (1). This enables data protection.

When Sequential Training is used, the aggregate function uses $SequentialTrain$. See 2.2 for a detailed discussion.

When Parallel Training is used, the aggregate function uses $ParallelTrain$. This algorithm is discussed in detail in Section 2.3.

The first section of this section will introduce the blockchain platform of the system. The second part mainly describes how all participants join a blockchain network for collaborative modeling in sequence training. We will verify whether the resulting global model is superior to the local model in convergence. The third section mainly describes how each participant collaboratively trains a global deep learning model based on parallel training. The purpose is to explore whether deep collaborative learning can learn an effective model under the mode of parallel training.

2.1 Blockchain

The consortium chain is the most widely used blockchain method under the current regulatory system in our country. This article uses the Hyperleger Fabric2.0 framework to build a consortium blockchain. The shared ledger mechanism of the consortium chain can significantly reduce the cost of reconciliation, improve data acquisition efficiency, and increase fault tolerance. It is very suitable for the current needs of untrusted parties who need a trusted platform for cooperative modelling. As we all know, POW-based consensus protocols consume many communication and computing resources. Fabric relies on a deterministic consensus algorithm. The ledger does not fork like in other public chains. This paper adopts the core consensus algorithm in Fabric, which is implemented through the Kafka cluster [20]. Compared to other consensus algorithms, the Kafka consensus algorithm is more efficient, energy-saving, and environmentally friendly. It also provides a fault-tolerant mechanism contributing to the system's stable operation.

The analysis of node and transaction flow in the blockchain is shown in **Fig. 2**. In the first step, the client submits a transaction proposal, which is sent to the endorsing node. In the second step, the endorsement node simulates the transaction according to the endorsement policy. The result set signature is generated when the node completes execution. In the third step, the client receives the result set (including the version number and signature of the record). In the fourth step, after the endorsement is successful, the client submits a response to ordering. The content of this response is a transaction to be sorted. Ordering performs a full sort on the proposal responses from the client. It is then packaged into blocks. In step 5, sending the packed blocks in step 4 to the peer node. In the sixth step, the peer node opens the block to verify the version number and signature of the result set according to the endorsement policy. If the condition is met, it is written to the local ledger and updated to worldstate, if not, it is also written to the local ledger but not updated to worldstate. The seventh step， peer also sends event status information to the client. This indicates that the transaction has been submitted to the ledger.

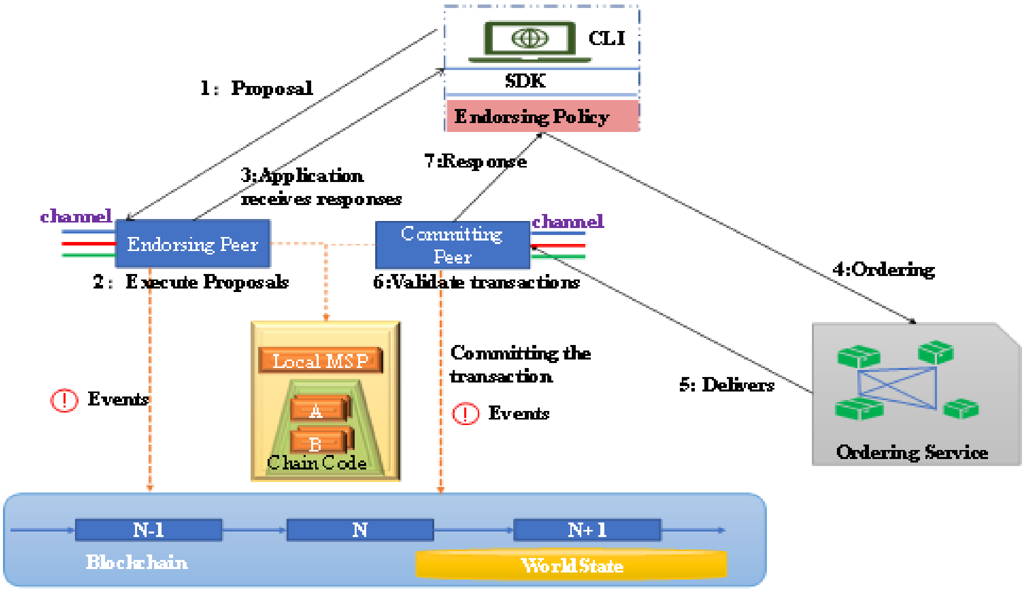


Fig. 2. Nodes and transaction flows

By analyzing the above transaction flow, the problem of fabric transaction replay can be solved. For example: when two transfer transactions are initiated, they are endorsed normally in the endorsing phase. However, in the final validation phase, when the first transaction is successfully executed, the result set changes. The second result assembly found that the version did not match, it would be considered illegal trading. This can avoid the same transaction in the network twice published.

The fabric has other security measures, including the TLS (Transport Layer Security) protocol at the transport layer. On a chain Fabric can isolate multiple ledgers through channels. MSP (Membership Service Provider) and various policies (endorsement policies, smart contract instantiation policies, etc.) implement permission control. Fabric has a very rigorous design and implementation in the communication transmission layer, ledger isolation and permission control, which is the important reason for its data security.

2.1.1 Data uploading and node verification algorithms on the consortium chain:

Algorithm 1 can determine which resources and information in the blockchain network can be accessed by participants. This prevents malicious requests or malicious nodes from attacking the consortium chain.

