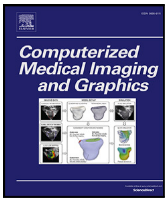
Computerized Medical Imaging and Graphics 102 (2022) 102139

Contents lists available at ScienceDirect

Computerized Medical Imaging and Graphics

journal homepage: www.elsevier.com/locate/compmedimag



Blockchain and homomorphic encryption based privacy-preserving model aggregation for medical images

Rajesh Kumar [a], Jay Kumar [c], Abdullah Aman Khan [d,b], Zakria [e], Hub Ali [f], Cobbinah M. Bernard [d], Riaz Ullah Khan [a], Shaoning Zeng [a,*]

[a] *Yangtze Delta Region Institute (Huzhou), University of Electronic Science and Technology of China, Huzhou 313001, China*
[b] *Sichuan Artificial Intelligence Research Institute, Yibin 644000, China*
[c] *Institute for Big Data Analytics – Dalhousie University, 6299 South St, Halifax NS B3H4R2, Canada*
[d] *School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*
[e] *School of Software Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*
[f] *Institute of Automation,Chinese Academy of Sciences, Beijing 100190, China*

ARTICLE INFO

MSC:
41A05
41A10
65D05
65D17

Keywords:
COVID-19
Privacy-preserved
Federated-learning
Blockchain

ABSTRACT

Medical healthcare centers are envisioned as a promising paradigm to handle the massive volume of data for COVID-19 patients using artificial intelligence (AI). Traditionally, AI techniques require centralized data collection and training models within a single organization. This practice can be considered a weakness as it leads to several privacy and security concerns related to raw data communication. To overcome this weakness and secure raw data communication, we propose a blockchain-based federated learning framework that provides a solution for collaborative data training. The proposed framework enables the coordination of multiple hospitals to train and share encrypted federated models while preserving data privacy. Blockchain ledger technology provides decentralization of federated learning models without relying on a central server. Moreover, the proposed homomorphic encryption scheme encrypts and decrypts the gradients of the model to preserve privacy. More precisely, the proposed framework: (i) train the local model by a novel capsule network for segmentation and classification of COVID-19 images, (ii) furthermore, we use the homomorphic encryption scheme to secure the local model that encrypts and decrypts the gradients, (iii) finally, the model is shared over a decentralized platform through the proposed blockchain-based federated learning algorithm. The integration of blockchain and federated learning leads to a new paradigm for medical image data sharing over the decentralized network. To validate our proposed model, we conducted comprehensive experiments and the results demonstrate the superior performance of the proposed scheme.

1. Introduction

The drastic spread of the novel coronavirus (COVID-19) around the globe caused a large number of deaths within a year. The COVID-19 virus causes acute respiratory disease, which directly infects the human lungs, resulting in intensive breathing difficulties. Due to its highly contagious nature, COVID-19 detection remains among the highest-priority tasks. Currently, various artificial intelligence (AI) techniques are being explored to discover better solutions for detection (Kumar et al., 2021a,b; Deng et al., 2021; Khan et al., 2020). Last year, a significant portion of the research focused on CT scan-based detection, which has proven to be a more reliable source. However, these techniques often require a large amount of data from a single source (hospital or research center) to train the classification model to predict more accurately. In contrast, data from a single source lacks the variance in feature distribution. The less variation in the data, the greater the sampling error and model loss, which consequently, affects the diagnosis performance. The data variation problem can be solved if many hospitals can share data. However, data confidentiality and privacy concerns confine hospitals from sharing the data to train the model. Due to this issue, traditional learning (where only local data is considered) may not generalize properly and perform well in real-world situations. In contrast, transfer learning enables sharing the model weights instead of sharing the actual data. Transfer learning exploits a general pre-trained model and fine-tunes the pre-trained

* Corresponding author.
E-mail addresses: rajakumarlohano@gmail.com (R. Kumar), jay_tharwani1992@gmail.com (J. Kumar), abdkhan@hotmail.com (A.A. Khan), zakria.uestc@hotmail.com (Zakria), hub@ia.ac.cn (H. Ali), vision2bernard2015@gmail.com (C.M. Bernard), rerukhan@gmail.com (R.U. Khan), zeng@csj.uestc.edu.cn (S. Zeng).

https://doi.org/10.1016/j.compmedimag.2022.102139
Received 23 February 2022; Received in revised form 4 August 2022; Accepted 20 October 2022
Available online 3 November 2022

model parameters (weights) (Das et al., 2020; Pathak et al., 2020). The sensitivity of a local model depends on the quality of the pre-trained model. For example, a hospital in a rural area probably has insufficient data to train the model. However, the hospital can collaborate with another hospital while considering the same goal without sharing the actual data. However, transfer learning still confines the base model to increase robustness while taking advantage of local data from the hospital. For this reason, hospitals are still unable to fully benefit from AI-based medical analysis.

A possible solution to overcome the data privacy issue is federated learning. Federated learning is capable of collaboratively training a common model without physically exchanging the actual data. The collaborative model training solves the problem of data variance and enables the evolution of the model over time for all hospitals. i.e., the model can be updated for the latest mutation samples, etc. Generally, federated learning is a collaborative learning framework that allows multiple collaborators to train their local model and share the learned weights for the aggregation process. This aggregation process helps in realizing a robust model that is up to date regarding the latest mutations and samples. Such procedure of gaining knowledge is in the form of a consensus model without moving the actual patients' data beyond the firewalls of the parent data centers (hospitals or research centers). To this point, the learning process occurs locally at each participating institution, and only the model parameters are shared using a federated server for global model aggregation. Originally, federated learning was developed for different domains i.e., distributed learning, edge devices, and mobile computing. Due to its vast scope of applications, it has gained considerable research attention for healthcare applications (Blanquer et al., 2020; Yang et al., 2021; Thwal et al., 2021; Malekzadeh et al., 2021; Li et al., 2020; Baheti et al., 2020; Brisimi et al., 2018).

Recent research has proven that the models trained by federated learning can achieve comparable levels of performance to the ones trained with centrally hosted medical data center (Can and Ersoy, 2021; Dinh et al., 2021; Cheng et al., 2020; Yang et al., 2020). However, there exist some privacy and security concerns with federated learning (Shokri and Shmatikov, 2015) where gradients of local models related to users can be shared without compromising the security and privacy of the data. To this end, previous methodologies are easily accessible to passive attackers, thus making them vulnerable (Dai et al., 2019; Tang et al., 2018) to attacks.

To tackle the security and scalability issues, the blockchain provides a ledger technology that enables model decentralization i.e., without involving any central server. In particular, blockchain provides the facility to collect the data model securely from various points or locations (i.e., Japan, China, Pakistan, USA, and UK) to train the global model. Recent research focuses on federated learning through the use of a central server topology (Blanquer et al., 2020; Yang et al., 2021; Thwal et al., 2021; Malekzadeh et al., 2021; Li et al., 2020; Baheti et al., 2020; Brisimi et al., 2018; Can and Ersoy, 2021). However, none of these studies explored decentralized blockchain-based federated learning for global model aggregation for medical image analysis. Nevertheless, Kumar et al. (2021a) proposed the blockchain-based federated learning-based image detection technique but they did not secure the gradients. Therefore, there is still a gap between secure federated learning (without the dependency of a central server) to provide secure model sharing and trust issues for data providers (e.g., hospitals).

Given the current situation, the model needs to be updated continuously to deal with new types of COVID-virus mutations while considering the previously discussed problems. In this paper, we propose a framework that integrates privacy-preserving federated learning over the decentralized blockchain. For federated learning, we designed a novel capsule network for local hospital training that utilizes image segments for classifying COVID-19 images. Our segmentation network aims to extract nodules from the patients' chest CT scan images. For every locally trained model, gradients are encrypted using a homomorphic encryption technique to preserve the privacy of each hospital. In this encryption technique, hospitals are assigned the same secret key for reducing the communication overhead of high-dimensional data in neural networks. In this way, the client or user side encryption knowledge, which guarantees user privacy using blockchain, ensures the data's reliability. The task of aggregation and learning the global model is carried out over the blockchain. We use the Direct Acyclic Graph (DAG) to improve the blockchains' computation efficiency. The following are the main contributions of this paper:

Table 1
Summary of the mathematical notations.

Notation	Description
$W_i(a)$	Local model weights
$m_i(t)$	Local model learned by devices
CW	The cumulative weight of transactions
W	Weights of the model
$P_{x,y}$	The transition probability of transactions
λ	The 0 and 1 selection state
$\mathbb{Z}_{\mathbb{N}}$	Plaintext space
$\mathbf{A} \xrightarrow{\$} \mathbb{Z}_p^{\kappa\times\tau}$	Matrix
g	Gradients vector for the model
pk/sk	Public/private key
$\otimes$	Product between two ciphertexts

1. We propose a blockchain-based federated-learning framework that enables collaborative data training and decentralization of federated learning models without the involvement of a central server.
2. We utilized a homomorphic encryption scheme for encrypting the weights of the locally trained model which ensures the privacy of the hospital data.
3. We designed a blockchain-based federated learning algorithm to build data models and share the data models instead of actual data. The algorithm aggregates the local model weights to realize the global model.
4. For local model training, we propose a Capsule Network for segmenting pneumonia infection regions and automatically classifying the COVID-19 chest CT images.
5. The proposed framework is continuously updated to deal with new mutations of COVID-virus and quickly exchange the most recent information with hospitals around the world.

2. Preliminaries

This section presents a brief introduction to deep learning, federated learning, homomorphic encryption, and blockchain-based federated learning followed by the system model. The main mathematical notations used in this article are listed in Table 1.

2.1. Deep learning

Usually, deep learning extracts features using deeper convolutional networks to extract features. Further, deep learning models utilize feedforward and backpropagation algorithms to train the model using the obtained features. A general deep learning model with a feedforward neural network is shown in Fig. 1. The feedforward function can be defined as $f(x, w) = y$, where x shows the input vector and w represents the parameter vector. The $D = (x_i, y_i); i \in I$ is the training dataset for the each instance of (x_i, y_i). Moreover l is the loss function, whereas the training dataset D on loss function defined as $L(D, w) = \frac{1}{|D|} \sum_{(\mathbf{x}_i, \mathbf{y}_i) \in D} l\left(\mathbf{y}_i, f\left(\mathbf{x}_i, \mathbf{w}\right)\right)$. However, the backpropagation phase utilizes methods such as stochastic gradient descent (SGD) for updating the parameters.

