





Privacy Preservation of Medical Images Through Synthesized Federated Learning Framework in Healthcare

Se-Jung Lim¹, Prabakaran G.², Kiruthika Mani^{3*}

¹ School of Electrical and Computer Engineering, Yeosu Campus, Chonnam National University, Yeosu-si 59626, Republic of Korea

² Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R & D Institute of Science and Technology, Chennai 600062, India

³ Department of Computational Intelligence, School of Computing, SRM Institute of Science and Technology, Chennai 603203, India

Corresponding Author Email: kiruthikatspsg@gmail.com

Copyright: ©2025 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ts.420424>

ABSTRACT

Received: 7 June 2024

Revised: 7 January 2025

Accepted: 12 June 2025

Available online: 14 August 2025

Keywords:

federated learning, homomorphic encryption, paillier cryptosystem, privacy preserving technique, securing medical images, ResNet50, DenseNet121

The correctness of AI depends on the correctness of the data. Although AI has brought numerous developments in the healthcare industry, we can't deny the fact that there are security breaches as well. This synthesized framework is proposed as a solution for providing security and privacy. Medical images of Alzheimer's disease are considered in this study. Medical images occupy huge space and also consume more bandwidth while being transmitted. A distributed learning model called federated learning is employed that allows the images to reside at any hospital. Training is performed at the client end itself and only the model parameters are shared to the server. It preserves privacy using Partially Homomorphic Encryption and the encrypted images are used for training. This proposed model uses FedAvg for model aggregation. Two CNN architectures are considered here-ResNet50 and DenseNet121. The experimental results show that the DenseNet121 model gives more accuracy than ResNet50 for the encrypted image dataset.

1. INTRODUCTION

Artificial Intelligence (AI) is playing a vital role in almost all the day-to-day activities. Its presence has become inevitable in various fields including Healthcare. AI is revolutionizing the healthcare industry in multiple aspects [1] like decision making, disease prediction, surgical procedure, clinical data storage, rehabilitation process and so forth. Huge volume of data is handled by healthcare industry and these data can be patient's electronic health record, supply chain data, clinical trial data, administrative data, medical imaging data, genomic data, research data, and many more [2]. Preserving privacy and security of data is the basic and essential requirement in healthcare. Medical centers preserve data for multiple reasons including safeguarding patient privacy, for preventing medical identity theft, for protecting against cyber and insider attacks, for protecting Intellectual property rights, etc. [3]. Healthcare centers must also comply with regulations and standards such as HIPAA (Health Insurance Portability and Accountability Act) and GDPR (General Data Protection Regulation), failing which may result in legal penalties and downfall of reputation.

Performance of AI depends on the quality and quantity of training data, training model selection, feature engineering, and tuning of hyperparameters. The conventional model usually works by gathering all raw data at one central server and training happens over the collected data. The new learning

method, called Federated Learning (FL) is a distributed machine learning model that works by keeping the raw data in the same machine where it is archived [4]. Model training happens locally on each of the edge device, and only the model updates are shared with a central server or a central coordinator. Since raw data is not transferred through communication channels, there is less threat to the data. FL is more suitable for the healthcare industry since a huge volume of data is handled at each client site and due to the high demand for security and privacy.

The proposed work considers the medical images as they are large in size and they play a key role in disease diagnosis, treatment planning, monitoring disease progression, telemedicine and remote consultations. Medical images should be shielded from unwanted access since it is extremely private and sensitive. Securing medical images and preserving their privacy are crucial; otherwise, unauthorized disclosure or tampering may lead to incorrect diagnosis and treatment. In this framework, the medical images are encrypted using Partially Homomorphic encryption (PHE) [5, 6] at the client end. Although there are many methods for preserving privacy like split learning, differential privacy, etc., they are not used in this work due to their limitations. Differential Privacy (DP) tends to add noise to the model updates in order to protect privacy but there is a risk of degrading the model accuracy, especially in medical industry. Split Learning (SL) involves intermediate exchange of model updates that happens with the

peers, which can lead to inference attack. So, PHE is used for protecting the medical images. The encrypted images are trained using two different Convolutional Neural Network (CNN) models - ResNet50 and DenseNet121 for comparison. After training, the weights and gradients of local model are communicated to the central server. The central server aggregates the model updates from multiple clients using Federated Averaging algorithm (FedAvg) [7]. When integrating PHE with FedAvg, privacy preservation is efficient because of the optimized model convergence and also ensures robustness of the model in heterogenous environment. The global update is then forwarded to clients and the clients initiate the training process again with the new model weights and gradients. The local and global updates and training process are iterated as many times as required in order to achieve accuracy. For this study, the Alzheimer disease classification dataset is considered.