Some of the parameters are as shown in Table 2:

Table 2. Algorithm 1 Parameter description

Parameter	Explanation
$Data_{file}$	Local weight file
$PK_{receive}$	Recipient's public key
$Threshold$	The minimum number of nodes required to complete the consensus
$Info_{success}$	Data is written successfully.
$Info_{failed}$	Data is written for failure.
$Peer_{num}$	Number of signed nodes

Algorithm1: Data encryption, consensus, and writing to the ledger

```
Input : Data_file, $PK_{receive}$, the signature of node
Output: $Info_{success}/Info_{failed}$
Function: Each participant will encrypt their data and upload it to the blockchain network
for each node in blockchain network do
   $Data_{file}$=$Encrypt(Data_{file}, PK_{receive})$
   Transaction=$Broadcast(Data_{file})$
   $Peer_{num}$=$Verify(signature\ of\ node, Transaction)$
if $Peer_{num}$>$Threshold$ then
  return $Info_{success}$
else
  return $Info_{failed}$
End
```

It is particularly noted that in the algorithm, the encryption algorithm used in this paper is RSA algorithm. Because RSA is an asymmetric encryption algorithm, it is more secure. Second, the algorithm implementation is also relatively simple. The digital signature algorithm used is ECDSA (Elliptic Curve Digital Signature Algorithm). Because the public and private keys of ECDSA are shorter in length, the encrypted message will be smaller. The computational processing time will be shorter, the memory and bandwidth requirements will be smaller, and the compatibility will be higher.

2.2 Train the global model sequentially

This section mainly introduces that each medical institution participating in collaborative training trains a global model through sequential training.

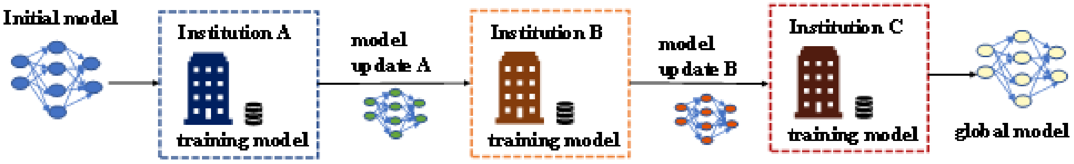


Fig. 3. Sequential training sequence diagram

The sequence diagram of the training process is shown in **Fig. 3**. Here we have three medical institutions A, B, and C. A first uses its data to train a model. During local deep learning training, we train the neural network by forward propagation and backward propagation. when the model converges, it saves the local weight $model_A$. Then the encrypted signature file is uploaded to the blockchain platform, and each node starts to verify the transaction. If the verification passes, the marketing is legal, and a consensus is reached. The transaction is recognized and will eventually be written into the blockchain. Then, B downloads the weight uploaded by A from the blockchain network to the local. It is loaded as the pre-training weight when B performs in-depth learning training. When the model begins to converge, B save the local weight $model_B$. The file will be uploaded to the blockchain platform after B encrypted and signed it. The node will write to the blockchain after a series of verifications. C is the same operation. C uses $model_B$ as the pre-training weight to continue training with its private medical data. The weights generated by C are called $model_C$. Finally, the model training ends when the A, B, and C data are used once. The weights generated by the training of the last node (C) are called global weights.

2.2.1 Sequential training pattern algorithm in consortium chain

Algorithm 2 describes the process of block linking and collecting local weights of each node, verifying transactions, and storing. Nodes download global weights from the blockchain network. Then, it continues the training process using local data.

Some of the parameters are as shown in **Table 3**:

Table 3. Algorithm 2 Parameter description

Parameter	Explanation
$epoch$	The number of times the entire data set is looped
$data_{file}$	Node generated weights file
$batchsize$	Number of training samples
lr	Learning rate
$PK_{receive}$	Recipient private key
W_k	Global weights file
W_{k+1}	The weights after the new round of node gradient update
w	Gradient change after each batchsize
b	Offset

Algorithm2: Data sequence training mode

```
Input: $epoch$,$batchsize$, $lr$, $PK_{receive}$
Output: $true/false$
Function: Description of server and client services based on sequential training
Server execute:
$W_k$=$Receive_{cpk}(node_k$,$data_{file})$
if $Validate(W_k) == true$
   $Save(W_k)$
   return $receive_{success}$
else
   $Discard(W_k)$
return $invalid_{proposal}$
client k+1 execute: //run on the client
$W_k$=$LoadFromServer(proposal$,$PK_{receive})$
$W_{k+1}$=$Parameter_{initial}(W_k)$
for $i = 0$; $i < epoch$ ; $i++$
    for $batchsize$ do
        $W_{k+1}$= $W_{k+1}$-$lr$*$\Delta(w, b)$
if loss converges
    return $W_{k+1}$ to server
End
```

The server in algorithm 2 refers to the whole blockchain network, not a single node.

2.3 Train the global model in parallel

Parallel training means that the nodes in the blockchain network are trained simultaneously using their respective local data. The flow chart of parallel training is shown in **Fig. 4**. Until the global model converges and stabilizes after 200 rounds of communication, each node stops training. It means that every round of communication requires the participation of all nodes in the network. The nodes are trained for 10 epochs per round. The SGD gradient descent algorithm is used to train the neural network. In generating the global model iteratively, each node continuously uses private data to train the model. There is no sequential requirement for uploading and downloading weight in a round of communication.

The parallel training method is different from the sequential training method mentioned above. Compared with the latter, the former does not need to wait for one party to finish the training before continuing the training. Multiple parties can participate in the training simultaneously. The time utilization rate is higher. In each round of training, all data held by each party is used to participate in the training.

