DOI: 10.6977/IJoSI.202504_9(2).0003
G. M. Fathima, L. Shakkeera, etc./Int. J. Systematic Innovation, 9(2), 26-36 (2025)

Secure mobile cloud data using federated learning and blockchain technology

G. Matheen Fathima*, L. Shakkeera, Y. Sharmasth Vali

School of Computer Science Engineering and Information Science, Presidency University, Bengaluru, Karnataka, India

*Corresponding author E-mail: matheen.20233CSE0025@presidencyuniversity.in

(Received 17 July 2024; Final version received 28 January 2025; Accepted 21 October 2024)

Abstract

In the current era, mobile cloud (MC) transactions raise concerns over the data stored in the MC. These data can be tampered with by third parties, leading to data loss and information misplacement. Such security breaches can be mitigated by implementing federated learning (FL). FL refers to a distributed data learning approach that trains data without revealing the information to the server or coordinator. It uses the current model data for training and then sends the updated model to the coordinator or server. The server collects the updated trained models from all clients and aggregates them into a single global model. This updated model is then communicated back to the clients. FL, when implemented with MC, protects user privacy, ensures efficient learning, and achieves higher accuracy compared to traditional machine learning algorithms. We propose the implementation of MC FL using blockchain, a model designed to protect user data by maintaining it on edge devices and sending the updated model to the server after training. Finally, the data-generated model will be stored in the blockchain network, preventing data tampering and providing a higher level of security and privacy for the data.

Keywords: Blockchain, Data Security and Integrity, Federated Learning, Mobile Cloud Computing

1. Introduction

Mobile cloud computing (MCC) (Noor et al., 2018) is a technology used to access the cloud environment through mobile devices. The data stored in the mobile cloud (MC) are easy to access using end devices. The MC storage service, Google App Engine (GAE) (Sharma et al., 2019), provided by Google, empowers developers to create and deploy scalable web and mobile applications. The ovarian cancer dataset is offloaded to GAE using differential privacy (DP). GAE accepts the user data and maintains the data with a high level of security and data integrity. The global women's population is being analyzed, focusing on those aged 35 – 75 who are affected by ovarian cancer, using data provided by the World Health Organization (WHO) (Reid & Bajwa, 2023). Ovarian cancer incidence is notably higher in countries such as the United States of America, the United Kingdom, Canada, and Australia.

Fig. 1 illustrates the global ovarian cancer rates sourced from the World Ovarian Cancer Coalition, which provides the statistical report for women affected by ovarian cancer in 2020, with detailed incidences and mortality cases across countries in Asia, Europe, North America, Africa, and Oceania. Similar statistics are predicted for 2040, estimating the number of ovarian cancer cases. In 2040, Asia is expected to record the highest incidence rate compared to 2020, with the mortality rate reaching approximately 175,000, which is higher than the 2020 mortality rate, which is around 110,000.

Data stored on the MC is vulnerable to security breaches, raising significant security concerns. Cloud data can be tampered with by third parties, leading to data loss and information misplacement. The objective is to protect the offloaded data in the MC and process it using federated learning (FL), where clients generate local updates (LU) based on their data. These updates are authenticated using proof of work (POW), which requires computational effort to verify the legitimacy of the update before accepting the LU. The locally

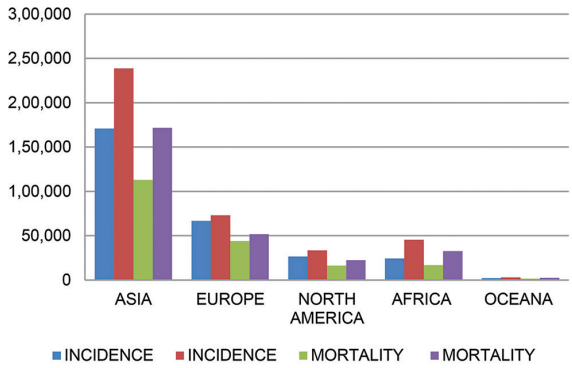

Fig. 1. Global ovarian cancer incidence and mortality rate for the years 2020 and 2040

updated models learned from the data across clients are aggregated into a global model (GM). DP enhances the privacy of individual data by adding Gaussian noise (GN) to the GM during aggregation. User data are validated after the training process, ensuring the generated update is authenticated. This process ensures the accuracy and integrity of the data used in the model. This approach aims to ensure complete access control, data privacy, and security measures for user data.

Gartner predicts that the transition to cloud-based information technology services will significantly impact enterprise spending. They anticipate that this spending, which currently stands at approximately $1.3 trillion, will rise to $1.8 trillion by 2025. Data stored in cloud environments face risks such as potential loss, privacy threats, compromised integrity, and security vulnerabilities. These concerns highlight the critical importance of robust measures to safeguard data across cloud platforms.

FL (Ray et al., 2021) is a distributed machine learning approach in which data are trained locally on smartphones or edge devices rather than being transferred to a central server. Instead of sharing raw data, only the updated model parameters are sent to the server, ensuring that sensitive information remains on the local device. This decentralized method significantly enhances data privacy by reducing the risk of data breaches, hacking, or unauthorized access. The server aggregates these local model updates from multiple devices, improving the GM without compromising individual data. This approach is highly beneficial for applications involving sensitive data, such as health care or Internet of Things (IoT) systems, where privacy and security are paramount.