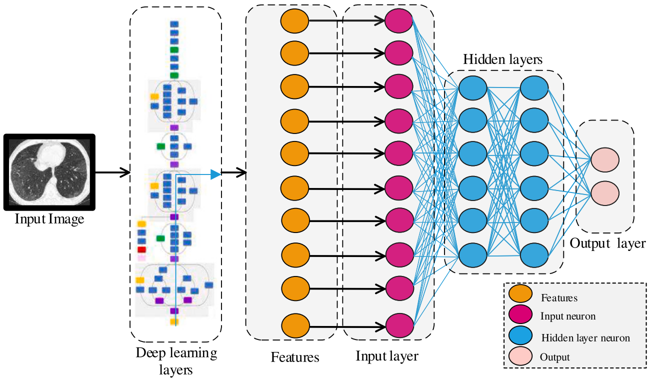


Fig. 1. Overview of general deep learning model.

$$\mathbf{w}^{t+1} \leftarrow \mathbf{w}^{t} - \eta \nabla_{\mathbf{w}} L\left(D^{t}, \mathbf{w}^{t}\right) \tag{1}$$

Where η is the learning rate of the hyperparameter and w^{t} is defined as a vector of i_{th} iteration. Moreover, D^{t} is the training dataset. Eq. (1) shows the standard training procedure to train the data for a hospital or user.

2.2. Swin UNetR segmentation

For the segmentation part, we utilized a Swin UNet Transformer network i.e., (Swin UNetR) (Hatamizadeh et al., 2022). This network helps in the segmentation of lung CT scans. The Swin UNetR encoder extracts with 5 different resolutions using shifted windows while employing a self-attention mechanism. Further, the output is connected to a decoder based on FCNN for each resolution using skip connections.

First, we give an overview of the encoder and later we summarize the decoder. The main architecture of Swin UNetR is illustrated in Fig. 4 (Local Model Feature Learning section). Swin UNetR takes an input of $\mathcal{X} \in \mathbb{R}^{H \times W \times D \times S}$ with a patch resolution of $\left(H', W', D'\right)$ having dimensions of $H' \times W' \times D' \times S$. The first partition layer has a dimension of $\left[\frac{H}{H'}\right] \times \left[\frac{W}{W'}\right] \times \left[\frac{D}{D'}\right]$ which projects the input sequence to an embedding space with dimension C. Non-overlapping windows are utilized for the self-attention mechanism which is created at the partition stage. The windows of size $M \times M \times M$ evenly partitions the input into regions of size $\left[\frac{H'}{M}\right] \times \left[\frac{W'}{M}\right] \times \left[\frac{D'}{M}\right]$ for a transformer encoder layer l. For the next layer i.e., $l+1$, the window is shifted by $\left(\left\lfloor\frac{M}{2}\right\rfloor, \left\lfloor\frac{M}{2}\right\rfloor, \left\lfloor\frac{M}{2}\right\rfloor\right)$ voxels. The subsequent layers can be represented as:

$$\begin{aligned}
\hat{z}^{l} &= \mathrm{W-MSA}\left(\mathrm{LN}\left(z^{l-1}\right)\right) + z^{l-1} \\
z^{l} &= \mathrm{MLP}\left(\mathrm{LN}\left(\hat{z}^{l}\right)\right) + \hat{z}^{l} \\
\hat{z}^{l+1} &= \mathrm{SW-MSA}\left(\mathrm{LN}\left(z^{l}\right)\right) + z^{l} \\
z^{l+1} &= \mathrm{MLP}\left(\mathrm{LN}\left(\hat{z}^{l+1}\right)\right) + \hat{z}^{l+1}
\end{aligned} \tag{2}$$

Where SW-MSA and W-MSA represent the multi-head self-attention modules. Moreover, $\hat{z}^{l}$ represents the output for W-MSA and $\hat{z}^{l+1}$ represents the output of SW-MSA. Furthermore, MLP stands for multi-layer perceptron while LN represents layer normalization. The encoder utilizes a patch size of $2 \times 2 \times 2$ with a dimension of $2 \times 2 \times 2 \times 4$. In total, there are four stages comprising two transformer blocks for each stage. At the first stage, linear embedding layer creates $\frac{H}{2} \times \frac{W}{2} \times \frac{D}{2}$ tokens. To further decrease the feature representation resolution, a patch merging layer is utilized with a factor of 2 (at the end of each stage). In other words, a patch merging layer groups and concatenates patches resulting in a $4C$ sized feature embedding. Further, at each stage, the feature size is reduced by $2C$.

The decoder part utilizes the feature representations extracted by the encoder using skip connections for each resolution. For each stage i of the encoder and the bottleneck the output is reshaped into $\frac{H}{2^{i}} \times \frac{W}{2^{i}} \times \frac{D}{2^{i}}$ and further a residual block of two $3 \times 3 \times 3$ convolutional layers is utilized along with normalization. By using this mechanism, the size of the feature map is increased, by a factor of 2, with the help of deconvolutional layers. Moreover, the outputs are concatenated with outputs of the previous stage. Later, another residual block is utilized. The final segmentation is computed by a $1 \times 1 \times 1$ convolutional layer with sigmoid activation. Further details about Swin UNetR can be found in Hatamizadeh et al. (2022).

2.3. Federated learning

Federated learning is a distributed and secure deep learning technique that enables the training of a shared model while preserving data privacy. Moreover, federated learning has introduced a mechanism to collect data from various parties or hospitals without breaching the hospitals' privacy. The advantage of the federated learning model is that it reduces the resource workload (i.e., memory and power) of a single participant and improves the quality of the training model. In other words, federated learning means learning the model collaboratively and sharing the trained model with the local machines. Suppose, each user $u \in U$ has its private dataset $D_{u} \subseteq D$. The equation for the mini-batch dataset $D^{t} = \bigcup_{u \in U} D_{u}^{t}$ with SGD is given as:

$$\mathbf{w}^{t+1} \leftarrow \mathbf{w}^{t} - \eta \frac{\sum_{u \in u} \nabla_{\mathbf{w}} L\left(D_{u}^{t}, \mathbf{w}^{t}\right)}{|U|} \tag{3}$$

Each user shares the local model over the blockchain distributed ledger for training the global shared model. The users/hospitals upload the new data, i.e., (gradients or weights) for updating the global model. Moreover, each user $u \in U$ has its private dataset with data samples for federated learning as shown in Fig. 2.

$$F_{i}(w) = \frac{1}{\left|D_{i}\right|} \sum_{j \in D_{i}} f_{j}\left(w, x_{i}, y_{i}\right) \tag{4}$$

For multiple devices or hospitals with dataset D, a global loss (Zhu and Jin, 2019) function $f_{j}\left(w, x_{i}, y_{i}\right)$ minimizes the loss. To differentiate between the estimated and real model for each hospital, $f_{j}\left(w, x_{i}, y_{i}\right)$, the global model function of the $F(w)$ is described as:

$$F(w) = \frac{1}{\left|M_{I}\right|} \sum_{i \in I} u_{i} \cdot F_{i}(w) = \frac{1}{\left|M_{I}\right|} \sum_{i \in I} \sum_{i \in D_{i}} u_{i} \cdot \frac{f_{j}\left(w, x_{i}, y_{i}\right)}{\left|D_{i}\right|} \tag{5}$$

where i represents a sample from dataset $\left(x_{i}, y_{i}\right)$ of the gallery $I = \{1, 2, \ldots, n\}$ (Tran et al., 2019), and u_{i} is the number of individual dataset models. In our proposed training process, we enhanced the accuracy of the model by iteratively minimizing the loss function of the global model. The equation of the loss function is given as:

$$Q(w, t) = \underset{i \in I, t \leq T}{\arg\min} F(w) \tag{6}$$

$$Pr\left(w_{i} \in \mathbb{R}_{d}\right) \leq \exp(\epsilon) Pr\left(w_{i}' \in \mathbb{R}_{d}\right) \tag{7}$$

$$\sum_{i=1}^{t} \Delta t(i) \leq \min\left(T_{1}, T_{2}, \ldots, T_{n}\right) \tag{8}$$

where $Pr\left(w_{i} \in \mathbb{R}_{d}\right) \leq \exp(\epsilon) Pr\left(w_{i}' \in \mathbb{R}_{d}\right)$ is the privacy of the users (Lu et al., 2019) of the parameters of the $\left(T_{1}, T_{2}, \ldots, T_{n}\right)$. Moreover, $\Delta t(i)$ is the execution time of the iteration.

2.4. Homomorphic encryption

Homomorphic encryption allows the calculation of encrypted data (ciphertext) without decryption. The new encrypted data matches the result of the operation performed on the unencrypted data after decryption. We utilize the BGV (Brakerski et al., 2014) encryption scheme algorithm, which takes the secret key with large noise and a ciphertext as inputs. It outputs an unencrypted version of the same data with a

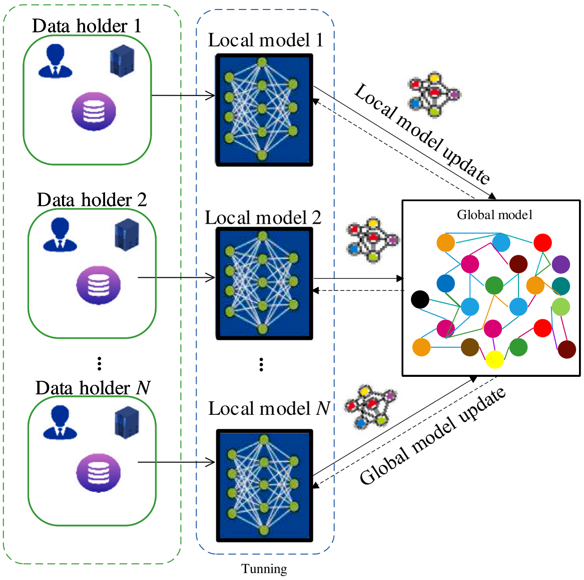


Fig. 2. Overview of a general federated learning process.

fixed amount of noise. Moreover, it utilized a key-switching procedure that allows converting a ciphertext encrypted with a secret key. We refer to the detailed encryption scheme for readers in Brakerski et al. (2014). Therefore, we utilize homomorphic encryption to encrypt the gradients (Aono et al., 2017; Bottou, 2010) to share the data over the blockchain distributed network. The previous research shared the encrypted gradients to the centralized server (Li et al., 2015, 2014). They did not consider a distributed blockchain network. It should be noticed that the blockchain database is cost-effective. For this reason, we use homomorphic encryption to encrypt the model and train the local model which further helps in aggregating the global model.