This framework is proposed to secure medical images from adversarial attacks and other attacks. Medical images are huge in size and it is difficult to transfer these images to a central location for training. When huge volume of data is trained in a central server, the process may be slower, resource intensive and more vulnerable to security threats. The proposed model addresses this issue by deploying Federated Learning. Also, the model uses encryption and secure aggregation techniques to preserve the privacy of the medical data, thereby complying the regulations and standards of healthcare industry.

Organization of the paper can be summarized as follows. Section 1 addresses the security challenges of medical data and the motivation of the proposed work. It introduces the proposed framework and briefs the methodologies used. Section 2 presents the background of important concepts used in the framework like Federated Learning, Partially Homomorphic Encryption and FedAvg aggregation for medical image modelling. Section 3 realizes the prevailing research works on privacy preserving healthcare models. Section 4 explains the proposed framework with the deep learning model employed for encrypted medical images. Section 5 evaluates the proposed architecture against few performance metrics and discusses the results. Section 6 reviews the significance of the work done by discussing the framework's advantages and possible area of improvement.

2. BACKGROUND

2.1 Federated learning

Federated Learning (FL) eliminates the need for data to be in a centralized setting by using a decentralized, distributed, and cooperative approach to machine learning. FL gives the data, the sophistication of being in the same machine where it is stored. This allows a huge amount of data to be processed without bothering about the communication cost. The federation model consists of multiple clients and a server, also known as an aggregator [8, 9]. Every client handles its dataset individually without sharing it to the Server. The server, after connecting to the clients, sends the initial global model for training. Clients train the model locally using the local data stored in the local machine and share only the model updates, i.e., gradients and weights, to the central server. All clients' local updates are sent to the server, which aggregates them into a global model, as shown in Figure 1. This global model is communicated with the clients and these steps are iterated a

few numbers of times until the model accuracy and expected performance are reached.

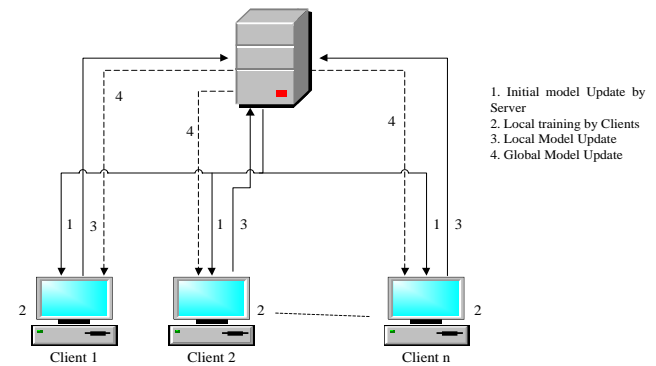


Figure 1. Overview of federated learning (FL) framework

FL can be categorized to Centralized or fully decentralized architecture based on the underlying network topology [10]. In the centralized architecture, there will be a central server that is responsible for collecting local updates from clients and share the global update back to the client. In decentralized architecture, there are no central servers, instead, each peer depends on other peers for model updates. The data partitioning in FL is of three types – Horizontal, Vertical and Transfer Learning [11]. In horizontal data partitioning, the features of the data are the same across all the clients and only the number of samples vary across the machines. Each client holds subset of samples. In vertical data partitioning, the features of the dataset are shared among the clients but the number of samples remain the same. In transfer model, the knowledge of the trained model over a particular domain of labelled data is transferred to a target domain.

The next criteria to be considered is the machine learning model to be used in a FL system. The models can be either homogenous or heterogenous. In homogenous model, all the participating clients use the same machine learning model for training the raw data and the server finally aggregates the gradients whereas in heterogenous model, each client may train the data with different algorithms and the server uses an ensemble technique to choose the accurate model.