The dataset used for securing is the ovarian cancer image data. The client's data, stored in the GAE, will be trained using the FL model. The training process occurs within the cloud environment, where data are trained by edge devices, and the model updates are shared with an aggregator or central server. FL provides collaborative model training without sharing the client's sensitive data, providing security and privacy for the client's data located in the MC.

The study focuses on the primary research objectives:

- This study addresses the critical security challenges in maintaining the integrity and privacy of user data stored in the MC environment. The sensitive ovarian cancer data stored in the MC is protected from unauthorized third-party access and tampering by applying DP techniques
- The DP algorithm employs the Discrete Fourier Transform to apply controlled noise to the data, concealing individual-level information while maintaining the overall statistical characteristics essential for effective data analysis. This approach guarantees the security of the offloaded data, which can then be reliably used for further processing and model training without compromising patient privacy
- This study proposes a framework that leverages FL and blockchain technology to safeguard sensitive data, such as ovarian cancer information, stored in the MC environment. FL enables collaborative model training without sharing client data, preserving privacy. The locally updated models are authenticated using POW and then aggregated into a GM, which is secured on the blockchain network
- This framework uses DP to enhance privacy and security for user data in the MC. Controlled noise is added during aggregation to conceal individual information while preserving overall statistics. The goal is to provide comprehensive access control, privacy, and security measures to ensure data integrity and confidentiality.

The MC FL using blockchain (MCFLB) framework is systematically designed to address the growing concerns surrounding data privacy and security in MC environments. By decentralizing data processing through FL and securing the aggregated model using blockchain technology, this approach employs an innovative, step-by-step methodology to resolve critical issues such as unauthorized access and data breaches. This framework depicts a systematic innovation approach by seamlessly integrating cutting-edge technologies to ensure that the model is scalable, secure, and adaptable across diverse industries, including health care, IoT, and finance. Systematic innovation lies in the identification of the problem and the application of an integrated technological solution that addresses both privacy and performance without compromising user control over sensitive data.

2. Literature Review

Offloading computational tasks and data storage from mobile devices to the MC has become a popular approach to enhancing the capabilities of resource-constrained mobile devices. This offloading technique leverages the cloud's abundant computing and storage resources, allowing mobile applications to process and store data more efficiently (Ali & Iqbal, 2022). However, the security and privacy of the data stored in the MC environment remain major concerns that must be addressed. Robust mechanisms are required to ensure integrity, confidentiality, and controlled access to the sensitive data offloaded to the cloud.

Guo et al. (2022) suggest using an ML approach along with a multiuser mobile edge computing network to tackle eavesdropping challenges. They leverage cloud access points to offload data from mobile devices, which helps reduce latency and energy consumption. To address issues related to resource allocation, bandwidth, and optimization, the researchers use an FL framework. This framework lowers the overall system costs in terms of energy and latency. However, the authors note that mobile devices have limited battery life; hence, they recommend exploring an energy-harvesting model as a potential solution.

Kairouz et al. (2021) discuss how the rapid proliferation of mobile devices and cloud computing has transformed data storage and processing, allowing mobile apps to leverage cloud resources. However, the security and privacy of data stored in the MC environment remain major concerns. Robust mechanisms are needed to protect the integrity, confidentiality, and controlled access of sensitive data offloaded to the cloud, as data breaches and unauthorized access pose significant risks to users and organizations.

Xu et al. (2020) proposed an FL framework to address data privacy and security challenges in health care. This approach allows health-care organizations to train local models on their own patient data without sharing raw, sensitive information. The locally trained models are then aggregated into a GM that can be used for predictions and insights without compromising individual privacy. This framework safeguards data confidentiality, improves model quality and accuracy by leveraging a larger dataset, and emphasizes the importance of incentivizing participation and ensuring GM precision to enhance the feasibility and effectiveness of the solution.

He et al. (2018) introduced an efficient privacy-preserving authentication protocol to secure MCC services. Their approach leveraged identity-based cryptography and a two-factor authentication mechanism to strengthen authentication and data confidentiality for MC users. The scheme employed an identity-based signature mechanism to address impersonation attacks observed in prior privacy-aware authentication solutions, thereby enhancing the overall security and resilience of the authentication process. Evaluations revealed that this enhanced privacy-aware authentication scheme incurred reduced communication overhead compared to earlier proposals, rendering it a more practical solution for safeguarding sensitive data in the MC environment.

Mothukuri et al. (2022) proposed an approach that uses FL to develop an anomaly detection and intrusion prevention system for IoT environments. By training local models on IoT devices and aggregating the learned parameters into a GM, this decentralized framework protects user privacy without requiring direct data sharing. The authors emphasize the importance of evaluating the system using live, device-specific datasets to identify both known and unknown IoT vulnerabilities, a crucial step for enhancing overall IoT security.

Su et al. (2022) proposed an FL framework for smart grid applications that leverages edge-cloud collaboration to address privacy and security challenges. The framework trains local models on IoT devices without sharing sensitive energy data, preserving user privacy. The locally trained models are then aggregated into a GM secured using blockchain technology. The authors emphasized the importance of evaluating the system using real-world data and highlighted the significance of an incentive mechanism, such as a deep reinforcement learning algorithm, to encourage participation and ensure high-quality model contributions.