Before tensor encryption, we define Z as the unencrypted matrix data of the mini-batch dataset having a size of $S * T$, and a private key matrix phi with the size of $S * S$ represented as:

$$\begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1S} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2S} \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{S1} & \phi_{S2} & \cdots & \phi_{SS} \end{bmatrix} \tag{9}$$

This key is only accessible to the users/participants who share the mini-batch dataset.

$$\begin{bmatrix} \mathbb{Z}_{(1)} \\ \mathbb{Z}_{(2)} \\ \vdots \\ \mathbb{Z}_{(S)} \end{bmatrix} = \begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1S} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2S} \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{S1} & \phi_{SS2} & \cdots & \phi_{SS} \end{bmatrix} \otimes \begin{bmatrix} Z_{(1)} \\ Z_{(2)} \\ \vdots \\ Z_{(N)} \end{bmatrix} \tag{10}$$

where $Z(i)$ shows the vector data of the i_{th} node of the blockchain ledger. The ⊗ operator shows the product between two ciphertexts given by:

$$\mathbb{Z}_{(i)} = \phi_{i1} Z_{(1)} + \phi_{i2} Z_{(2)} + \cdots + \phi_{iN} Z_{(S)} \tag{11}$$

Fig. 3 shows the homomorphic encryption function with the linear transformation of a matrix. In this way, the linear transformation maintains a low-rank functionality. The function $\phi_{ij} \in [0, 1)$, and $\sum_{j=1} \psi_{i,j} = 1$ shows the homomorphic encryption with private key.

2.5. Blockchain-enabled federated learning

Training a better AI model for the industry requires collecting data from multiple sources without leaking the privacy and authentication of the users. Therefore, we use federated learning with the blockchain distributed ledger to update the global AI model. The blockchain collects the data model from different nodes and aggregates the local and global models. The smart contract then uploads the weights and updates the models. The proposed architecture integrates blockchain with federated learning for full decentralization and enhanced security. Also, decentralization provides higher accuracy of the model and enables the poisoning-attack-proof.

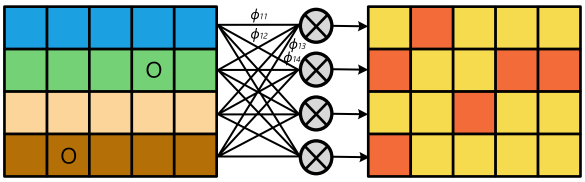


Fig. 3. Graphical representation of homomorphic encryption.

Some issues are not resolved for federated learning, i.e., insufficient incentives, poisoning attacks, etc. Therefore, some authors (Lu et al., 2020b; Qu et al., 2020) design the blockchain with federated learning. Similarly, Pokhrel and Choi (2020) designed a technique to protect privacy. The major issue with the previous papers was that they did not include the encryption technique with the blockchain model gradient sharing. Therefore, this paper uses the directed acyclic graph with the Proof-of-Work (PoW) consensus algorithm for the aggregation of gradients. Additionally, this work is fully decentralized and trains an accurate model without leaking the privacy of the user.

3. Secure data sharing

This section provides an introduction to the high-level architecture of the system and technical details in Fig. 4. Our proposed scheme consists of multiple users sharing the data securely using federated learning with blockchain. The proposed architecture has multiple phases.

Local model:

1. Input COVID-19 images to train the model.
2. Learn the local model and calculate the local gradients.
3. Encrypt the gradients of the local model.

Send the weights to the blockchain network for aggregation model:

1. Aggregate all user weights ciphertext i.e., 1u $W_i(a) \leftarrow \frac{1}{\sum_{i\in S_t} |D_i|} \sum_{i\in S_t} |D_i| W_i(a)$

Broadcast the weights:

1. Update the deep learning model based on global weights.
2. Upload the local model updates.

3.1. Local model training

In this section, we provide the details regarding training the local model for the detection of COVID-19. The main model is divided into three parts: (i) Segmentation Network (ii) Classification (iv)Probabilistic Grad-CAM Saliency Map Visualization.

3.1.1. Segmentation process

We obtained the ground-truth lung masks and extracted lung region using a learning method (Liao et al., 2019; LaLonde and Bagci, 2018). We removed the unnecessary or failed data manually, and the remaining segmentation data is taken as ground-truth masks. The 3D Lung mask also serves as input together with the whole image for training and testing data. The training objective is to adopt the capsule network segmentation. Where $r_{t_i^\ell|xy}$ is the routing coefficient, $b_{t_i^\ell|xy}$ shows the

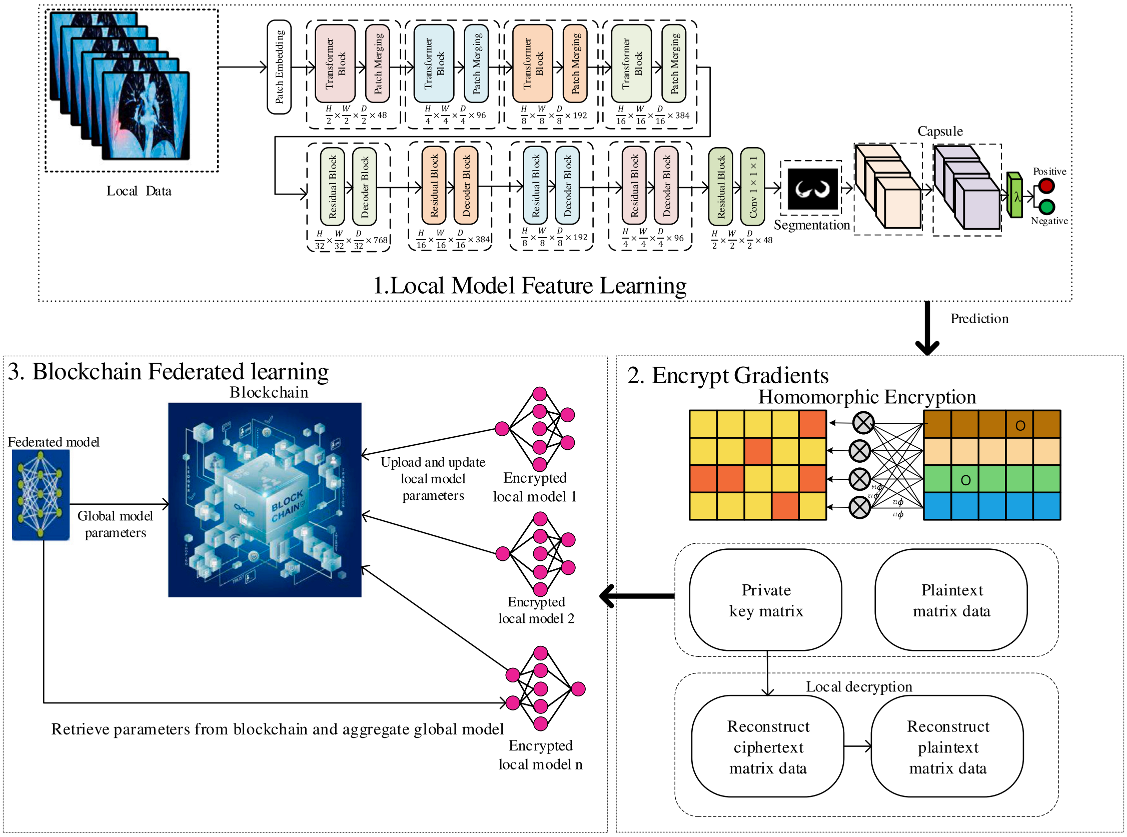

Local Data
Patch Embedding
Transformer Block
Patch Merging
$\frac{H}{2}\times\frac{W}{2}\times\frac{D}{2}\times 48$
$\frac{H}{4}\times\frac{W}{4}\times\frac{D}{4}\times 96$
$\frac{H}{8}\times\frac{W}{8}\times\frac{D}{8}\times 192$
$\frac{H}{16}\times\frac{W}{16}\times\frac{D}{16}\times 384$
Residual Block
Decoder Block
$\frac{H}{32}\times\frac{W}{32}\times\frac{D}{32}\times 768$
Conv 1 × 1 × 1
$\frac{H}{2}\times\frac{W}{2}\times\frac{D}{2}\times 48$
Segmentation
Capsule
Positive
Negative
1.Local Model Feature Learning
Prediction
3. Blockchain Federated learning
Blockchain
Federated model
Global model parameters
Upload and update local model parameters
Encrypted local model 1
Encrypted local model 2
Encrypted local model n
Retrieve parameters from blockchain and aggregate global model
2. Encrypt Gradients
Homomorphic Encryption
Private key matrix
Plaintext matrix data
Local decryption
Reconstruct ciphertext matrix data
Reconstruct plaintext matrix data

Fig. 4. Architecture of blockchain-based secure data sharing using homomorphic scheme. Step 1: training and segmentation of CT scans using capsule network, Step 2: Encrypt the gradients using the homomorphic scheme. Step 3: use a blockchain-based federated learning model for training the global model.

pixel of images, s and y shows the ground truth label. The segmentation is represented as follows:

$$r_{t_i^\ell|xy} = \frac{\exp\left(b_{t_i^\ell|xy}\right)}{\sum_k \exp\left(b_{t_i^\ell k}\right)} \tag{12}$$

To determine the final output of the segmentation using the non-linear squashing function, we have:

$$\mathbf{v}_{xy} = \frac{\left\|p_{xy}\right\|^2}{1+\left\|p_{xy}\right\|^2}\frac{p_{xy}}{\left\|p_{xy}\right\|} \tag{13}$$

where $\mathbf{v}_{xy}$ is the output of the segmented image with the spatial location (x, y) and p_{xy} is the final input.