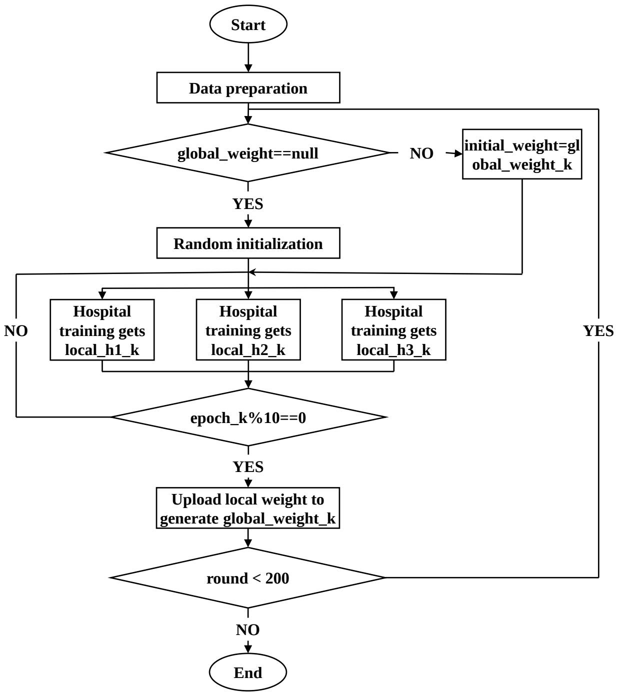

Start
Data preparation
global_weight==null
NO
initial_weight=global_weight_k
YES
Random initialization
NO
Hospital training gets local_h1_k
Hospital training gets local_h2_k
Hospital training gets local_h3_k
YES
epoch_k%10==0
YES
Upload local weight to generate global_weight_k
round < 200
NO
End

Fig. 4. Parallel training flow chart

2.3.1 Parallel training algorithm 3 in consortium chain

This section will introduce the whole process of parallel training mode. Firstly, in the first round of communication, nodes randomly initialize parameters for local deep learning training. Each node is loaded with the current global weight as a pre-training parameter in the subsequent communication. The blockchain network is responsible for aggregating local weights to generate the current new global weight.

Parameters of algorithm three are shown in **Table 4**:

Table 4. Algorithm 3 Parameter description

Parameter	Explanation
$EPOCH$	Went through all the samples in the training set once
$round$	Number of rounds of communication
m	Number of nodes
$batchsize$	The size used in 1 iteration
lr	Learning rate
$seed$	Random seeds are used for weight initialization
W_{k0}	Initial weighting
W_{kt}	Resulting weights after model convergence
W_{kt+1}	On the set of weights of the k nodes
W_{t+1}	Global weights
W	Gradient of node

Algorithm3: Node parallel training mode

```
**Input**: $EPOCH, round, m, batchsize, lr, seed$
**Output**: $true/false$
**Function:**
$W_{k0}$ = $ParameterInitial(seed), k \in m$
$W_{kt}$ = $Train(W_{k0})$
**Server execute:**
**for** $i$=0; $i < round; i++$
  **for** each client in parallel **do**
    $W_{kt+1}$=$ClientUpdate(k, W_{kt})$
  $W_{t+1}$=$FedProx(W_{kt+1})$
**Client update**($k, w$): //run on client k
**for** each epoch from 1 to $EPOCH$ do
   **for** $batchsize$ do
     $W = W - lr * \Delta(w, b)$
     **return** W to server
**End**
```

2.3.2 Selection of aggregate function in parallel training

In the parallel training aggregation mode, we initially adopted the $FedAvg$ federated average algorithm (a weighted average update algorithm based on local stochastic gradient descent) as the aggregation scheme. The formula of $FedAvg$ polymerization gradient [5] is shown in formula (6).

$$f(w) = \sum_{K=1}^{N} \frac{D_k}{D} F_k(w) \quad where \quad F_k(w) = \frac{1}{D_k} \sum_{i \in P_k} fi(w) \tag{6}$$

In the above equation, $F_k(w)$ is the weight information of the kth node and $f(w) = l(x_i; y_i; w)$ represents the loss function of the node. N denotes the number of nodes and K denotes the kth node. P_k denotes the coordinate system. D is the total data, D_k is the amount of data held by the kth node, and $f(w)$ is the new weight information after weighting. The goal of aggregation is to minimize $f(w)$.

However, we found that the local update scheme of the $FedAvg$ algorithm has some problems. The algorithm requires each model to maintain the same learning rate and the number of iterations during training. In each renewal round, each node runs E (local iterations) rounds of SGD locally. Due to $FedAvg$'s ruleset, E can reduce communication costs by increasing local iteration times. Still, too many iterations make some nodes with limited computing power unable to complete training. It makes the local model of the device deviate from the global model easily. Global convergence might be affected. There are some device heterogeneity issues. In addition, as nodes belong to a user or enterprise, data distribution is often very different. A non-IID (non-independent and identically distributed data) problem exists. Heterogeneous distribution means that the data distribution is very different; Non-independent implies that the data may be related due to geography, affiliation, etc. In article [21], the non-IID data dilemma in federated learning is sorted out and introduced. A large number of experiments show that a variety of distributed algorithms, including $FedAvg$, will fail in the non-IID case.