Lim et al. (2020) discussed how FL offers an optimal solution for edge device training by enabling local model development while only sharing updated model parameters, thus preserving data privacy. This approach decentralizes data processing, enhancing responsiveness and reducing latency compared to cloud-based approaches that require sharing raw data and risk privacy breaches. The FL framework can effectively leverage edge resources while maintaining data privacy and security.

Zhan et al. (2022) emphasized the importance of effective incentive mechanisms to motivate active and reliable client participation in FL. Such mechanisms can incentivize participants by offering them a share of the revenue generated from their local datasets. However, the paper also highlights the challenge of designing innovative incentive schemes that accommodate the complexities of multiparty FL while ensuring security. Overcoming this challenge is crucial for developing FL frameworks that preserve data privacy and security while incentivizing widespread participation and collaboration, thus enhancing the overall effectiveness and real-world applicability of the approach.

3. Materials and Methods

3.1. MCFLB

MCFLB is a framework implemented to provide security and privacy to offloaded MC data. These data were trained using the FL approach, and the generated model was secured (Matheen Fathima et al., 2024) using a blockchain network. The process began with the following steps: (i) data preprocessing, which involved removing unwanted images and converting them into JPG format; (ii) image segmentation, which separated the training and testing images; and (iii) feature extraction, which was conducted on the ovarian images for further processing.

This research focused on securely maintaining MC data, specifically images, by local users. The objective was to safeguard the pictures stored in the cloud from tampering or data modification. ML is a centralized approach where training images use algorithms to achieve a higher level of accuracy in image data analysis. Unlike traditional ML approaches, FL is a decentralized approach that enables clients to train their data without sharing their images with a central server. FL was implemented to address this challenge by allowing local clients to train models using local data (LD) generated by the server. Subsequently, the server collected updated local models from each client and clustered them into the GM by calculating the network weights. The aggregator parameter then combines all the updated local models into a refined GM, ensuring that user data remains protected from third-party attacks.

The ovarian cancer dataset was utilized to demonstrate the effectiveness of the FL approach, specifically in preserving data privacy while improving model performance. Data from multiple institutions were kept locally, and models were trained at each location, with the results aggregated into a GM. This approach resulted in better accuracy compared to traditional centralized methods while ensuring that patient data remained private. The FL method also promoted collaboration among institutions without requiring the sharing of sensitive information, proving its scalability and potential for use in healthcare research.

The data were collected from the University of British Columbia from the Kaggle. The ovarian cancer data consists of magnetic resonance imaging scans of patients (image data). It includes the following datasets: CC (5,295), EC (6,250), HGSC (8,747), LGSC (2,720), and MC (2,491), totaling 25,503 images.

3.1.1. Dataset description

As mentioned in Table 1, the model we proposed and analyzed uses the ovarian cancer datasets provided by the WHO (Reid & Bajwa, 2023). Ovarian carcinoma is a type of cancer that occurs in the ovaries of the female reproductive system. There are five common subtypes of ovarian cancer: Clear cell carcinoma (CC), endometrioid carcinoma (EC), high-grade serous carcinoma (HGSC), low-grade serous carcinoma (LGSC), and mucinous carcinoma (MC). This cancer is influenced by genetic mutations of the BRCA gene and typically affects women later in life, usually after the age of 50. It is often associated with an improper balance in the life cycle and a lack of physical exercise. The dataset was split, with 80% used for training and 20% used for testing.

Table 1. Image datasets

Ovarian cancer	Count
CC	5,295
EC	6,250
HGSC	8,747
LGSC	2,720
MC	2,491
Total	25,503

Abbreviations: CC: Clear cell carcinoma; EC: Endometrioid carcinoma; HGSC: High-grade serous carcinoma; LGSC: Low-grade serous carcinoma; MC: Mucinous carcinoma.

3.1.2. Data preprocessing

During the preprocessing stage, where the raw data were cleaned and structured for processing. The unwanted or null images were removed from the datasets, and the images were converted into the required format for further processing. The raw images were transformed into red, green, and blue (RGB) image formats. In addition, the images were resized to fixed pixel dimensions ranging from 0 to 255 for further image processing. The dataset utilized consists of ovarian cancer images.

3.1.3. Image segmentation

The datasets contain images classified into two categories: Whole slide images and tissue microarray. Whole slide images are captured at 20× magnification and tend to be large in size. Conversely, tissue microarrays are smaller (approximately 4,000×4,000 pixels in dimension) but are captured at a higher magnification of 40×.

The ovarian cancer subtypes (CC, EC, HGSC, LGSC, and MC) were classified, with 70% of the images used for training and 30% for testing. This means that 17,500 images were used for training, and 7,500 were used for testing.

A normalization technique was applied to the images to maintain consistency in feature extraction

and lighting, ensuring that all the images were within a boundary range of [0,1].

$$k_i = a^i - \frac{(a)}{(a)} - min(a) \quad (1)$$

where $a_i = a_1, a_2, a_3, \ldots\ldots a_n$, k_i is the i^{th} normalized data, min(a) is the minimum value in the datasets, and max(a) is the maximum value in the datasets.

3.1.4. Feature extraction

The pixel values of the ovarian images ranged from 0 to 255 in RGB color code format. These images underwent preprocessing and feature extraction to facilitate the classification of ovarian cancer subtypes such as CC, EC, HGSC, LGSC, and MC. The preprocessing involved several steps, including normalization (scaling pixel values between 0 and 1), noise reduction through filters such as Gaussian blur or median filtering, contrast enhancement for improved visibility, and resizing for uniform image dimensions.