3.1.2. Classification

We designed a Capsule Network due to the nature of its inverse graph, which helps to detect medical images and achieve high performance. Therefore, the capsule network is capable of predicting the instantiating parameters for any medical image or object. The estimated probability of an object is represented via the length of a vector. However, the Capsule Network technique provides augmented transformations (i.e., rotation, stretching, skewed, thickness, etc.) to improve the performance on a smaller amount of data. The possible probability of the length of the vector is between 1 and 0 using the squashing function of the capsule network. Each layer is connected to the previous layer, and the previous layer is the output of the next layer. Capsule Network does not use a dot product to make the prediction and improve the accuracy. Instead of using dot products, they used the path in a hierarchy or a dynamic routing mechanism to find multiple objects in an image, so they could recognize them. The Capsule Network contains four layers: (i) convolution layers, (ii) primary capsule, (iii) DigitCaps (second capsule), and (iv) fully connected layers. Each layer of the network is composed of multiple capsules in terms of convolution layers (i.e., Conv1 and Conv2); Relu (rectified linear unit) is adopted for activating the convolution layers. As an outcome, each capsule (i.e., Conv1 and Conv2) generates different feature maps. Similarly, the second layer of DigitCaps presents the output layer of the capsule. The loss is calculated after the digits cap. Finally, the fully connected layer is used to reconstruct the images. $W_{i,j}$ are a pair of weighted matrices. A pose vectorU_i is rotated and translated by the weighted matrix to a vector $\hat{u}_{i|j}$ for each component. The instantiation parameters of capsules at higher levels are predicted by the transformation matrix at the same capsule level.

$$\hat{u}_i^j = w_{i,j} \cdot u_i$$

$$\begin{bmatrix} \hat{u}_i^j(1) \\ \vdots \\ \vdots \\ \hat{u}_i^j(16) \end{bmatrix}_{16\times 1} = \begin{bmatrix} w_{i,j}(1) & \cdots & w_{i,j}(8) \\ \vdots & \ddots & \vdots \\ \vdots & & \vdots \\ w_{i,j}(120) & \cdots & w_{i,j}(128) \end{bmatrix}_{16\times 8} \cdot \begin{bmatrix} u_i(1) \\ \vdots \\ u_i(8) \end{bmatrix} \tag{14}$$

In contrast, the prediction vector is defined as follows:

$$\hat{u}_{i|j} = W_{i,j} u_i \tag{15}$$

Predictions from each lower-level capsule are combined to form the next higher level capsule (s_j) the total sum of predictions layers is $c_{i,j}$ represented as a coefficient of $c_{i,j}$ with coupling defined as:

$$S_j = \sum_i c_{i,j}\hat{u}_{i|j} \tag{16}$$

Here, $c_{i,j}$ is a routing softmax function defined as:

$$c_{i,j} = \frac{e^{b_{ij}}}{\sum_k e^{b_{ik}}} \tag{17}$$

As shown in Fig. 4, the parameter $c_{i,j}$, a squashing function is applied for scaling the output probabilities in the range of 0 and 1 and can be defined as:

$$a = \frac{\|a\|^2}{1+\|a\|^2}\frac{a}{\|a\|} \tag{18}$$

For additional information, refer to the original study in Sabour et al. (2017).

3.1.3. CAM map visualization

By visualizing COVID-19 slices, we find the interpretability of the proposed capsule network. The most widely used method is GRAD-CAM (Selvaraju et al., 2017). The GRAD-CAM map takes input as an image using the following equation:

$$l^c(x) = Upsampling\left(\sigma\left(\sum_M \alpha_M^c f^M(x)\right)\right) \in I \tag{19}$$

Upsampling the input image $m \times n$ with the feature vector $u \times v$ produces a σ that is defined as the ReLU layer. However, the probability is determined by:

$$\left[l^c_{prob}(x)\right]_i = \frac{1}{M_i}\left[\sum_{M=1}^{M} r^c\left(x_M\right) Q_M\left(l^c\left(x_M\right)\right)\right]_i \tag{20}$$

where K is the slice of the each image x pixel, $l^c\left(\boldsymbol{x}_M\right)$ computed the GRAD-CAM by using Eq. (19) with respect to frequency of the image. M is computed after the softmax layer of the capsule network. Eq. (20) shows the average probability of each pixel of the class for the global saliency map.

3.2. Architecture of gradients encryption & decryption

The data provider P, which holds the private medical images I, trains the local model and encrypts the local model vector. Then send it to the blockchain network B. The blockchain federated learning model aggregates the encrypted vector using the global federated learning model. Moreover, the gradient encryption & decryption techniques for secure weight sharing were proposed by ElGamal (1985) based on the Ring-LWE scheme. Suppose $\Phi_n(X)$ is the reducible polynomial function The degree of the polynomial function $\phi(n)$, $R_p = R/pR$ and $R = \mathbb{Z}[X]/\left(\Phi_n(X)\right)$ is the polynomial ring. The samples $(a, b = s\cdot a+e)$ of the Ring-LWE, where s, e indicates the Gaussian distribution. We utilized the (Hao et al., 2019) homomorphic scheme for the blockchain ledger.

We define a ciphertext and plaintext space. Ring $R_p = (\mathbb{Z}/q\mathbb{Z})[X]/\left(\Phi_n(X)\right)$ defined as plaintext with the modulus q. Similarity, $R_{p_1} = \left(\mathbb{Z}/p_1\mathbb{Z}\right) \mid X\Big] / \left(\Phi_n(X)\right)$ defined as internal ciphertext RBGV and $R'_{p_1} = \left(\mathbb{Z}/p_1\mathbb{Z}\right)[X]/\left(\Phi_{n'}(X)\right)$ defined as external ciphertext. However, the $\phi(n) = 2l\phi(n)$ with the $l = \lceil \log p_1 \rceil$ and $p_1 = p \cdot p_0$ for primes p, p_0.

Then, we describe some widely used sampling subroutines for better readability represented as follows:

1. $\mathcal{ZV}(n)$: is represented as a vector space of n numbers ranging from $\{-1, 0, 1\}$ with the probabilities of each element being $Pr_{-1} = \frac{1}{4}, Pr_0 = \frac{1}{2}, Pr_1 = \frac{1}{4}$
2. $GM(n, \sigma)$: The n integers are represented as a vector space, with the Gaussian distribution σ and the standard deviation mean 0.
3. $\mathcal{VN}(n, p)$: The n numbers from a randomly uniform distribution modulo p are represented as a vector space.

3.2.1. Setup

Suppose $N \in N$ is the number of devices, and K is the security parameter; for more details, internal encryption is defined as:

1. $Draw\tilde{a} \leftarrow \mathcal{VN}\left(\phi(n), p_1\right)$ and $\widetilde{s}, \widetilde{\gamma} \leftarrow \mathcal{G}M(\phi(n), \tilde{\sigma})$.
2. Compute $b = \widetilde{a}\cdot\widetilde{s} + q\cdot\widetilde{\gamma}$
3. Output $pk = (a, b) \in R_{p_1} \times R_{p_1}$ as public key and $SK_{C_2} = \widetilde{s} \in R_{p_1}$ as part of secret key for the distributed ledger blockchain.
4. Output $sk_i = s_i \in R'_{p_1}$ as secret key for participant i and $SK_{C_1} = -\sum_i s_i \in R'_{p_1}$ as another part of secret key for the distributed ledger blockchain.

3.2.2. Gradients encryption

The following mappings are used during the encryption phase to connect the vector Z^n and the ring R encryption phases:

1. $Map_{R\rightarrow Z}n$: A coefficient representation of an input ring elements of n entities.
2. $Map_{\mathbb{Z}^n} \rightarrow R$: A matrix over the same ring as the vector containing the coefficients representations of the vector

3.2.3. The architecture of gradients encryption decryption for external ciphertext

1. Set $\mathbf{v}_i = Map\left(\widetilde{c}_{i,0} \parallel \widetilde{c}_{i,1}\right) \in \mathbb{Z}_{p_1}^{2\phi(n)}$
2. Invoke Algorithm 1 to sample $\mathbf{e}_i \in \mathbb{Z}_{p1}^{2\phi(n)l}$ subject to the distribution $\Lambda^{\perp}_{\mathbf{v}_i}(\mathbf{G})$, where $\mathbf{e}_i = sample\left(v_{i_1}, \sigma\right)$.........$sample\left(v_{i_{2\phi(n)}}, \sigma\right)$
3. Set $e_i = \left(Map_{Z^{\phi(n'}\,)\rightarrow R'_{n!}}\left(\mathbf{e}_i\right)\right)$
4. Compute $c_i = a \cdot s_i + e_i \in R_{p1}$
5. Send final ciphertext c_i to the blockchain network.
6. Aggregate all the ciphertexts $c = \sum_i c_i = a \cdot \sum_i s_i + \sum_i e_i \in R'_{p_i}$ in the blockchain network
7. Compute the sum of errors terms $\$e = c + a \cdot SK_{C_1} = \sum_i e_i \in R'_{p_1}$, where $SK_{C_1} = -\sum_i s_i$.
8. Set $\mathbf{e} = Map_{R'_{p1}\rightarrow\mathbb{Z}^{\phi(n')}}(e)$