Therefore, this paper adopts the improved $FedAvg$ algorithm. It is known as the FedProx algorithm [22]. Based on (6), the $FedProx$ aggregation formula introduces a proximal term as formula (7):

$$\underset{w}{min} h_k(w; w^t) = F_{k(w)} + \frac{u}{2}\|w - w^t\|^2 \tag{7}$$

In the equation above, $F_{k(w)}$is the loss function of the K-th node. w denotes the weight of this round， μ is used as a parameter term to control the distance between the local and global parameters. The existence of μ can prevent too much deviation from the original global model. w^t represents the global parameter of the previous round. The original $F_k(*)$ was changed into $h_k(*)$. In this way, the gap between the local update and the initial global model would not be too large to minimize the influence on non-IID. It will not be affected by system heterogeneity.

Compared with $FedAvg$, FedProx does not require all nodes to use the same local epoch in each round of global update. $FedProx$ allows different local epochs according to each node's computing power and available resources. In this way, these settings can solve the problem of device heterogeneity. It can work with any optimizer, not just SGD and non-IID scenarios. $FedProx$ can address heterogeneous federated environments while maintaining privacy and computing advantages. It provides better stability and robustness for heterogeneous, decentralized networks.

3. Verification and analysis

3.1 Experimental background description

This dataset contains localized CT images of patients diagnosed with nasopharyngeal carcinoma in the research center hospital in the past two years. The data are real data from the West China Hospital Oncology Centre of Sichuan University. As shown in **Fig. 5**: the original image on the left and the ground truth on the right. We converted each patient's 3D CT images to 2D CT images, removing images without labels [23]. A total of 8889 2D CT images were available for the study.

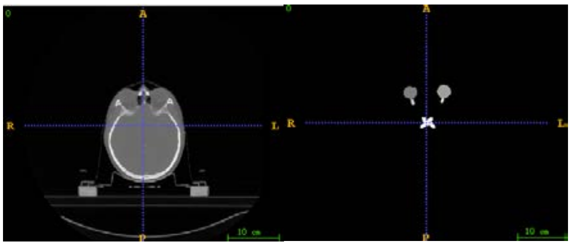


Fig. 5. CT of a patient with nasopharyngeal carcinoma

The original resolution of each image is 512×512. During the training, we cropped the image to 128 × 128 and input it into the network model for training. The demarcation of the target area follows the unified demarcation standard. There are many types of lesions in this dataset, and the amount of data is not small. It is feasible to use this dataset to simulate the data distribution of primary medical institutions. The dataset identified 14 lesion types: left ocular ball, right ocular ball, left optic nerve, right optic nerve, optic chiasma, left mandible, right mandible, left parotid gland, right parotid gland, brain stem, spinal cord, temporomandibular joint, cochlea, and lens.

Dice was used as the evaluation index of the segmentation model. Dice's loss function has the following formula (8):

$$Dice = 1 - \frac{\sum_{n=1}^{N} p_n \gamma_n + \epsilon}{\sum_{n=1}^{N} p_n + \gamma_n + \epsilon} - \frac{\sum_{n=1}^{N} (1-p_n)(1-r_n) + \epsilon}{\sum_{n=1}^{N} (1-p_n) + (1-r_n) + \epsilon} \quad (8)$$

In the above (8), p_n is the predicted probability value, and γ_n is the actual label value. ϵ is a constant in the range (0,1). N denotes the number of samples. n denotes counting from the first sheet. The dice loss function is a classic segmentation loss. The essence of segmentation is pixel-level classification. Since a large number of pixel categories are highly unbalanced, the Dice function can effectively avoid the problem of class imbalance, thus improving the accuracy of segmentation.

To verify the method's effectiveness proposed in this paper, we use the classical standard segmentation network for verification. The validated network models are FCN[24], UNet[25], SegNet[26], PSPNet[27], RefineNet[28], DeepLabv3[29], FastSCNN[30], LEDnet[31]. What we want to verify is whether the collaborative training mode of each node based on the block chain network can be generalized.

Because each case image contains 14 different types of tissues and tumour target areas, different network architectures have specific differences in the accuracy of image segmentation. However, that is not what we want to test. We want to test whether the global model performs better than the local model trained only with our private data.

3.2 Experimental Environment

This section will introduce the experimental environment of the article. The hardware and software configurations for the experiments are shown in **Table 5**.

Table 5. Experimental Configuration Information

Configuration	Model
CPU	Intel(R) Core(TM) i7-7700k 4.20GHz
Memory	64G
Hard disk	20T
Operating System	Ubuntu 18.04.6
Algorithm software	PyCharm 2021.3.2
Blockchain Framework	Fabric 2.0
Deep learning segmentation frameworks	Pytorch 1.8

3.3 Experimental details

In this experiment, we set 3 nodes as master nodes to simulate three medical institutions in the real world. The total data set is the positioning of CT images of the aforementioned nasopharyngeal cancer patients, and the local data of each client node is derived from this entire data set. Each client's data is randomly selected from the complete data set. Each client's data is guaranteed to be disjoint. In this way, the characteristics of non-IID distribution are simulated. The original data set is divided into three disjoint subsets. These subsets serve as local private datasets for each simulated client. We split the allocated dataset into the training set, validation set, and test set with a ratio of 6:2:2, respectively.

In parallel training, the number of communication rounds adopted in this paper is 200. We assume that each client node will not be offline in each round of communication and will receive data packets from other nodes in the network. Suppose a party cannot communicate with other nodes in a round of communication because of network bandwidth or computer failure. In that case, this round of communication is invalid. Packets are sent again until the data is successfully exchanged in the round. Each round of communication includes three significant processes: blockchain network broadcast, node-local data update, and blockchain network aggregation. In the case of training the same model, the number of parameters contained in each round of communication is fixed.