3.2. Working on MCFLB Framework for Image Datasets

Data security presents a critical challenge for cloud users, particularly concerning unauthorized data tampering and modification. Users often lack full control over their data in MC environments. To address this issue and enhance user control in the cloud domain, accessing user data for training through FL on the cloud platform is essential. FL is a technique in which data are trained on devices without being shared with a central server, thus preventing data loss, privacy breaches, and tampering.

FL safeguards user privacy by securely storing data on individual devices and utilizing convolutional neural network (CNN) (Chauhan et al., 2018) training to produce models. These models, created by participating clients, are consolidated by a central server to form an updated GM, which is subsequently shared with clients for further training until the desired accuracy level is reached. To reinforce security, the updated GM is protected through a blockchain network, with model storage distributed across network nodes, ensuring data security within the MC platform.

As shown in Fig. 2, the input dataset consists of ovarian cancer images, which were then transferred to the MC and protected using a DP offloading algorithm. Within the cloud, the cancer dataset is securely stored. However, directly utilizing MC datasets for training in an FL setting is not feasible. Instead, the Google Kubernetes Engine serves as an intermediary for training the images. In Google Kubernetes Engine, an application is developed, with Docker encapsulating the necessary requirements for running the application or executing tools. Once the application is constructed, it must be deployed on the cloud platform using Kubernetes Orchestration (KO).

In FL, each user held distinct ovarian cancer datasets, which were trained locally. Updates following training were gathered by the server or central coordinator. The trained ovarian images underwent classification using the CNN algorithm, which categorized the images into subtypes of ovarian cancer, including CC, EC, HGSC, LGSC, and MC.

The CNN model effectively captured image parameters from each image and constructed a model

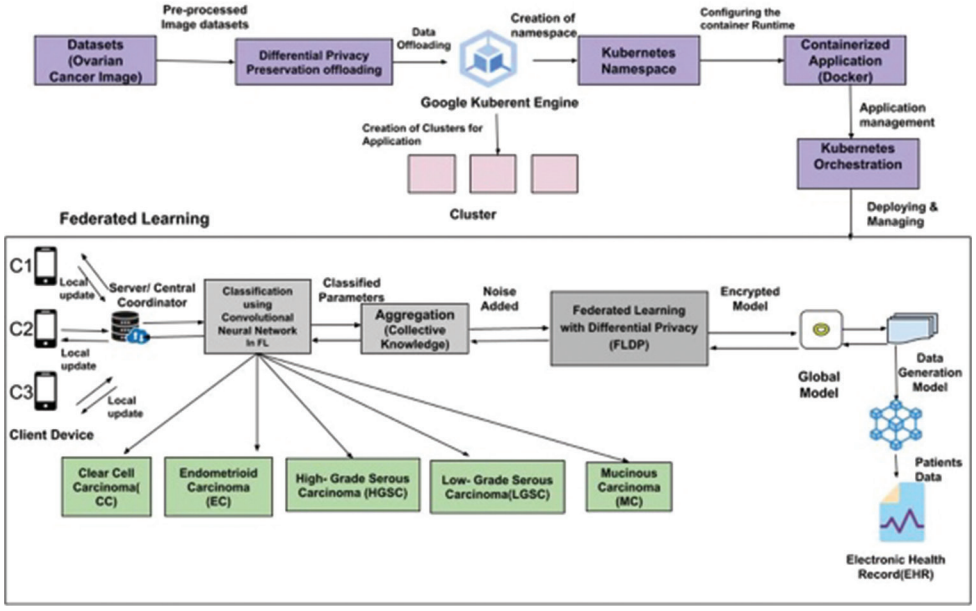

Fig. 2. Mobile cloud federated learning using blockchain framework for image datasets

through aggregation, where the image was learned by the client or user without data sharing with the central server. This aggregation process continued until the model was accurately learned from the datasets. To safeguard against data tampering and modification, the model generated was fortified by incorporating noise using FL with differential privacy (FLDP). This is an encrypted model, which was then integrated into a GM. These GM-generated models are tailored to the requisite datasets, resembling data-generated models. These data-generated models securely housed patient cancer datasets, serving as Electronic Health Records (EHR) (Wang & Zhou, 2021) within the blockchain network.

3.2.1. MC data offloading using DP preservation offloading

The ovarian cancer data were offloaded to GAE, a serverless platform that executes and offloads large datasets with certain storage limits, providing a free-of-cost service. The normalized data was offloaded to the GAE service, where data preprocessing and image segmentation were performed. The goal was to provide security for the offloaded data using the DP preservation offloading (Zhao et al., 2024) algorithm.

Algorithm 1: DP preservation offloading

1: The cancer image (I) was divided into "n" equal parts.

T: $k\rightarrow,k_1\times k_2\times k_3\ldots.\times k_n$, where i = 1 to n.

2: Discrete Fourier transform was applied to find the original image values, which were then split into real and imaginary components of the image values.

$I^T\rightarrow I_R, I_I, F_{\leftrightarrow} F_R, F_I$

where,

I_R is the image real value,

I_I is the image imaginary value,

F_R is the filter real image,

F_I is the filter imaginary image.

Offloading the Datasets

3: The images were processed in "n" parts on the cloud server.

$\emptyset{:}K\rightarrow C_1\times C_2\times C_3\times\ldots\ldots C_k$

where $C_1,C_2,C_3,\ldots\ldots C_n$ are the cloud servers.