3.2.4. The architecture of gradients encryption decryption for internal ciphertext

1. Set $gd_i^t = Map_{\mathbf{Z}^{\phi(n)}\rightarrow R_q}\left(\mathbf{gd}_i^t\right) \in R_q$.
2. Draw $e_0, e_1 \leftarrow \mathcal{GS}(\phi(n), \sigma)$ and $v \leftarrow \mathcal{ZV}(\phi(n))$.
3. Compute $\widetilde{c}_{i,0} = \widetilde{b}\cdot\tilde{v} + q\cdot\widetilde{e}_0 + gd_i^t$ and $\widetilde{c}_{i,1} = \widetilde{a}\cdot\widetilde{v} + q\ e_1$ for modulus p_1.
4. Output internal ciphertext $\widetilde{c}_i = \left(\widetilde{c}_{i,0}, \widetilde{c}_{i,1}\right) \in R_{p_1} \times R_{p_1}$
5. Recover the sum of RBGV ciphertext by computing $\mathbf{v} = \sum_i \mathbf{v}_i = \mathbf{G}\cdot\mathbf{e} \bmod p_1 \in \mathbb{Z}_{p1}^{2\phi(n)}$
6. Split the vector $\mathbf{v} = \left(\mathbf{c}_0, \mathbf{c}_1\right) \in \mathbb{Z}_{p_1}^{\phi(n)} \times \mathbb{Z}_{p_1}^{\phi(n)}$.
7. Set $\tilde{c}_0 = Map_{\mathbb{Z}^{\phi(n)}\rightarrow R_{p1}}\left(\mathbf{c}_0\right) \in R_{p_1}$ and $\tilde{c} = Map_{\mathbf{Z}^{\phi(n)}\rightarrow R_{p1}}\left(\mathbf{c}_1\right) \in R_{p_1}$
8. Invoke algorithm Scale ($\left(\widetilde{c}_o, \widetilde{c}_1\right), p_1, p_0$) to switch modulus and produce the scaled ciphertext $\tilde{c}_o, \tilde{c}_1$ modulo p_0
9. Decrypt the ciphertext and produce the sum of plaintext by $gd^t = \sum_{i\in[N]} gd_i^t = \widetilde{c}_o - SK_{C_2}\cdot\vec{c}_1 \mod q \in R_q$
10. Set$gd^t = Map_{R_q\rightarrow\mathbf{Z}^{\phi(n)}}\left(gd^t\right)$.
11. Broadcast the global gradients gd^t

3.3. Consensus in permissioned blockchain federated learning

The main goal of this section is to exaggerate the global model with the blockchain DAG mechanism. The local DAG is responsible for synchronous global training via federated learning. Consequently, the storage capability of the model by using DAG is improved. Based on the federated learning and permissioned blockchain, the following steps are taken to adjust the decentralized model for aggregation. Firstly, we select the users' nodes and then perform local training and encrypt the weights. Then, we aggregate the weights in the global model. The consensus (i.e., POW) for data sharing is high cost. To address the problem, we proposed a hybrid DAG-based scheme that is provided in Algorithm 2. We combine the update weight process of federated learning with the quality verification process using the blockchain DAG. Algorithm 12 shows the global aggregation of the model gradients for federated learning.

Algorithm 1: Global Federated Learning aggregation algorithm.

```
$\theta^{t-1}_{\text{global}}$ ← global model;
$\left\{G^{t}_{I(j)}\right\}^{m}_{j=1}$ ← legal gradient vectors;
$g^{t}_{global}$ ← 0;
$l$ ← 0;
for $k=1,2,..m$ do
    if $G^{t}_{I(k)} \neq \perp$ then
        Compute $G^{t}_{\text{global}} \leftarrow G^{t}_{\text{global}} + \alpha_{I(k)} l_{I(k)} G^{t}_{I(k)}$;
        Compute $\leftarrow l + \alpha_{I(k)} l_{I(k)}$
    end
    Compute $G^{t}_{global} \leftarrow \frac{1}{l} G^{t}_{global}$ ;
    Update $\theta^{t}_{\text{global}} \leftarrow \theta^{t-1}_{\text{global}} - \eta G_{\text{global}}$
end
```

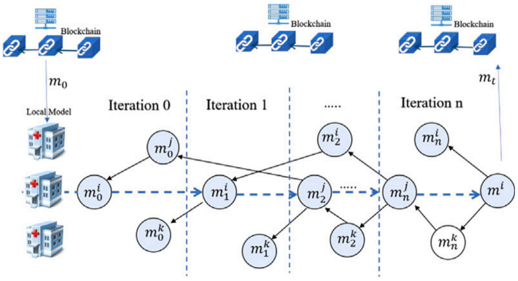


Fig. 5. Communication graph.

3.3.1. *The local directed acyclic graph (DAG)*

The local DAG structure is used individually for each user. In each iteration, t represents federated learning, and permissioned blockchain nodes are selected to verify the aggregation of model u_a. For local weight aggregation of deep learning models, weights $u_i \in u_P$ are transferred to the updated model $m_i(t)$ to the nearby users. Fig. 5 shows the communication graph for the neighboring node. The model accuracy of weights $W(m_i(t))$ is calculated as:

$$W\left(m_i(t)\right) = \frac{|d_i| + \rho \cdot \sum_j d_{m_j}}{\sum_{i=1}^{N} |d_i| + \sum_j d_{m_j}} \cdot s_i \cdot Acc\left(m_i(t)\right) \tag{21}$$

where i is the local training and $|d_i|$ is the dataset size of the model, $\sum_{\mathbf{j}} \mathbf{d}_{\mathbf{m_j}}$ represents the accumulated dataset size of the deep learning local model. S_i execute the each user training slots and $Acc\left(m_i(t)\right)$ shows the accuracy of the each trained model. To verify the reliability of the transaction weights, we calculate weight transaction $CW(m_i(t))$ as:

$$CW\left(m_i(t)\right) = W\left(m_i(t)\right) + \frac{1}{M}\sum_{j=1}^{M} \Delta Acc_j \cdot W(j) \tag{22}$$

where $\Delta Acc_j = Acc_j\left(m_i(t)\right) - W\left(m_i(t)\right)$, $W(j)$ are the weight of the each transaction j, where $m_i(t)$. Acc_j verifies the accuracy of the $m_i(j)$.

3.3.2. *Add the transaction into the blockchain DAG*

To add the transaction to the blockchain DAG and to update the deep learning model first requires validating the local models' two transaction accuracy. Then attach all the hashes and generate a new block. The new block (new transaction) updates the blockchain DAG which can broadcast the nodes in the local model blockchain DAG. The Markov-chain Monte Carlo prototype is used to check the probability of every step. The equation of Markov-chain Monte Carlo is defined as:

$$E[f(x)] \approx \frac{1}{m}\sum_{i=1}^{m} f\left(x_i\right) \qquad \left(x_0, x_1, \ldots, x_m\right) \sim MC(p) \tag{23}$$

3.3.3. *Confirmation and consensus*

The transactions are confirmed or validated based on the cumulative weights. This article utilized the weighted walk method based on credibility, which can validate the transaction by selecting the unverified transactions. When a new transaction is generated, two walkers will be added to the blockchain DAG to select the transaction. More transaction has been passed for verification to achieve a high cumulative weight for verification.

$$P_{xy} = \frac{e^{CW(y)-CW(x)}}{\sum_{z:z\rightarrow x} e^{CW(z)-CW(x)}} \tag{24}$$

where P_{xy} is the transition probability towards the unverified transaction of x and y. z is the neighboring node of a transaction belonging to x, and $y \in \{z : z \rightarrow x\}$. In this approach, the PoW is faster than a traditional PoW because of the reduction in complexity.

Algorithm 2: Federated Learning Empowered with Blockchain Network

```
$D_1 \leftarrow \{M_1, m_2, \ldots, v_N\}$ dataset ;
$m_0$ ← Initialize global weights with the permissioned blockchain BC and
  DAG ;
$r_0$ ← select the users to $M_P \subset M_I$ by the node selection $\{r_1, r_2, \ldots, r_N\}$;
for $e \in [episode]$ do
    Select the leader $r_0$ ;
    for $t \in [timeslot]$ do
        for $D_i$ $dataset \in M_p$ do
            $m_i$ matrix global model $M_{t-1}$ from permissioned blockchain
              BC ;
            $m_i$ = local training $w_i(t) = w(t) - \eta \cdot \nabla F_i\left(w_{t-1}\right)$;
            $m_i$ = get local models updates DAG;
            $m_i$ run the local aggregation model and get the updated local
              model $mi_t$ ;
            $m_i$ add the transactions to the DAG ;
        end
    end
    $r0 \leftarrow (t) = \frac{\sum_{i=1}^{N} C_i w_i(t)}{\sum_{i=1}^{N} C_i}$ DAG blockchain updated the model, and
      averaging the models into $M(e)$;
    $r0$ broadcasts model $M(e)$ to other nodes for verification, and add all
      the transactions into the blockchain ledger ; $r0$ include the $M(e)$
      global model form the blockchain ledger;
end
```

4. Security and performance analysis

4.1. *Dataset*

We collected the dataset from five different hospitals with various types of CT scanners. The total number of patients was 170 and 6 different CT scanners from Chengdu city, Sichuan Province, China. In addition, to validate the proposed method, we combine the open-source dataset with the collected data. All the patients conformed to the antibody tests or nucleic acid tests. However, this research collects the dataset from different sources due to federated learning and blockchain. One dataset is collected from CC-19 Dataset (https://github.com/abdkhanstd/COVID-19) which contains 170 patients from 6 hospitals shown in Table 4 and another dataset is downloaded from Dataverse HARVARD repository[1] to validate the model(see Table 2).

4.2. *Security analysis*

The use of permissioned blockchain distributed technology achieved a secure mechanism for the various devices. We integrate the consensus blockchain process with federated learning to address the trust of the security threats and privacy of the data.

[1] https://doi.org/10.7910/DVN/6ACUZJ

Table 2
Dataset collected from 5 different hospitals A, B, C, D, and E.