The devices on which the data are available determines the FL to be Cross-silo FL and Cross-device FL. When the same type of devices is used in client environment and the number of devices is below 200, then it is Cross-silo FL. In Cross-device FL, the devices vary from mobile devices to smart phone to IoT devices and it is completely scalable.

After devising all these classifications, the most important one is about the aggregation technique used by the Server to combine the updates from the models sent by the Clients and proposes a single global model. There is multiple aggregation techniques used in FL as FedAvg, Scaffold, Adaptive Federated Optimization, FedMA, FedBoost, FedProx, FedPer, Weighted Aggregation, etc. [12, 13]. Of these aggregation algorithms, FedAvg is quite popular and the most commonly used algorithm. In order to update the global model appropriately, the server averages the local updates.

2.2 Partially homomorphic encryption

Homomorphic encryption (HE) is a type of public-key ciphering technique that converts a plaintext to ciphertext, ensuring confidentiality. HE allows the ciphertext to be

processed directly as if it were in its original form. Mathematical operations, like addition and multiplication, are directly performed on the ciphertext. It is categorized into Partially Homomorphic Encryption (PHE), Somewhat Homomorphic Encryption (SHE), and Fully Homomorphic Encryption (FHE) [14]. The reason behind choosing PHE than FHE is that PHE reduces the computational cost significantly, and tries to offer a balanced trade-off between efficiency and security. Efficiency is improved in PHE as it performs selective encryption on the medical data, which results in less computation over encrypted gradients. Also, this requires less extensive decryption during model updates.

In this context, HE is more useful as it helps in protecting regulatory compliance. Also, HE protects patient privacy by encrypting medical images (e.g., MRI scans, X-rays, CT scans, ultrasounds, etc.). Partially homomorphic encryption makes it possible to process the encrypted data more effectively. There's no need to decrypt the data first, which can take a while, because the mathematical operations can be done immediately on the ciphertext. This is especially helpful for handling big datasets involving images.

Algorithm 1: Paillier Cryptosystem

Key Generation:

- 1 Choose the private key pair (x, y) such that
 - 2 x, y are large primes
 - 3 $\text{GCD}(xy, (x-1)(y-1))=1$
 - 4 Derive public key pair (a, b) as
 - 5 Compute $a=x*y$ and $\ell=\text{LCM}(x-1, y-1)$
 - 6 Select a random integer b, such that
 - 7 $b \in \mathbb{Z}_a^2$ and $\text{GCD}(a, L(b^{\ell \bmod a^2}))=1$, where $L(i)=(i-1)/a$ for every i in subgroup \mathbb{Z}_a^2
-

Encryption: (Done with Public Key pair (a, b))

- 8 $C=\text{Enc}(m)=b^{m \cdot r^a} \pmod{a^2}$
- where, C–Cipher text, m–plain text, r–random number
-

Decryption: (Done with Private Key pair (x, y))

- 9 $m=\text{Dec}(C)=(L(C^{\ell} \bmod a^2)/L(b^{\ell} \bmod a^2)) \bmod a$
- where, C–Cipher text, m–plain text, r–random number
-

PHE method yields a valid result even after an arbitrary endless number of ciphertext additions or multiplications but not both. Since arbitrary number of addition or multiplication operations are allowed in PHE on encrypted data, there is no limitations on depth or complexity. Of the three homomorphic encryption methods, PHE method is computationally efficient and have lower complexity than FHE [15]. It provides the most capability and versatility, enabling a broad range of image processing operations to be carried out on encrypted images. PHE schemes may have simpler key generation, encryption, and decryption processes compared to FHE. The scheme used here is the Paillier cryptosystem [16], which is based on the composite residuosity problem. The algorithm is formally presented in Algorithm 1.

2.3 FedAvg algorithm

The most popular aggregation technique that is widely used in FL is FedAvg [17]. The FL Server selects few number of clients or all clients for each round of aggregation process. The weights / gradients sent by each client are aggregated by finding the average and a global value is proposed. This global update is sent to the corresponding clients and the clients update the local model accordingly. This particular procedure is iterated multiple times to derive an accurately working

model. The speed of FedAvg highly depends on the number of clients that are considered for each iteration. The more the number of clients, the faster the convergence speed [18].