3.4 Experimental results and analysis of sequential training

When using sequential training methods, each node first uses its private data in the case of random initialization parameters to train the network model until convergence. According to each node model training effect, we will give rank from high to low. Assuming that there are 3 nodes A, B and C. These three nodes respectively represent three hospitals in real life. Suppose that after the first round of training, the rank of training effect from high to low are C, B, A. Then in the second round of training, B loads the weight of C as the pre-training weight to continue training, and the training result is new_B. In the third round of training, A loads new_B as the pre-training weight to continue training, and so on. When the data of all nodes are used once, the weight obtained by the last node training is the global weight. This global file will be broadcast as a transaction across the blockchain network. At this point, nodes in the same channel can download the global weight file. Then, they use their respective local test sets to calculate the accuracy to evaluate the effect of the global model.

The above is the whole process of generating a global model based on the sequential training method of each node of the blockchain network. The experimental results obtained in this way are shown in **Fig. 6.** As shown in the figure, the abscissa represents the standard segmentation model used in the training process. The ordinate represents the segmentation accuracy. The orange pentagon in Hospital 1, the red pentagon in Hospital 2 and the yellow pentagon in Hospital 3 respectively represent the training effect of the global model. The rest represents the training effect of the local model. It can be seen from the above experiments that the training results of the global model of the three nodes are improved to different degrees compared with the training results using their private data. The best performing model in this training mode is the UNet model in Hospital 3, with 69% accuracy. The performance of this model gets better and better as medical data input increases. This shows that the accuracy of the deep learning model can be improved through collaborative modelling. It will not disclose the private data of all parties or aggregate the original data of all parties.

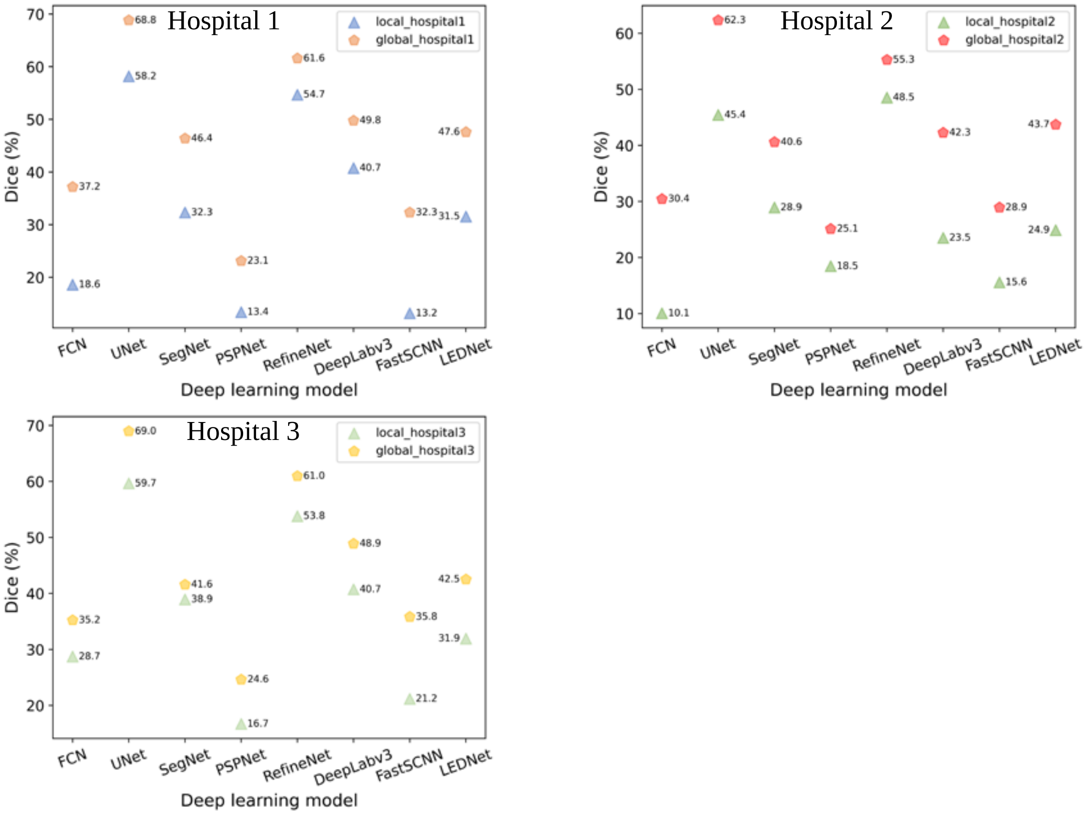


Fig. 6. sequential training experiment effect

3.5 Experimental explanation and analysis of parallel training

Using $FedProx$ as the aggregation algorithm, equation (7) mentioned above 2.3.2, the experimental results of the three hospitals are shown in **Fig. 7**. As shown in the figure, the abscissa represents the classical segmentation model, and the ordinate represents segmentation accuracy. The orange in hospital 1, the gray in hospital 2, and the blue in hospital 3 all represent the training effect of the global model. The rest are the training effects of the local model. As can be seen from the above experimental results, the training effects of different models in the three hospitals have been improved to varying degrees by adopting the method of parallel training. The global model trained by all parties is better than the model trained by each node using its private data in the local test. This also shows that such a training method is feasible and has a specific generalization. In particular, the UNet network has the best segmentation effect on this dataset, reaching 71.148% in the hospital 3 test. The improvement of hospital 2 is the most noticeable improvement at 17.831%. In addition, we employ a federated learning aggregation algorithm, and the results demonstrate that the decentralized architecture proposed in this paper is comparable to its training effect. Our advantage is that we use a decentralized platform blockchain to achieve trustworthiness compared to the centralized architecture of federated learning. In this paper, to further compare the federated learning aggregation method with the centralized data model training paradigm, in the latter, we collect the data of three hospitals together for training. Under the UNet neural network model, the segmentation accuracy reaches 73.192% when the data are pooled together for training. For the centre's collaborative learning model, the accuracy improved by 2.04%. However, this centralized training paradigm requires additional data collection and storage. In general, the federated learning aggregation algorithm and a decentralized platform have more significant technical advantages and more tremendous development potential.