Hospital ID	A	B	C	C	D	D	E	F
CT scanner ID	1	2	3	4	5	6	7	8
CT scanner	Brilliance iCT	Samatom Definitation Edge	Brilliance 16P iCT	GE 16-slice CT scanner	Brilliance iCT	SAMATOM scope	Brilliance iCT	SAMATOM scope
CT scanner Company	Philips	Siemens	Philips	Philips	Philips	Siemens	Philips	Siemens
Number of patients	17	3	5	55	50	10	13	20
Number of slices	128	16	16	120	256	64	16	128
Matrix	512*512	512*512	512*512	512*512	512*512	512*512	512*512	512*512
Tube voltage (K vp)	100	140	120	120	120	120	120	120
Collimation (mm)	128*0.6	128*0.625	128*0.625	16*1.25	128*0.625	128*0.6	16*1.2 5	128*0.6
Rotation time (second)	0.35	0.2	0.35	1.0	0.35	0.5	1.0	0.5
Lung window level (HU)	−500	−400	−600	−550	−600	−600	−600	−600
Lung window width (HU)	1200	1200	1600	1500	1600	1600	1200	1600
Pitch	1.0	1.0	1.0	1.75	1.0	1.2	0.938	1.2
Slice thickness (mm)	2	2	5	5	5	5	5	5
Slice increment (mm)	2	2	5	5	5	5	5	5
Tube voltage (K vp)	140	120	120	110	120	120	120	120
Infection annotation	ROI - level	ROI - level	Voxel-level	Voxel-level	Voxel-level	Voxel-level	Voxel-level	Voxel-level

1. **Differential Privacy:** According to the privacy of users, our proposed protocol is used to generate indistinguishable random values. We select the random vector for the generation of the ciphertext $\tilde{c}_{i,}$ using the BGV scheme (Brakerski et al., 2014). Where K is an indistinguishable security parameter for the random values. Then v_i is transferred from the polynomial $\tilde{c}_i$ (for random values).
$$\frac{\Pr[F(x)=S]}{\Pr\,[F\,(x')=S]} \leq e^{\epsilon} \tag{25}$$
A function that satisfies differential privacy is often called a mechanism. We say that a mechanism F satisfies differential privacy if all neighboring datasets x and x' have possible outputs S
2. **Data Access:** The proposed technique uses federated learning with blockchain technology. The core idea is to develop the privacy of the data. The proposed model achieves data privacy by aggregating the encrypted technique with blockchain, which grantee the privacy protection of the data.
3. **Trust:** To aggregate the sum of weights, the blockchain, and the local model client provide the security as follows :

   3.1 Setup: In the first step, the security algorithm generates the public parameters for the model.

   3.2 Encrypt: client specify the parameter (i, m), where i is the index of the entity and m is the plaintext. Finally, it returns the $Enc(i, m)$ value to the model.

   3.3 Compromise: The model comprises an i entity, then the aggregate model returns the secret keys SK_c, this phase repeat many times.

   3.4 Challenge: It is allowed only once throughout the entire cycle. Generate and send two plain text messages, m_1 and m_2, for every $i \in K$. If bit is equal to zero, compute $c_i = Enc\left(m_i\right)$. Otherwise, it will be encrypted in the same way and sent as c_i.

   3.5 Guess: The final output is 1 or 0
4. **Removing Centralized Trust:** It removes third-party trust and allows users or hospitals to comment on the network.
5. **Secure Data Management:** To ensure the model's reliability, only the trusted data provider uploads the data to the network. Also, the cryptography algorithm ensures the security of data.
6. **Guarantee Quality Model:** To ensure the quality of the model, the consensus process guarantees the quality of the learned data.

4.3. Performance analysis

To evaluate the proposed methods' performance, we adopted the federated learning model as a classifier to conduct the experimentation. We analyze and evaluate our model in terms of accuracy. The deep learning model contains fully connected convolutional layers where each of the layers consists of 128 neurons. Two factors affect the accuracy and running time of the federated learning model i.e., the number of hospitals and gradients per hospital. We examined both factors for different value ranges as shown in Fig. 6 and Fig. 7 respectively. It shows the execution time and accuracy with a different number of iterations. Here, the number of iterations indicates the update of parameters. We compared the effect with different numbers of gradients per hospital, and we distributed data over six hospitals. We assume no user has dropped out to conduct the experimentation in the basic setting. It can be seen that increasing the number of gradients per hospital leads to higher performance. Whereas, it leads to a computation overhead as shown in Fig. 6(b). Therefore, to reduce the computation overhead in a practical environment, an appropriate number of gradients can be empirically chosen. In terms of model iterations, it can be observed that model accuracy converges after a certain number of iterations.

The required time to train the local model (local gradients) also depends on the size of the data and the number of selected hospitals. We analyzed the accuracy of different users to train the model. The classification accuracy and execution time can be seen in Fig. 7. Similar to the previous observation, naturally increasing numbers of iterations and hospitals consume high computation costs. However, due to independent gradient computation for each user, the number of hospitals leads to high accuracy. The data is split into many chunks as per the hospital. Therefore, the local gradient for A will be calculated and combined to produce high accuracy.

4.4. Local model analysis

In this section, we analyze the local deep learning models, which are divided into three parts. (i) Segmentation, (ii) Classification, and (iii) Visualization of the Attention Map.

4.4.1. Segmentation network results

Capsule network-based localization of the lung's COVID-19 region is shown in Fig. 8. We extracted the region of the lung of COVID-19 patients. We fix the parameters of the blockchain-based federated learning where total communication costs$T = 300$ and validate each model in every round to select the best local model from the blockchain nodes. Moreover, we set the Adam optimizer learning rate at 0.0001. Each round contains 300 iterations with a batch size of 4. Table 3 shows the federated learning model for the five hospitals. The first three rows show the hospitals (I/II/II). We compute the average of five hospitals' accuracy in the "global test average". This measure shows the global model and blockchain nodes as the major metrics for performance evaluation. Additionally, Table 5 compares the segmentation performance stage-wise.

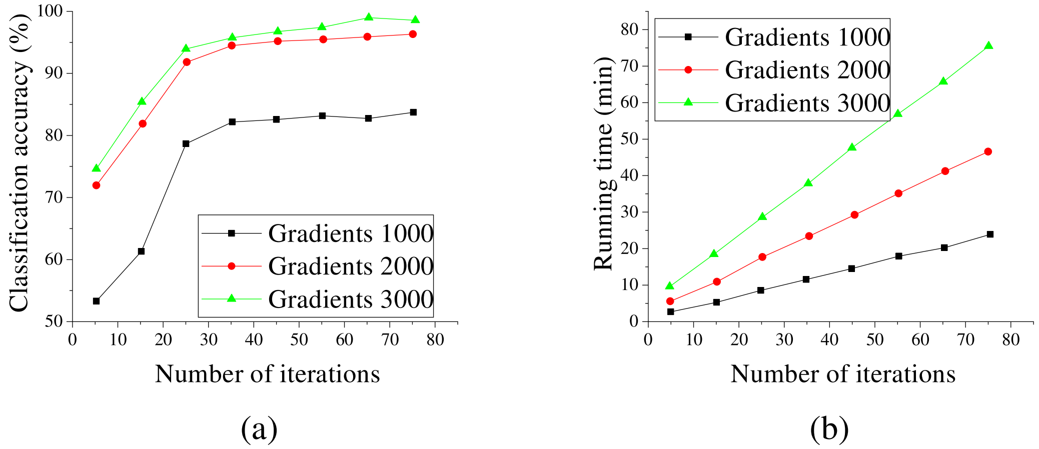


Fig. 6. Hospitals=5, no dropout, classification accuracy and running time for the various number of gradients per hospital.

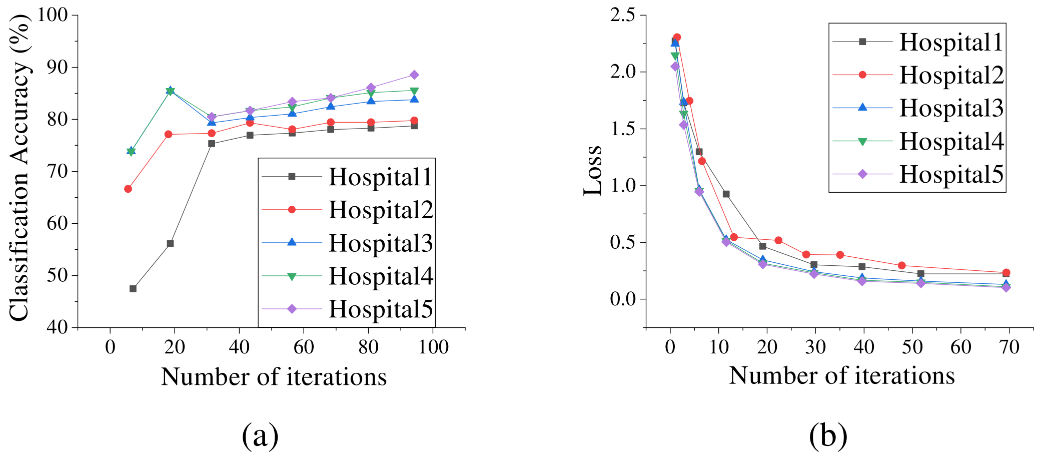


Fig. 7. Gradient=1000, no dropout, classification accuracy and loss for various numbers of hospitals.

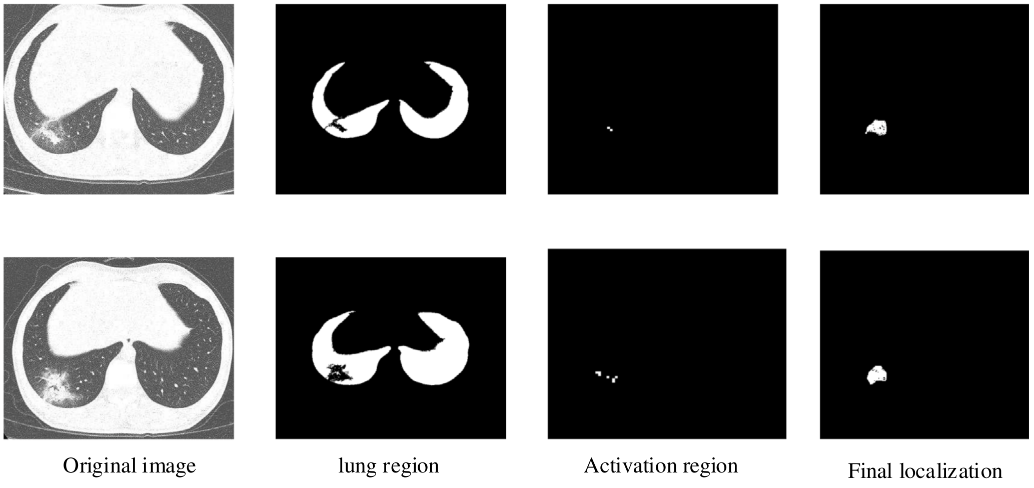


Fig. 8. Activation mapping algorithm segmentation results.