Assume the number of clients considered for each iteration as C, and the weight update that is given by each client is w_i . By applying FedAvg, the global weight, \hat{W} , is computed as in Eq. (1) [19]. The simplified procedure of FedAvg aggregation is given in Algorithm 2.

$$\hat{W} = \frac{1}{C} \sum_{k=0}^C w_i \quad (1)$$

3. LITERATURE REVIEW

The decentralized security model proposed by Sultana et al. [20] secures electronic health records using blockchain and the principle of zero trust. Blockchain has been widely used in healthcare for privacy preservation, immutability, transparency, and decentralized access. Zero trust security model enhances security by providing authentication and authorization to users and devices. Three layers are used for login authentication, checking of health parameters and encryption.

Algorithm 2: FedAvg

C-number of clients, D_c -data handled by each client c, \hat{W} -global model parameter, w_i -local model parameter of each client c, η -learning rate, \mathcal{E} -number of local training epochs, SGD (Stochastic Gradient Descent)-local optimization algorithm.

- 1 Initialize the global model parameters \hat{W} , that is a random or pre-trained weight.
- 2 For each client c,
- 3 Initialize local model parameters, $w_i=\hat{W}$.
- 4 Train the local model with the local dataset D_c for \mathcal{E} epochs using SGD as $w_i=\text{SGD}(D_c, w_i, \eta)$
- 5 The server aggregates the model parameters as

$$\hat{W} = \frac{1}{C} \sum_{k=0}^C w_i$$

- 6 Steps 2 to 5 are iterated multiple times until convergence criteria are met.
 - 7 End
-

The framework presented by Feki et al. [21] uses federated learning to collaborate and train chest x-ray images of COVID-19. It takes advantage of properties like non-IID and unbalanced data distributions across the clients. ResNet50 and DenseNet121 are used for classifying the covid case and non-covid case based on X-rays. Mini batch stochastic gradient descent is used for training the data locally. The model introduced by Adnan et al. [22] uses FL for classifying histopathology images based on data from both simulated and real-world hospital environments. The privacy is guaranteed by the use of the Differential Privacy (DP) framework without degrading the performance. The difficulty in the model is the lack of publicly available medical data. Histopathology image analysis is done using bag preparation and Multiple-Instance Learning (MIL) using a memory-based model.

In reference [23], the decentralized solution called FedLCon is developed that uses FL for detecting COVID-19 from medical imaging data. FedLCon eliminates the need for a

coordinating server and the single point of failure as it applies the consensus paradigm to the Adaptive Federated Learning (AdaFed) algorithm, which extends the original FL algorithm. The privacy-preserving federated averaging (PP-FedAvg) protocol is put forward by Shin et al. [24] to protect the local dataset. This protocol uses additively homomorphic encryption (AHE) to securely compute and transmit encrypted ciphertexts between clients and the central server. The work is compared with BatchCrypt and PEFL in terms of computation and communication costs, showing a smaller number of operations from the server side.

The work modelled by Makkar and Santosh [25] proffers a secure federated learning technique (SecureFed)-an aggregation method for analyzing lung abnormalities in chest X-rays for the diagnosis of COVID-19 infections. The framework compares the proposed method with FedAvg, FedMGDA+ and FedRAD aggregation methods. Of these, SecureFed claims to improve robustness, privacy and fairness by producing two vectors, namely Markov and Temp. The research performed by Han et al. [26] addresses the security and privacy concerns of the tele-dermatology healthcare system using a strong zero-watermarking technique based on federated learning. It trains the sparse autoencoder network through F to extract image features from the dermatology medical image. Low-frequency transform coefficients from the image are chosen using the Two-dimensional Discrete Cosine Transform (2D-DCT) for zero-watermarking creation. The experimental results demonstrate that the suggested scheme performs better and is more resilient to geometric and conventional attacks.