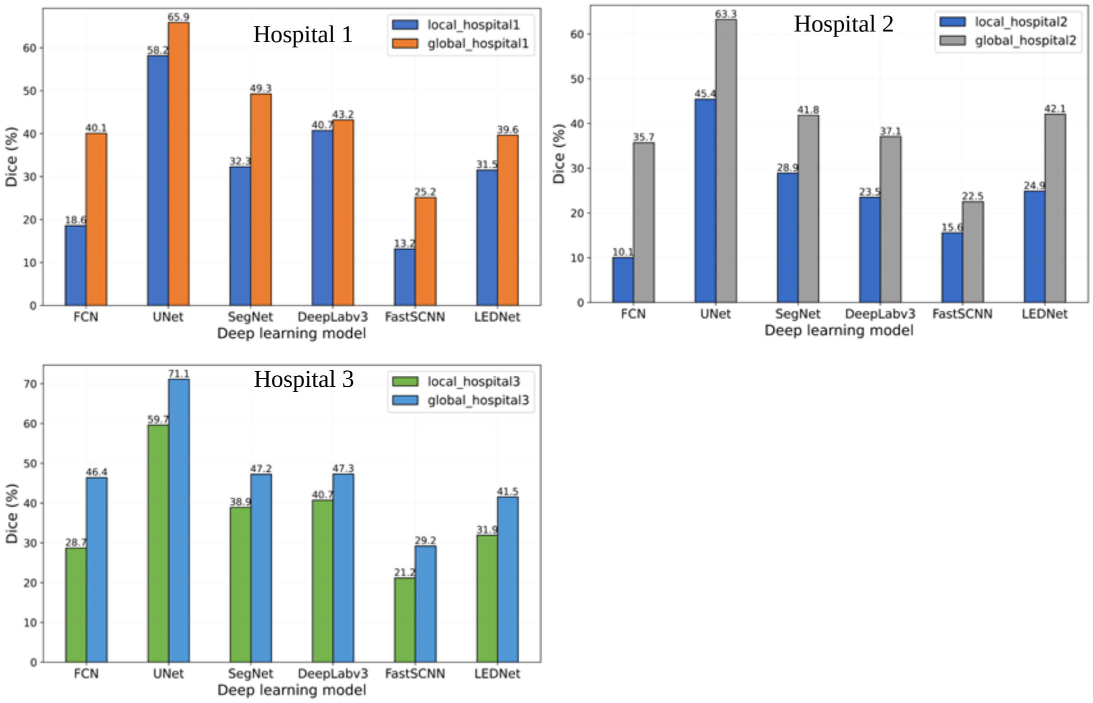


Fig. 7. parallel training experiment effect

3.6 Discussion of the two training methods

There are similarities and differences between the two training methods. The similarity lies in that both sequential training and parallel training carry out deep learning model training under the premise that local private data is not made public. In addition, both training methods utilize all the data held by all nodes for training. That is to say, all data are involved in the modelling process. Finally, from the perspective of the training effect, the results obtained by the two cooperative training methods proposed in this paper (global model) are better than the local model (node only using local data training model).

The difference is that sequential training requires waiting. A node needs to wait for the training of another node to finish before training. The model generated by the last node training is the final global model. In other words, only the last node uses all the data. The number of communication rounds depends on the number of nodes. Assuming that there are

N nodes, N-1 rounds need to be communicated. One round indicates node data download and upload. Local training and the cost of sorting model effects after training are not included. Parallel training does not have to wait. All nodes participate in model training in each round. Every round of training uses data from all nodes. The number of communication rounds is artificially set. This article sets the communication to 200 rounds. Communication costs a lot more than sequential training.

Finally, we also compare the global models generated by the two training modes. **Table 6** shows the results. Bold font indicates that parallel training is better than sequential training in the same institution. Underline indicates that sequential training is better than parallel training. The table shows that the parallel training mode performs better in FCN, UNet, and SegNet segmentation models. The most significant difference between the two was hospital 3's performance in the FCN model, and the result of parallel training was 11.179 percentage points higher than that of sequential training. The sequential training mode performs better in DeepLabV3, FastSCNN, and LEDNet segmentation models, with an average increase of 4.45, 6.72, and 3.51 percentage points. However, there is no significant difference in the overall effect. We believe that the main reason for the gap is that each segmentation model has a different network structure and sensitivity to data sets. In addition, there are some differences between the two training methods. The change in the number of parameters, the change in communication time, and the change in parameter Settings may affect the training effect of the model. This paper mainly discusses whether the global model is better than the local model and whether it is feasible to train the deep learning model based on such a decentralized platform. On the premise that the training effect of the global model is better than that of the local model, we consider the selection of training mode and network model.

Table 6. Global model results from sequential and parallel cooperative trainings.