$I_i\leftrightarrow C_n=\varphi(I_i,x)=(l_1,l_2,l_3,\ldots l_n)$

where i = 1 to n, $n\leq k$, *and* $l_i = T$ I was assigned to the server i, $\forall_i = 1,2,3,\ldots.k,\forall_x\in(r,i)$.

4: At each server C_i, the image part I_i was multiplied by F.

P: k $\times k\rightarrow k$

$I_{i.x}\times Fx\leftrightarrow Ri.x = P'(I_{i.x})\ \forall x\in$(r,i)

The DP preservation offloading algorithm divided the image into "n" equal parts. The discrete Fourier transform technique was applied to determine the frequency of the input image. The image was classified into real values "r" and imaginary values "i." The filter "F" was applied to enhance the quality of image processing. The cloud server distributed the image across different servers to process the request on their end. At the server end, the image I_i was multiplied by F to process a clearer pixel value.

3.2.2. Namespace creation and orchestration

a) Kubernetes namespace

Kubernetes, also known as Kube or K8s, is an open-source application used to deliver, scale, and customize container applications. The application is developed using Kubernetes and can be executed on any available nodes within the network. It can schedule instances of the application based on user requirements, manage resource allocation, and prevent central processing unit and memory overutilization.

Kubernetes is managed and organized into namespaces within the cluster. These namespaces consist of pods, services, and volumes, which are integral to the container application. Pods contain one or more containers that include storage and networking capabilities, allowing the management of server nodes within the same containerized environment. Services are responsible for running the pods for the application, similar to running a browser application on a mobile phone for user access. Volumes act as storage files managed by pods.

Kubernetes were used to create a containerized application for executing FL. Pods were utilized to create nodes within the FL network and provide a hosting environment. Services were scheduled for the execution of the containerized application. Volumes were used to store the ovarian cancer image datasets. Namespace creation in Kubernetes was defined by names that start with lowercase letters, may include numerals, and do not exceed 64 characters.

Command

$ kubectl create namespace <namespace_name>

$ kubectl create namespace apple

namespace/apple created.

b) KO

KO (Carmen et. al., 2023) refers to the automated management of container applications in a group of clusters. KO is scheduled based on the capacity of nodes or individual machines, while configuration files are installed according to the infrastructure of the application. It balances the workload memory usage and scales based on the requirements. In addition, KO manages sensitive data using secrets.

i) Classification using CNNs in the FL environment

The images were deployed within the

containerized application in the FL environment. Clients were created in the FL and were assigned specific images for training. The model located on the server was distributed to all the clients placed in the nodes of the network.

The data were preprocessed, segmented, and augmented to enhance the efficiency of the training data. For classification, 70% of the data was allocated for training and 30% for testing. DenseNet-201 was utilized for classification tasks on large datasets, yielding higher accuracy levels. The images were initialized with pre-trained weights specific to the dataset, enhancing both efficiency and accuracy.

Fig. 3 illustrates the evaluation of CNN's performance based on metrics such as accuracy, precision, recall, and F1-score, which measure its effectiveness. In Dense-Net, batch normalization and ReLU activation functions were applied after each convolutional layer. This combination stabilized the training process and accelerated convergence by normalizing the activations of each layer, reducing internal covariate shifts, and enhancing the network's robustness during training.

ii) Aggregation model in FL

First, an initial model was shared with the client by the central server for processing the LD placed on the client's device. The LD was then trained using the model on the client's device, generating LU from each client. The LU obtained from each client was shared individually with the central server. The aggregation process combined all the LUs obtained from the LD to form a GM. The GM was then circulated to all the clients, and the process was repeated until the desired accuracy level was achieved from LD.

Fig. 4 shows how the preprocessed dataset is distributed to the clients, with each client holding a portion of the input dataset. These images were trained using a model initially provided by the central server. The input ovarian cancer data were not directly shared with the central server; instead, the model was given to the clients to train the data on their devices. The LD was trained using the model, generating an LU by efficiently learning from the input data stored on the client's device. The model generated by each client after training was shared with the central server. The aggregation process combined the LUs from all the clients into a GM, which was then distributed back to the clients. This process continued until the desired accuracy level was achieved for the LD.

4. FL with DP

FL was utilized to secure the images using an aggregator parameter to obtain the overall average of the local models. FLDP (Wei et al., 2020) was

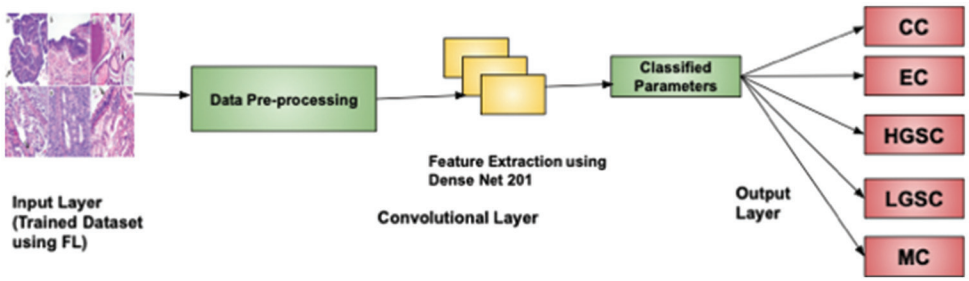

Fig. 3. Classification of ovarian cancer using convolutional neural networks
Abbreviations: CC: Clear cell carcinoma; EC: Endometrioid carcinoma; Fl: Federated learning; HGSC: High-grade serous carcinoma; LGSC: Low-grade serous carcinoma; MC: Mucinous carcinoma