Tan et al. [27] proposed a transfer learning approach to classify breast cancer using federated learning framework. It utilizes the three stages of FL—initial update of the model, local training, and aggregating the global model. The method uses transfer learning for extracting data features from an image's region of interest (ROI) in order to facilitate careful pre-processing and data enhancement for data training purposes. The data is processed using the Synthetic Minority Oversampling Technique (SMOTE) to improve the performance. Also, it uses FedAvg-CNN and MobileNet in an FL framework to protect patient's privacy and provide security. The results focus more on improving recall factor rather than improving accuracy in an attempt to minimize false negatives.

A medical image encryption scheme is proposed by Castro et al. [28] associated with secure fingerprint-based authenticated communication. With the aim of ensuring integrity, authenticity, confidentiality of transmission of medical images and medical data, the scheme incorporates within a dominant image, an encrypted medical image, an encrypted physician fingerprint, and the patient's electronic health record (EHR). The fingerprint feature vector and the medical picture are coupled with a chaotic encryption algorithm that utilizes a permutation key. In order to protect the permutation key, a hybrid asymmetric encryption scheme based on the Elliptic Curve encryption (ECC) and AES was implemented. Simulations and comparative research verified that this approach demonstrates lesser visual security of the encrypted image while maintaining higher quality in the reconstruction of the medical image as compared to other secure picture encryption methods.

The employment of algorithms based on machine learning poses serious security risks to user privacy. The utilization of user data is a must for smart health management. Pri-HF technique is emphasized by Shen et al. [29], which makes use

of federated learning to guarantee the security of data related to smart health management. The backbone network used for training is ResNet-50. The outcomes demonstrate that the Pri-HF method's efficiency and accuracy are more suitable than GoogleNet, ResNet, and VGGNet. Federated learning does, however, restrict the algorithm's performance, which needs to be addressed in the future.

A deep learning model based on the FL framework is described by Kundu et al. [30] for classifying viruses that lead to monkeypox disease. The work is carried over in three sections. Initially, deep learning models like MobileNetV2, Vision Transformer, and ResNet50 are used for classification. After classifying, a cycle-consistent generative adversarial network (GAN) is used for training the data samples. Finally, the federated learning environment is employed for security. The tests are carried out on publicly open datasets, and the experiments prove that the ViT-B32 model achieves an astounding 97.90% accuracy rate, highlighting the stability of the suggested framework and its potential for safe and precise classification of the monkeypox virus.

The comparative method to Federated Learning is Split Learning and this method is reviewed by Kiruthika et al. [31]. The electronic health records of patients are preserved securely using split learning, where no centralized servers are employed for aggregation. A completely distributed nature is followed for training and aggregation of datasets, which assures to provide more security and privacy. Multiple architectures of Split Learning, like Vanilla SL model, Vertical SL model, U-shaped SL model, and Extended Vanilla SL model are compared. Though SL seems to be a completely secure data management environment, there are a few open issues and challenges that are addressed in the work.

Although the works discussed so far ensure security and privacy using FL, they lack in preserving the medical images in the client's end. The proposed synthesized FL framework is novel as it tends to preserve the medical images using Partially Homomorphic encryption. The gradients/model updates that are communicated between the clients and the server are not the original image values but only the values of the encrypted image. This method works in compliance with GDPR and HIPAA, as the actual images are not collected and communicated to others. Hence, employing this framework ensures privacy and security of medical images not only during training but also during communication.

4. PROPOSED FRAMEWORK

4.1 Image encryption on the client side

Medical images are huge in size, and multiple angles of the same body part are required for perfect diagnosis. Each image ranges from 100 kilobytes to 30 megabytes [32]. Because of this, the images occupy huge memory space and it is difficult to be transmit them to a central location for training. In the proposed model, this difficulty is overcome by keeping the images in the local / client machine itself, where they are stored and trained. Instead of training the original images, the images are encrypted using Partially Homomorphic Encryption, as in Figure 2. The images are converted to NumPy arrays based on the pixel values. Encryption is performed over the NumPy values only. The medical images are mostly in gray-scale and so the NumPy arrays consist of values between 0 and 1. The image dataset that is considered

for study is Alzheimer's Dataset [33]. The advantage of using PHE is that the encrypted image can be used for training directly, and decryption is not needed. The trained model weights are sent to the server end. Since the training happens

on an encrypted image, the weights it produces are also encrypted. Even if an intruder get hold of the parameters, he may be clueless. This assures high security and confidentiality of the trained model.