Institution	**Model**	**Network**					Dice%
		FCN	UNet	SegNet	DeepLabV3	FastSCNN	LEDNet
Hospital1	Global_sqe	37.17	68.83	46.37	49.77	32.31	47.58
	Global_par	**40.08**	65.89	**49.28**	43.19	25.16	39.65
Hospital2	Global_sqe	30.44	62.33	40.59	42.27	28.92	43.72
	Global_par	**35.72**	**63.26**	**41.83**	37.07	22.52	42.08
Hospital3	Global_sqe	35.23	68.97	41.58	48.89	35.81	42.51
	Global_par	**46.41**	**71.15**	**47.24**	47.32	29.20	41.54

The two training methods show that the aggregation scheme will affect the training effect of the final global model. Sequential training will be more efficient than parallel training in terms of efficiency. From the perspective of segmentation accuracy, the two training methods have reached experimental expectations.

4. Conclusion

This paper combines blockchain and deep learning to address the current need for healthcare organizations to collaborate across multiple institutions without sharing patient data. Blockchain trusted platform provides data confidentiality for data exchange and ensures data security between nodes. The consensus mechanism and smart contract guarantee the consistency of multi-participant data and model parameters. Blockchain develops a global model from a scattered set of data by aggregating the parameters of each node. The local private data control is always held by itself in the training process, and only the local model parameters need to be broadcast in the network. We train the deep learning model to protect privacy by designing parameter passing instead of collecting source data.

In this paper, a deep learning segmentation model is constructed for local CT image data of nasopharyngeal carcinoma. The data include 14 lesions. In this paper, two approaches are proposed for modelling collaboratively segmentation in a simulated multi-healthcare organization (multi-user) environment, and the simulation results are shown below. (1) In the sequential training mode, the average accuracy improvement is greater than 7%. In the parallel training mode, the average accuracy improvement is greater than 8%. The multi-institutional blockchain-based platform for collaborative segmentation modelling will significantly outperform the segmentation modelling using their private data. (2) Under the UNet segmentation model, the centralized data training model paradigm (73.192% accuracy) is only about 2% higher compared to the decentralized training model paradigm (71.148%) collaboratively. With model-specific optimization, we protect data privacy and obtain similar model prediction results compared to directly integrating multi-user nasopharyngeal cancer data for segmentation modelling.

This is an effective solution for collaborative modelling of medical images. It can break the barrier that data cannot be shared directly between different medical institutions in traditional segmentation modelling. It solves the problem that it is difficult to collect and share patient data due to medical data privacy. Such an approach breaks the "data island" between medical institutions. This helps to perform collaborative medical image segmentation modelling with privacy preservation. It is also suitable for extension to other related areas of biomedical privacy computing.

References

[1] Sarker, Iqbal H, "Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions," *SN Computer Science*, vol.2, no.6, pp.1-20, 2021. Article (CrossRef Link)

[2] China Academy of Information and Communications Technology, *White Paper on China's Computing Power Development Index*, Beijing, China: CHN, China Academy of Information and Communications Technology, 2021. Article (CrossRef Link)

[3] Azarm-Daigle, Mana, Craig Kuziemsky, and Liam Peyton, "A review of cross organizational healthcare data sharing," *Procedia Computer Science*, vol.63, pp.425-432, 2015. Article (CrossRef Link)

[4] Pfitzner, Bjarne, Nico Steckhan, and Bert Arnrich, "Federated learning in a medical context: a systematic literature review," *ACM Transactions on Internet Technology (TOIT)*, vol.21, no.2, pp.1-31, 2021. Article (CrossRef Link)

[5] McMahan B, Moore E, Ramage D, et al, “Communication-efficient learning of deep networks from decentralized data,” *Artificial intelligence and statistics, PMLR*, pp.1273-1282, 2017. Article (CrossRef Link)

[6] Pfitzner, Bjarne, Nico Steckhan, and Bert Arnrich, "Federated learning in a medical context: a systematic literature review," *ACM Transactions on Internet Technology (TOIT)*, vol.21, no.2, pp.1-31, 2021. Article (CrossRef Link)

[7] Deist, Timo M., et al, "Infrastructure and distributed learning methodology for privacy-preserving multi-centric rapid learning health care: euroCAT," *Clinical and translational radiation oncology*, vol.4, pp. 24-31, 2017. Article (CrossRef Link)

[8] Price, Gareth, Marcel Van Herk, and Corinne Faivre-Finn, "Data mining in oncology: the ukCAT project and the practicalities of working with routine patient data," *Clinical Oncology*, vol.29, no.12, pp. 814-817, 2017. Article (CrossRef Link)

[9] Chang, Ken, et al, "Distributed deep learning networks among institutions for medical imaging," *Journal of the American Medical Informatics Association*, vol.25, no.8, pp.945-954, 2018. Article (CrossRef Link)

[10] Kumar, Abhishek, et al, "A Hybrid Secure Cloud Platform Maintenance Based on Improved Attribute-Based Encryption Strategies," *International Journal of Interactive Multimedia and Artificial Intelligence*, pp.1-8, 2021. Article (CrossRef Link)

[11] Zhang, Guofeng, et al, "STAIBT: Blockchain and CP-ABE Empowered Secure and Trusted Agricultural IoT Blockchain Terminal," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol.7, no.5, pp.66-75, 2022. Article (CrossRef Link)

[12] Amo Filvá, Daniel, et al, "Local technology to enhance data privacy and security in educational technology," *International journal of interactive multimedia and artificial intelligence*, vol.7, no.2, pp.262-273, 2021. Article (CrossRef Link)