4.4.2. Comparison of the global and local model

This paper presents results from global and local deep learning models i.e., Local I, Local II, Local III, Fed AVG, FedGlobal, and FedProxy. We compared the performance and adopted deep learning models with different layers. Moreover, Fig. 9 shows the performance comparison concerning the segmentation task. Additionally, we evaluate the performance comparison of the capsule network concerning accuracy. Fig. 9 demonstrates the local and global models; the global model achieves high-level detection performance through the network.

4.4.3. Visualizations of the attention map regions

For a better understanding, we computed the probabilistic CAM for each CT image of COVID-19. The capsule network visualizes the patient's CT images from the normal and COVID-19 classes, and a

Table 3
COVID-19 lesion segmentation. The global test average shows the Federated Learning global model. n spices the number of patients.

Parameters	Local - I	Local - II	Local III	FedAvg	FedAvg - Blockchain	FedProx
I (n = 40)	80.2	64.12	57.0	82.13	78.93	82.53
II (n = 20)	84.02	82.15	74.74	85.99	86.51	87.18
III (n = 17)	74.00	72.38	88.05	82.72	87.18	82.65
Global test avg	85.99	82.15	73.16	83.61 ± 0.18	84.21 ± 0.43	84.12 ± 0:58
Local avg	84.07			84.67	84.44	61.99
Local gen	70.99			81.0	81.48	80.53

Table 4
Federated learning segmentation performance at early, progressive, and severe stages.

Method	Early (75%)	Progressive (15%)	Severe (10%)	RMSE	Recall	Dice	Worst-case
FedAvg - Blockchain	0.895	0.925	0.943	**0.025**	0.789	**0.795**	**0.577**
FedAvg	0.769	0.896	0.882	0.082	**0.802**	0.573	0.125
FedProx	0.799	0.912	0.924	0.028	0.702	0.692	0.032
DeepLabV3	0.726	0.820	0.868	0.048	0.759	0.896	0.175
UNet	0.758	0.805	0.855	0.076	0.625	0.459	0.087

Table 5
Federated learning time and memory consumption details.

Method	Predication (s)	Training (h)
FedAvg - Blockchain	12	5.5
FedAvg	13	5.4
FedProx	15	6
DeepLabV3	12	3.5
UNet	10	2

Table 6
Ablation study on the effect of number of users dropout on our proposed blockchain-based federated learning (FedAvg-Blockchain).

Dropout no.	Early (75%)	Progressive (15%)	Severe (10%)	RMSE	Recall	Dice
1	0.882	0.912	0.935	0.030	0.780	0.786
2	0.851	0.901	0.918	0.080	0.703	0.734
3	0.823	0.870	0.881	0.124	0.680	0.690

Table 7
Tradeoff between global average accuracy and privacy.

Privacy budget (ϵ)	Global test avg
0.10	54 ± 0.75
0.20	69 ± 0.92
0.30	73 ± 0.02
0.40	76 ± 0.83
0.50	84 ± 0.21

noticeable activation map is shown in Fig. 10. Moreover, the applied CAM (LaLonde and Bagci, 2018; Liao et al., 2019) function visualizes each image slice.

4.5. Trade off between accuracy and privacy

For this experiment, we measure the global test average of the proposed model corresponding to different privacy budgets, and in each case, we set the sensitivity of the additional noise added to the model's weights as the optimal probability that increases the privacy of the framework. Here, the privacy budget indicates the overall end-to-end privacy loss for the participating user and smaller implies higher privacy. As shown in Table 7, by increasing the privacy budget $\epsilon < 1$ value from 0.10 to 0.50, the corresponding test average keeps increasing, matching the intuition that higher privacy corresponds to lower accuracy.

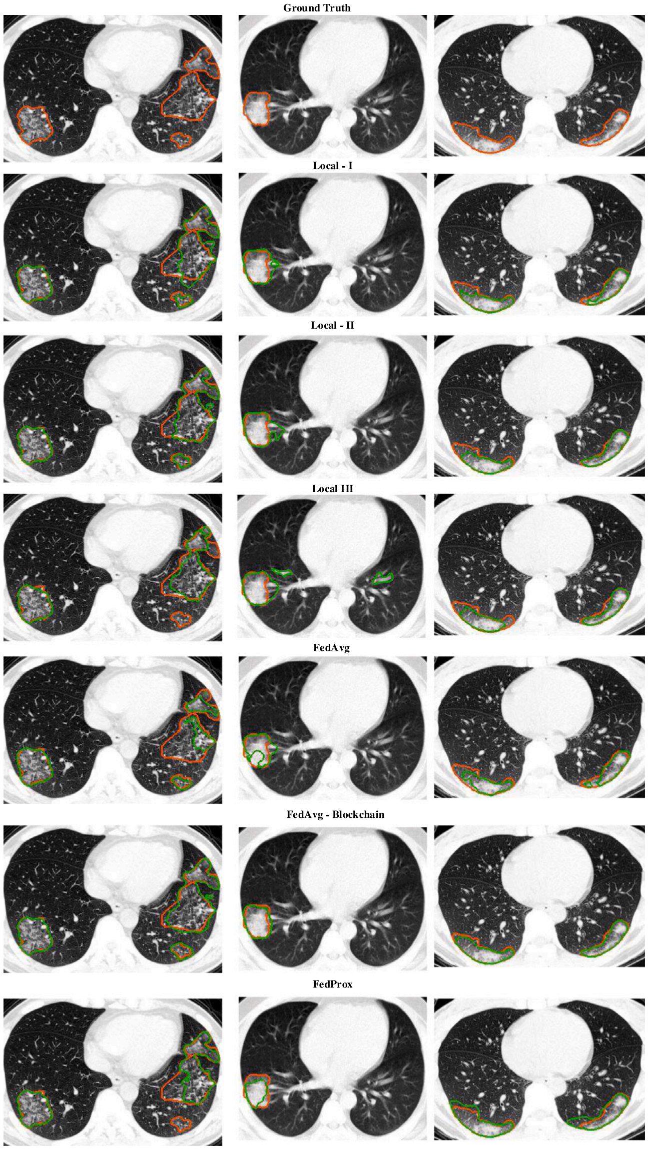


Fig. 9. Activation mapping algorithm segmentation results.

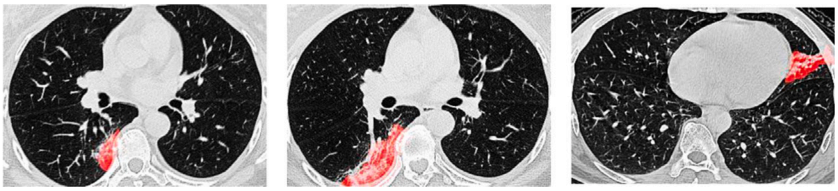


Fig. 10. Visualizations of the attention map regions.

Table 8
Comparison with the security analysis. Furthermore, DA represents data authentication, P/E represents Privacy/Encryption of data Data, DaA represents Data Access and CeT represents Centralized Trust.

Study	Blockchain	Server	DA	P/E	DaA	CeT
Ours	Yes	No	Yes	Yes	Yes	Yes
Kumar et al. (2021a)	Yes	No	Yes	No	Yes	Yes
Kim et al. (2019)	Yes	No	Yes	No	Yes	Yes
Lu et al. (2020b)	Yes	No	Yes	No	Yes	Yes
Lu et al. (2020a)	Yes	No	Yes	No	Yes	Yes
Xu et al. (2019)	No	Yes	No	Yes	Yes	No
Yang et al. (2014)	No	Yes	No	Yes	Yes	No

4.6. Comparison with other methods

To prove the local models' accuracy and effectiveness, the proposed model (as we can observe), the capsule network achieved 98% accuracy in detecting the COVID-19 CT scans. Although Han et al. also achieve 98% accuracy, they do not consider the data sharing techniques. Furthermore, we compare our scheme with the security analysis shown in Table 8. Moreover, Bonawitz et al. (2017) adopted federated learning and proposed a framework to secure the aggregation of gradients. Whereas, Zhang et al. (2017) introduced the scheme of homomorphic encryption (HE) and threshold secret sharing to secure the gradients. The main problem with sharing the model is uncertainty about user authenticity. In other words, there is still a lack of trust among various groups. Thus, the proposed approach bridges this gap and achieves the desired result of trust between parties.

4.7. Ablation study

We further performed additional experiments to ascertain the effect of user dropout for our proposed federated blockchain framework for medical images. Users participating in the federated learning task may drop from the federated learning systems at any time due to several reasons, such as low device battery, poor connectivity, etc., which may affect the performance of the model. For each dropout setting, we randomly drop out $n \in \{1, 2, 3\}$ number of users for each training round. In Table 6, it could be realized that the performance of our proposed method reduces as the number of drop-out users increases, which is consistent with the literature. The reason for the performance drop is likely due to the divergence of weights of the local models from the global model (Sattler et al., 2019). Also, when users drop out from the federated learning round, it indirectly decreases the total number of data instances to aid the training process. Even though there was a reduction in performance as the number of users dropped out from the training, the performance reduction is insignificant.

5. Conclusion

This paper proposes a secure data sharing scheme for distributed multiple hospitals for the internet of things applications that includes both local model training and secure global model training. We secure the local model through the homomorphic encryption scheme which helps build an intelligent model without leakage of the data providers' privacy and creates trust in the data training process. However, the blockchain-based algorithm aggregates the local model updates and provides the authentication of the data. The experiment results confirm the performance and effectiveness of the model. In future work, we aim to enhance the latency of the blockchain and provide a more cost-effective solution.

CRediT authorship contribution statement

Rajesh Kumar: Writing – original draft, Conceptualization, Methodology. **Jay Kumar:** Data curation, Visualization. **Abdullah Aman Khan:** Data curation, Visualization. **Zakria:** Draft, Review, Validation. **Hub Ali:** Formal analysis. **Cobbinah M. Bernard:** Software, Supervision. **Riaz Ullah Khan:** Writing – review & editing, Validation. **Shaoning Zeng:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant U2033212.