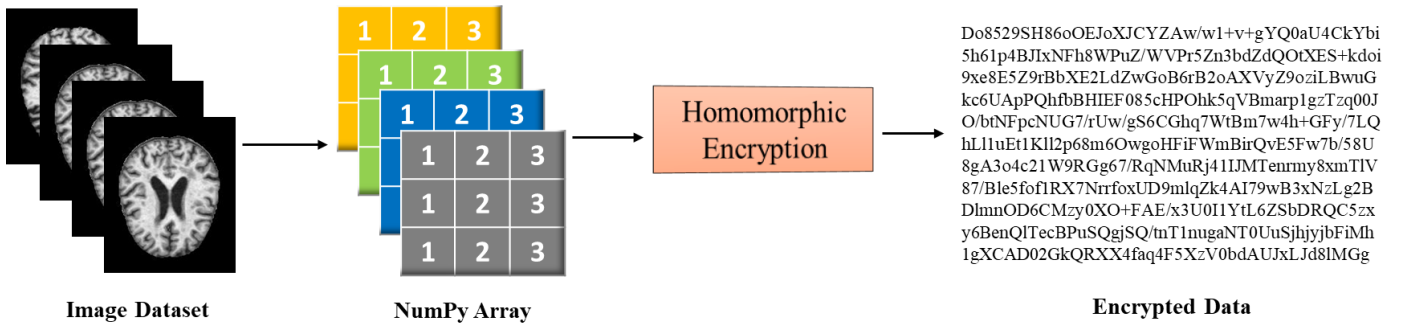


Figure 2. Homomorphic encryption of images

4.2 Implementation of federated learning

The FL model considered here is assumed to use a horizontal learning process as the images across the medical network will share the same properties. Cross-silo model is used here as the number of participating clients is similar and the clients are homogeneous as they use the same machine learning models. Model updates and the merging of local and global models are accomplished cooperatively in the synthesized federated learning framework. Using their data, each client trains a local model, and the model updates are done locally without exchanging raw patients' images. The FedAvg algorithm is employed to aggregate and merge the updated local models into a global model. By using this aggregation procedure, data privacy is maintained, and the global model gains knowledge from all clients. Hence, the steps in implementing FL can be consolidated as model selection, local model training, aggregation of local models, and global model update.

Let us consider that there is a Server, S , with 'n' number of clients. Each of the clients (C_i) holds a particular set of image dataset, ID_i . Training is done at the client's end, and each client produces local weights, W_i . The weights are aggregated by the server and it produces the output 'W', which is then iterated back to the clients. Algorithm 3 and Algorithm 4 detail the federated learning process at the Server and Clients' sides respectively.

Algorithm 3: Federated Learning at Server (S)

Input: NOR (Number of Rounds) as integer

Procedure Server Aggregation (C_r, n)

Round 1:

- 1 Initialize the weight of global model \hat{W}_0 and send to all Clients, C_i , where i ranges from 1 to n .

Round 2...NOR:

- 2 Select a random number 'r' of Clients, C_r
- 3 for each client $C_i \in C_r$ do
- 4 Send w_i to client C_r
- 5 $w_i = \text{ClientModel}(n, w_i, \eta)$
- 6 end for
- 7 Server aggregates the model parameters as

$$\hat{W} = \sum_{k=i}^n w_i$$

- 8 End procedure

4.3 CNN architectures

CNN is the most widely used mathematical based architecture that performs a convolution operation for medical image classification [33]. It includes multiple layers that are fully connected to provide a fine-tuned classification. CNN compresses the images for classification and feature learning. The basic architecture of CNN is composed of Convolutional layers, Pooling layers and Fully-connected layers. Each of these layers consists of multiple sub-layers that help in making the recognition of features in a fine-tuned manner.