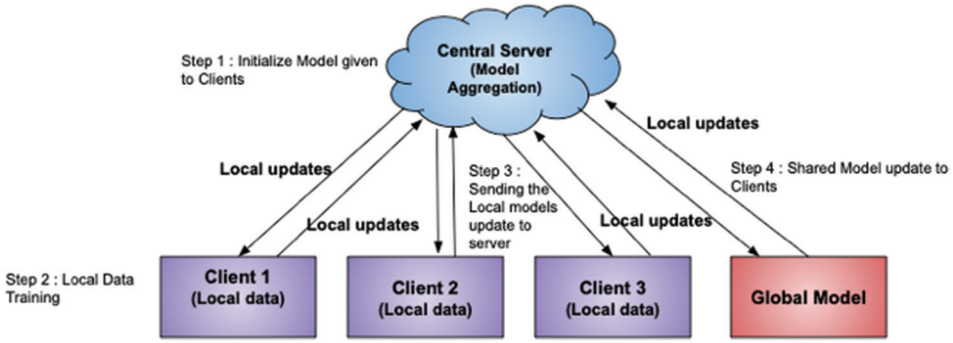

Fig. 4. Aggregation model in federated learning

employed to add noise to protect the ovarian images from potential attackers. The GM incorporated the real image data with added noise. In the model of distributed DP for FL, each client participating in the FL process only needed to introduce a small amount of noise. The noise used in DP was GN, which concealed sensitive data by adding noise, thereby ensuring privacy for the data. GN not only preserves privacy but also enhances data accuracy. This approach ensures that the aggregation performed by the central server satisfies central DP. However, since the noise added by each client was minimal, it provided guaranteed security to the LD.

Algorithm 2: FLDP

Step 1: Q_0 *was initialized*

For p ϵ [P], a random sample S_p *was taken* with a sampling probability of S/N.

Step 2: The gradient was calculated for each i ϵ S_p, where $g_t(x_i) \leftarrow \nabla Q_t S(Q_t, x_i)$,

$$g_t(x_i) \leftarrow g_t(x_i) / max(1, \frac{\| g_t(x_i) \| 2}{c})$$

Step 3: The noise was added

$$g_t \leftarrow \frac{1}{S}(\sum\nolimits_p g_t(x_i) + N(0, \sigma^2 c^2 I))$$

$Q_{t+1} \leftarrow Q_t \gamma_t g_t$

Descent, $Q_{t+1} \leftarrow Q_t \gamma_t g_t$

Step 4: Output was Q_t, and the overall privacy cost (ϵ,β) was computed.

After aggregating the classified parameters, noise was added to the generated model to prevent data tampering using the FLDP algorithm. This noise was added to the frequency of data, resulting in an encrypted model. Based on the input data given, a GM was generated by learning from the input weights calculated from the image, which was a data-generated model, without directly sharing the image data with the server or coordinator.

The data-generated models were stored on the nodes of the blockchain network, providing a dual-layer data protection mechanism. The model was hashed using a secure hashing algorithm encryption within the network. The blockchain comprised nodes that accepted this model based on a consensus algorithm. Subsequently, data stored in the network was maintained as an EHR. These EHRs could be shared by patients with their healthcare providers during check-ups or consultations.

5. Results and Discussions

This section discusses the performance of users in terms of communication during each iteration and data distribution. The existing TB-PRE system (Zhang et al., 2017) was compared with the proposed MCFLB system. Storage and computational performance were measured in terms of bytes and milliseconds (ms). Data were stored and retrieved using a secret key, and the time taken to process the data was recorded. The MCFLB system outperforms TB-PRE in terms of speed and provides more secure transactions.

The parameters in Table 2 were processed on a MacBook Air with a 1.8 GHz Dual-Core Intel Core i5 processor and 8 GB of memory. The results demonstrate that the proposed system is comparable to the existing one.

The communication of data processing and distribution were compared with the existing data files N_k and the new data files N'. The data were calculated based on parameters such as upload, modification, retrieval, deletion, permission, and distribution. The performance of the data operations was evaluated by comparing the existing file with the new file being added.

Table 3 defines the user-uploaded data file for processing and the time taken for calculations. It also presents the average time required to retrieve the offloaded data from the cloud to the user. The overall communication performance between the mobile user and the cloud is efficient, effectively handling the overhead communication in the cloud environment.

This section provides a comprehensive analysis of user performance, focusing on communication

Table 2. Storage and computation comparisons between TB-PRE and mobile cloud federated learning using blockchain (MCFLB)

Performance metrics	Parameters	TB-PRE	MCFLB
Storage	R_k	33 bytes	20 bytes
	C_1	193 bytes	150 bytes
	C_2	416 bytes	230 bytes
Computation	Enc	4.64 ms	2.75 ms
	Re-Enc	16.39 ms	16.20 ms
	Dec	18.33 ms	17.92 ms
	C_1C_2	3.99 ms	2.84 ms

Abbreviations: Dec: Decryption; Enc: Encryption; Re-Enc: Re-encryption.