References

Aono, Y., Hayashi, T., Wang, L., Moriai, S., et al., 2017. Privacy-preserving deep learning via additively homomorphic encryption. IEEE Trans. Inf. Forensics Secur. 13 (5), 1333–1345.

Baheti, P., Sikka, M., Arya, K.V., Rajesh, R., 2020. Federated learning on distributed medical records for detection of lung nodules. In: Farinella, G.M., Radeva, P., Braz, J. (Eds.), Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP 2020, Volume 4: VISAPP, Valletta, Malta, February 27-29, 2020. SCITEPRESS, pp. 445–451.

Blanquer, I., Brasileiro, F.V., Brito, A., Calatrava, A., Carvalho, A., Fetzer, C., Figueiredo, F., Guimarães, R.P., Marinho, L.B., Jr., W.M., da Silva, A.S., Alberich-Bayarri, A., Camacho-Ramos, E., Jimenez-Pastor, A., Ribeiro, A.L.L., Nascimento, B.R., Silva, F., 2020. Federated and secure cloud services for building medical image classifiers on an intercontinental infrastructure. Future Gener. Comput. Syst. 110, 119–134.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A., Seth, K., 2017. Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. pp. 1175–1191.

Bottou, L., 2010. Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010. Springer, pp. 177–186.

Brakerski, Z., Gentry, C., Vaikuntanathan, V., 2014. (Leveled) fully homomorphic encryption without bootstrapping. ACM Trans. Comput. Theory (TOCT) 6 (3), 1–36.

Brisimi, T.S., Chen, R., Mela, T., Olshevsky, A., Paschalidis, I.C., Shi, W., 2018. Federated learning of predictive models from federated electronic health records. Int. J. Med. Inf. 112, 59–67.

Can, Y.S., Ersoy, C., 2021. Privacy-preserving federated deep learning for wearable IoT-based biomedical monitoring. ACM Trans. Internet Techn. 21 (1), 21:1–21:17.

Cheng, Y., Liu, Y., Chen, T., Yang, Q., 2020. Federated learning for privacy-preserving AI. Commun. ACM 63 (12), 33–36.

Dai, H.-N., Zheng, Z., Zhang, Y., 2019. Blockchain for internet of things: A survey. IEEE Internet Things J. 6 (5), 8076–8094.

Das, N.N., Kumar, N., Kaur, M., Kumar, V., Singh, D., 2020. Automated deep transfer learning-based approach for detection of COVID-19 infection in chest X-rays. Irbm.

Deng, J., Khokhar, M.S., Aftab, M.U., Cai, J., Kumar, R., Kumar, J., et al., 2021. Trends in vehicle re-identification past, present, and future: A comprehensive review. arXiv preprint arXiv:2102.09744.

Dinh, C.T., Tran, N.H., Nguyen, M.N.H., Hong, C.S., Bao, W., Zomaya, A.Y., Gramoli, V., 2021. Federated learning over wireless networks: Convergence analysis and resource allocation. IEEE/ACM Trans. Netw. 29 (1), 398–409.

ElGamal, T., 1985. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inform. Theory 31 (4), 469–472.

Hao, M., Li, H., Luo, X., Xu, G., Yang, H., Liu, S., 2019. Efficient and privacy-enhanced federated learning for industrial artificial intelligence. IEEE Trans. Ind. Inf. 16 (10), 6532–6542.

Hatamizadeh, A., Nath, V., Tang, Y., Yang, D., Roth, H., Xu, D., 2022. Swin UNETR: Swin transformers for semantic segmentation of brain tumors in MRI images. arXiv preprint arXiv:2201.01266.

Khan, A.A., Shafiq, S., Kumar, R., Kumar, J., Haq, A.U., 2020. H3DNN: 3D deep learning based detection of COVID-19 virus using lungs computed tomography. In: 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing. ICCWAMTIP, IEEE, pp. 183–186.

Kim, H., Park, J., Bennis, M., Kim, S.-L., 2019. Blockchained on-device federated learning. IEEE Commun. Lett. 24 (6), 1279–1283.

Kumar, R., Khan, A.A., Kumar, J., Zakria, A., Golilarz, N.A., Zhang, S., Ting, Y., Zheng, C., Wang, W., 2021a. Blockchain-federated-learning and deep learning models for covid-19 detection using ct imaging. IEEE Sens. J..

Kumar, R., Wang, W., Kumar, J., Yang, T., Khan, A., Ali, W., Ali, I., 2021b. An integration of blockchain and AI for secure data sharing and detection of CT images for the hospitals. Comput. Med. Imaging Graph. 87, 101812.

LaLonde, R., Bagci, U., 2018. Capsules for object segmentation. arXiv preprint arXiv:1804.04241.

Li, X., Gu, Y., Dvornek, N.C., Staib, L.H., Ventola, P., Duncan, J.S., 2020. Multi-site fMRI analysis using privacy-preserving federated learning and domain adaptation: ABIDE results. Med. Image Anal. 65, 101765.

Li, H., Liu, D., Dai, Y., Luan, T.H., Shen, X.S., 2014. Enabling efficient multi-keyword ranked search over encrypted mobile cloud data through blind storage. IEEE Trans. Emerg. Top. Comput. 3 (1), 127–138.

Li, H., Liu, D., Dai, Y., Luan, T.H., Yu, S., 2015. Personalized search over encrypted data with efficient and secure updates in mobile clouds. IEEE Trans. Emerg. Top. Comput. 6 (1), 97–109.

Liao, F., Liang, M., Li, Z., Hu, X., Song, S., 2019. Evaluate the malignancy of pulmonary nodules using the 3-d deep leaky noisy-or network. IEEE Trans. Neural Netw. Learn. Syst. 30 (11), 3484–3495.

Lu, Y., Huang, X., Dai, Y., Maharjan, S., Zhang, Y., 2019. Differentially private asynchronous federated learning for mobile edge computing in Urban informatics. IEEE Trans. Ind. Inf. 16 (3), 2134–2143.

Lu, Y., Huang, X., Dai, Y., Maharjan, S., Zhang, Y., 2020a. Blockchain and Federated Learning for Privacy-Preserved Data Sharing in Industrial IoT. IEEE Trans. Ind. Inf. 16 (6), 4177–4186. http://dx.doi.org/10.1109/TII.2019.2942190.

Lu, Y., Huang, X., Zhang, K., Maharjan, S., Zhang, Y., 2020b. Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles. IEEE Trans. Veh. Technol. 69 (4), 4298–4311.

Malekzadeh, M., Hasircioglu, B., Mital, N., Katarya, K., Ozfatura, M.E., Gündüz, D., 2021. Dopamine: Differentially private federated learning on medical data. CoRR. arXiv:2101.11693.

Pathak, Y., Shukla, P.K., Tiwari, A., Stalin, S., Singh, S., 2020. Deep transfer learning based classification model for COVID-19 disease. Irbm.

Pokhrel, S.R., Choi, J., 2020. Federated learning with blockchain for autonomous vehicles: Analysis and design challenges. IEEE Trans. Commun..

Qu, Y., Gao, L., Luan, T.H., Xiang, Y., Yu, S., Li, B., Zheng, G., 2020. Decentralized privacy using blockchain-enabled federated learning in fog computing. IEEE Internet Things J..

Sabour, S., Frosst, N., Hinton, G.E., 2017. Dynamic routing between capsules. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA. pp. 3856–3866.

Sattler, F., Wiedemann, S., Müller, K.-R., Samek, W., 2019. Robust and communication-efficient federated learning from non-iid data. IEEE Trans. Neural Netw. Learn. Syst. 31 (9), 3400–3413.

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 618–626.

Shokri, R., Shmatikov, V., 2015. Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. pp. 1310–1321.

Tang, W., Ren, J., Zhang, Y., 2018. Enabling trusted and privacy-preserving healthcare services in social media health networks. IEEE Trans. Multimed. 21 (3), 579–590.

Thwal, C.M., Thar, K., Tun, Y.L., Hong, C.S., 2021. Attention on personalized clinical decision support system: Federated learning approach. In: Unger, H., Kim, J., Kang, U., So-In, C., Du, J., Saad, W., Ha, Y., Wagner, C., Bourgeois, J., Sathitwiriyawong, C., Kwon, H., Leung, C.K. (Eds.), IEEE International Conference on Big Data and Smart Computing, BigComp 2021, Jeju Island, South Korea, January 17-20, 2021. IEEE, pp. 141–147.

Tran, N.H., Bao, W., Zomaya, A., NH, N.M., Hong, C.S., 2019. Federated learning over wireless networks: Optimization model design and analysis. In: IEEE INFOCOM 2019-IEEE Conference on Computer Communications. IEEE, pp. 1387–1395.

Xu, G., Li, H., Liu, S., Yang, K., Lin, X., 2019. Verifynet: Secure and verifiable federated learning. IEEE Trans. Inf. Forensics Secur. 15, 911–926.

Yang, K., Jia, X., Ren, K., 2014. Secure and verifiable policy update outsourcing for big data access control in the cloud. IEEE Trans. Parallel Distrib. Syst. 26 (12), 3461–3470.

Yang, W., Liu, B., Lu, C., Yu, N., 2020. Privacy preserving on updated parameters in federated learning. In: ACM TUR-C'20: ACM Turing Celebration Conference, Hefei, China, May 22-24, 2020. ACM, pp. 27–31.

Yang, D., Xu, Z., Li, W., Myronenko, A., Roth, H.R., Harmon, S.A., Xu, S., Turkbey, B., Turkbey, E., Wang, X., Zhu, W., Carrafiello, G., Patella, F., Cariati, M., Obinata, H., Mori, H., Tamura, K., An, P., Wood, B.J., Xu, D., 2021. Federated semi-supervised learning for COVID region segmentation in chest CT using multi-national data from China, Italy, Japan. Med. Image Anal. 70, 101992.

Zhang, X., Ji, S., Wang, H., Wang, T., 2017. Private, yet practical, multiparty deep learning. In: 2017 IEEE 37th International Conference on Distributed Computing Systems. ICDCS, IEEE, pp. 1442–1452.

Zhu, H., Jin, Y., 2019. Multi-objective evolutionary federated learning. IEEE Trans. Neural Netw. Learn. Syst. 31 (4), 1310–1322.