[13] Namasudra, Suyel, et al, "Blockchain-based medical certificate generation and verification for IoT-based healthcare systems," *IEEE Consumer Electronics Magazine*, vol. 12, no. 2, pp. 89-93, 2022. Article (CrossRef Link)

[14] Sharma, Pratima, et al, "Blockchain - based IoT architecture to secure healthcare system using identity - based encryption," *Expert Systems*, vol. 39, 2021. Article (CrossRef Link)

[15] García-Peñalvo, Francisco, et al, "Application of artificial intelligence algorithms within the medical context for non-specialized users: the CARTIER-IA platform," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol.6, no.6, pp.46-53, 2021. Article (CrossRef Link)

[16] Lopez, Miguel Angel, et al, "Towards a solution to create, test and publish mixed reality experiences for occupational safety and health learning: training-MR," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol.7, no.2, pp.212-223, 2021. Article (CrossRef Link)

[17] Zerka, Fadila, et al, "Systematic review of privacy-preserving distributed machine learning from federated databases in health care," *JCO clinical cancer informatics*, vol.4, pp.184-200, 2020. Article (CrossRef Link)

[18] Talanov, S. L, "Deviant Behavior in Higher Educational Institutions of the Central Federal District and the Northwestern Federal District: Causes, Scale, Varieties, and Prospects of Control and Prevention," *Russian Education & Society*, vol.56, no.12, pp. 69-81, 2014. Article (CrossRef Link)

[19] Zheng Z, Xie S, Dai H, et al," An overview of blockchain technology: Architecture, consensus, and future trends," *IEEE international congress on big data (BigData congress)*, Ieee, pp. 557-564, 2017. Article (CrossRef Link)

[20] Androulaki E, Barger A, Bortnikov V, et al, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proc. of the thirteenth EuroSys conference*, pp.1-15, 2018. Article (CrossRef Link)

[21] Hsieh, Kevin, et al., "The non-iid data quagmire of decentralized machine learning," *International Conference on Machine Learning, PMLR*, pp. 4387-4398, 2020. Article (CrossRef Link)

[22] Li T, Sahu A K, Zaheer M, et al, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol.2, pp.429-450, 2020. Article (CrossRef Link)

[23] Patil, Dinesh D., and Sonal G.Deore, "Medical image segmentation: a review," *International Journal of Computer Science and Mobile Computing*, vol.2, no.1, pp.22-27, 2013. Article (CrossRef Link)

[24] Long J, Shelhamer E, Darrell T, "Fully convolutional networks for semantic segmentation," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp.3431-3440, 2015. Article (CrossRef Link)

[25] Ronneberger O, Fischer P, Brox T, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. of International Conference on Medical image computing and computer-assisted intervention*, Springer, Cham, pp.234-241, 2015. Article (CrossRef Link)

[26] Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol.39, no.12, pp.2481-2495, 2017. Article (CrossRef Link)

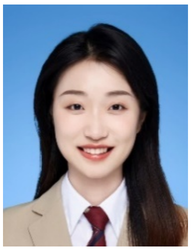
[27] Zhao H, Shi J, Qi X, et al., "Pyramid scene parsing network," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp.2881-2890, 2017. Article (CrossRef Link)

[28] Lin G, Milan A, Shen C, et al., "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp.1925-1934, 2017. Article (CrossRef Link)

[29] Chen, Liang-Chieh, et al, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017. Article (CrossRef Link)

[30] Poudel, Rudra PK, Stephan Liwicki, and Roberto Cipolla, "Fast-scnn: Fast semantic segmentation network," *arXiv preprint arXiv:1902.04502*, 2019. Article (CrossRef Link)

[31] Wang Y, Zhou Q, Liu J, et al, "Lednet: A lightweight encoder-decoder network for real-time semantic segmentation," in *Proc. of IEEE International Conference on Image Processing (ICIP), IEEE*, pp.1860-1864, 2019. Article (CrossRef Link)



Yang Luo received her Bachelor of Engineering degree in Computer Science and Technology from Chengdu University of Information Technology, China in 2020. She is currently pursuing a Master of Engineering degree in Computer Technology at Chengdu University of Information Technology. Her research interests are blockchain and artificial intelligence.

Jing Peng She received the Dr.Sc (Tech) degree in Computer Software and Theory from the University of Chinese Academy of Sciences in 2022. From 2022, she was a lecturer at Chengdu University of Information Technology, China. Her major research directions are image processing, medical image analysis, and artificial intelligence.

Hong Su received the MS, Ph.D degrees, in 2006 and 2022, respectively, from Sichuan University, Chengdu, China. He is currently a researcher of Chengdu University of Information Technology. His research interests include blockchain and the Internet of Value.

Tao Wu received her B.S. and Ph.D. degrees in Computer Science from Southwest Jiaotong University, China, in 2007 and 2014, respectively. She is currently a professor in the School of Computer Science of Chengdu University of Information Technology, China. Her research interests include artificial intelligence, Network security, etc.

Xi Wu received his M.S. and B.S. degrees from the University of Electronic Science and Technology in 2006 and Sichuan University in 2003, respectively. In addition, he received his PhD degree in Information and Communication Engineering in 2012, jointly trained by Sichuan University and Vanderbilt University (USA). He is currently the Dean of School of Computer Science at Chengdu University of Information Technology, China. His research interests are: image analysis and computational imaging, high performance and parallel distributed computing.