Table 3. Communication in terms of bytes for new data files (N') and existing data files (N_k)

Operations	Communication overhead
Upload	220
Retrieval	28 $N'\log N_k$+5N'+180
Modification	28 $N'\log N_k$+5N'+190
Deletion	28 $N'\log N_k$+5N'+190
Permission	<220
Distribution	28 $N'\log N_k$+5N'+420

during each iteration and data distribution between the existing TB-PRE system and the proposed MCFLB system. Key performance metrics, such as storage, computation, and communication overhead, were thoroughly examined to highlight the advantages of MCFLB over TB-PRE.

5.1.1. Storage and computation

The storage and computation performances of both systems were analyzed in terms of bytes and milliseconds, respectively. The storage metrics include three parameters: R_k, C_1, and C_2. The proposed MCFLB system demonstrated a significant reduction in storage requirements across all parameters. Specifically, the MCFLB system used 20 bytes for R_k, 150 bytes for C_1, and 230 bytes for C_2, compared to the TB-PRE system, which required 33 bytes, 193 bytes, and 416 bytes, respectively. This reduction in storage not only optimized the system's efficiency but also enabled faster data processing.

In terms of computation, three primary operations were analyzed: Encryption (Enc), re-encryption (Re-Enc), and decryption (Dec). In addition, the computation times for C_1 and C_2 were also measured. The MCFLB system showed improved performance, with Enc taking 2.75 ms compared to 4.64 ms in TB-PRE. Re-Enc and Dec times were also reduced, with MCFLB showing times of 16.20 ms and 17.92 ms, respectively, compared to TB-PRE's 16.39 ms and 18.33 ms. The computation for C_1 and C_2 in MCFLB was similarly optimized, resulting in a faster and more secure data processing environment.

5.1.2. Communication overhead

The communication of data processing and distribution between the existing data files (N_k) and new data files (N′) were also compared. The operations examined include upload, modification, retrieval, deletion, permission management, and distribution. The communication overhead for these operations was calculated based on parameters involving N_k and N′. For instance, the upload operation required 220 bytes of communication overhead, whereas retrieval, modification, and deletion operations involved a more complex formula: 28N′log Nk+5N′+19028N′loN_k+5N′+190 bytes. Permission management incurred <220 bytes of overhead, while distribution had the highest overhead at 28N′log N_k+5N′+42028N′logN_k+5N′+420 bytes. These results indicate that the MCFLB system is not only faster but also more efficient in managing communication overhead, especially in cloud environments. The reduction in storage and computation requirements translates to lower latency and faster data processing, making MCFLB a superior choice for secure transactions.

When comparing these results with previous studies, it becomes evident that the MCFLB system represents a significant advancement in secure data management. Earlier research focused on reducing computational overhead in proxy Re-Enc systems, but these methods often resulted in higher storage requirements or compromised security. The MCFLB system addresses these shortcomings by balancing storage efficiency with computational speed, providing a more robust solution without sacrificing security. Moreover, the reduction in communication overhead observed in MCFLB aligns with recent trends in cloud computing, where minimizing latency and optimizing resource use are critical. Studies have shown that reducing communication overhead is essential for improving the overall performance of cloud-based systems, especially in mobile environments where bandwidth and processing power are limited. The MCFLB system's ability to streamline these processes makes it a valuable contribution to the field.

The proposed FL and blockchain-based model addresses key data privacy concerns, exposes vulnerabilities in centralized systems, and prompts organizations to rethink their data management strategies. By enabling decentralized machine learning, it reduces the risk of data breaches while maintaining efficiency. The integration of blockchain ensures data integrity, boosting trust and regulatory compliance. This adaptable framework supports industries such as health care, finance, and IoT, allowing secure collaboration without compromising sensitive data. The MCFLB model drives innovation by prioritizing user trust, data security, and adaptability to evolving business challenges.

6. Conclusion

MCFLB framework represents a significant advancement in securing MC transactions by ensuring user data security while maintaining efficient and accurate machine learning processes. By leveraging FL and blockchain technology, MCFLB offers robust protection against data tampering and privacy breaches. The integration of ovarian cancer image datasets demonstrates the model's practical application and effectiveness, maintaining data on edge devices and reducing vulnerabilities associated with centralized storage. As data security challenges continue to evolve, MCFLB promotes a future where privacy and security are paramount in MC environments. This model not only addresses current security concerns but also sets a new standard for future developments in secure and private

MC transactions, thereby enhancing trust in MC technologies.

Acknowledgment

The authors would like to express their sincere gratitude to Presidency University, Bengaluru, for providing all the necessary facilities.

References

Ali, A., & Iqbal, M.M. (2022). A cost and energy efficient task scheduling technique to offload microservices based applications in mobile cloud computing. *IEEE Access*, 10, 46633–46651. https://doi.org/10.1109/access.2022.3170918

Carmen, C. (2023). Kubernetes scheduling: Taxonomy, ongoing issues and challenges. *ACM Computing Surveys*, 55(7), 138. https://doi.org/10.1145/3539606

Chauhan, R., Ghanshala, K.K., & Joshi, R.C. (2018). Convolutional Neural Network (CNN) for Image Detection and Recognition. In: *2018 First International Conference on Secure Cyber Computing and Communication* (*ICSCCC*). IEEE, Jalandhar, India, p278–282. https://doi.org/10.1109/ICSCCC.2018.8703316

Guo, Y., Zhao, R., Lai, S., Fan, L., Lei, X., & Karagiannidis, G.K. (2022). Distributed machine learning for multiuser mobile edge computing systems. *IEEE Journal of Selected Topics in Signal Processing*, 16(3), 460–473. https://doi.org/10.1109/JSTSP.2022.3140660

He, D., Kumar, N., Khan, M.K., Wang, L., & Shen, J. (2018). Efficient privacy-aware authentication scheme for mobile cloud computing services. *IEEE Systems Journal*, 12, 1621–1631. https://doi.org/10.1109/JSYST.2016.2633809

Kairouz, P., Yu, H., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R.G.L., Rouayheb, S.E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P.B., Gruteser, M., & Zhao, S. (2021). Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14, 1–210. https://doi.org/10.1561/2200000083

Lim, W.Y.B., Luong, N.C., Hoang, D.T., Jiao, Y., Liang, Y.C., Yang, Q., Niyato, D., & Miao,C. (2020). Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(2), 2031–2063. https://doi.org/10.1109/COMST.2020.2986024

Matheen Fathima, G., Shakkeera, L., & Sharmasth Vali, Y. (2024). Secure data transactions in mobile cloud computing using FAAS. *International Journal on Recent and Innovation Trends in Computing and Communication*, 12(1), 299–305.

Mothukuri, V., Khare, P., Parizi, R.M., Pouriyeh, S., Dehgantanha, A., & Srivastave, G. (2022). Federated-learning-based anomaly detection for IoT security attacks. *IEEE Internet of Things Journal*, 9(4), 2545–2554. https://doi.org/10.1109/jiot.2021.3077803

Noor, T.H., Zeadally, S., Alfazi, A., & Sheng, Q.Z. (2018). Mobile cloud computing: Challenges and future research directions. *Journal of Network and Computer Applications*, 115, 70–85. https://doi.org/10.1016/j.jnca.2018.04.018

Ray, N.K., Puthal, D., & Ghai, D. (2021). Federated learning. *IEEE Consumer Electronics Magazine*, 10(6), 106–107. https://doi.org/10.1109/MCE.2021.3094778

Reid, F., & Bajwa, A. (2023). *World the World Ovarian Cancer Coalition Atlas 2023*. World Ovarian Cancer Coalition, Toronto.

Sharma, D., Shukla, R., Giri, A.K., & Kumar, S. (2019). A Brief Review on Search Engine Optimization. In: *9th International Conference on Cloud Computing, Data Science & Engineering* (*Confluence*). Noida, India, p687–692.

Su, Z., Wang, Y., Luan, T.H., Zhang, N., Li, F., Chen, T., & Cao, H. (2022). Secure and efficient federated learning for smart grid with edge-cloud collaboration. *IEEE Transactions on Industrial Informatics*, 18(2), 1333–1344. https://doi.org/10.1109/TII.2021.3095506

Wang, H., & Zhou, R. (2021). The Application of Blockchain to Electronic Health Record Systems: A Review. In: *2021 International Conference on Information Technology and Biomedical Engineering* (*ICITBE*). Nanchang, China, p397–401.

Wei, K., Li, J., Ding, M., Ma, C., Yang, H.H., & Farokhi, F. (2020). Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15, 3454–3469. https://doi.org/10.1109/TIFS.2020.2988575

Xu, J., Glicksberg, B.S., Su, C., Walker, P., Bian, J., & Wang, F. (2020). Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, 5(1), 1–19. https://doi.org/10.1007/s41666-020-00082-4

Zhan, Y., Zhang, J., Hong, Z., Wu, L., Li, P., & Guo, S. (2022). A survey of incentive mechanism design for federated learning. *IEEE Transactions on Emerging Topics in Computing*, 10(2), 1035–1044. https://doi.org/10.1109/TETC.2021.3063517

Zhang, J., Zhang, Z., & Guo H. (2017). Towards secure data distribution systems in mobile cloud computing. *IEEE Transactions on Mobile Computing*, 16(11), 3222–3235. https://doi.org/10.1109/TMC.2017.2687931

Zhao, P., Yang, Z., & Zhang, G. (2024). Personalized and differential privacy-aware video stream offloading in mobile edge computing. *IEEE Transactions on Cloud Computing*, 12(1), 347–358. https://doi.org/10.1109/TCC.2024.3362355

AUTHOR BIOGRAPHIES

Matheen Fathima G received her M.Tech degree from B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, India, in 2022. She is currently pursuing her Ph.D. in Computer Science and Engineering at Presidency University, Bengaluru, India. Her research interests include mobile cloud computing, blockchain technology, and Internet of Things (IoT).

Shakkeera L received her Ph.D. degree from B.S. Abdur Rahman Crescent Institute of Science and Technology, Anna University, Chennai, India, in 2018. She is currently a Professor & Associate Dean in the School of Computer Science and Engineering & Information Science at Presidency University, Bengaluru, India. Her research interests include mobile cloud computing, machine learning, IoT, mobile *ad hoc* networks (MANET), information security, and data analytics.

Sharmasth Vali. Y received his Ph.D. degree from B.S. Abdur Rahman Crescent Institute of Science and Technology, Anna University, Chennai. He is currently an Assistant Professor (Selection Grade) in the School of Computer Science and Engineering & Information Science at Presidency University, Bengaluru, India. His research interests include cyber security, ethical hacking, wireless networks, cryptography, network security, MANET, and networks.