Algorithm 4: Federated Learning at Client (C_i)

Input: η (Learning Rate), ϵ (Local epochs), ID_i (Local Image Dataset), $\$$ (Loss Function)

Procedure ClientModel (w_i)

- 1 $w_i \rightarrow \hat{W}$ (local weight)
- 2 for each epoch ϵ do
- 4 Compute local gradient $G_\phi \leftarrow \nabla \(\hat{W})
- 5 Update local model as $\hat{W} \leftarrow \hat{W} - \eta G_\phi$
- 6 end for
- 7 return \hat{W} to Server
- 8 End procedure

A convolutional layer performs the convolution operation and is responsible for extracting features that are available in any part of the image, including corners and edges. The convolution parameters have Kernels, K (filters) to learn the image. Let us say the parts of the image as I . So, $K \cdot I$ (K dot I) is calculated by sliding the kernel over the input image. The output of this layer is a Feature Map that is given to the pooling layer. The pooling layer is responsible for reducing the size of the feature map, so that the computation can be faster and less complex. Multiple pooling functions can be applied, such as Max pooling, Min pooling, Average pooling, and Global pooling. The choice of the type of pooling completely depends on the application designer. The Fully Connected layer receives flattened input from the previous layers, and this layer performs the classification process. This layer consists of neurons that operate based on weights and biases. The output of the fully connected layers is given to the activation function like the sigmoid function, to convert the real values to target class probabilities. Finally, the output will classify the image to be demented or not.

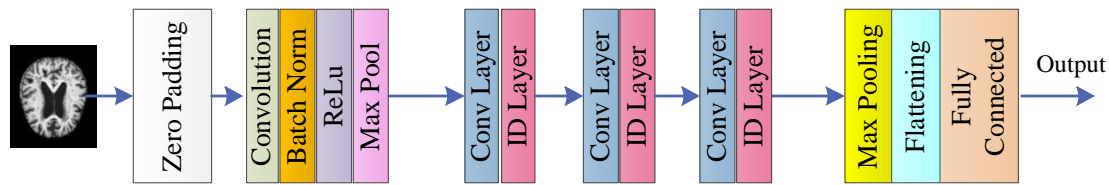


Figure 3. Overview of ResNet50 architecture

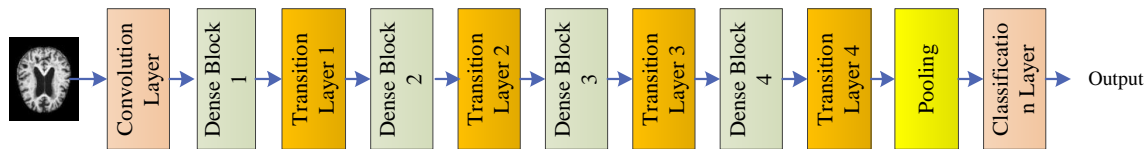


Figure 4. Overview of DenseNet121 architecture

This architecture is applied and used by different models like LeNet, GoogLeNet, AlexNet, VGGNet (Visual Geometry Group), ResNet, DenseNet and many more. Here, ResNet50 and DenseNet121 are considered based on the competitive performance of these two architectures especially in classifying medical images and level of complexity.

4.3.1 ResNet50

ResNet50 is a version of the convolutional neural network architecture that adapts the basic methodology of CNN with slight variations [34]. It is a trained deep-learning network that is capable of classifying images accurately with the help of Residual blocks. These blocks handle the problem of vanishing gradients in deep neural networks by skipping a few layers. They bypass a few layers, thereby not allowing the problem of vanishing gradient. Layers of CNN are used in conjunction with an additional layer in ResNet50 called the Identity layer, which adds the input back to the output after passing it through the convolutional layers. As a result, the network can learn residual functions, which convert input into desired output. Prior to the 3×3 convolutional layers, the number of filters is decreased by adding a 1×1 convolutional layer. ResNet50 extracts the features accurately by using the filters. In order to normalize the activation of the layers and enable faster and more efficient network training, ResNet50 employs Batch normalization. Figure 3 gives an overall structure of the ResNet50 model.

In this paper, ResNet50 model is employed for extracting features in medical images. The output of the last convolutional layer is fed via a flattened layer, which transforms the output to a 2D array. After the feature map has been flattened, it is run through a Dropout layer that has a drop rate of 0.5. This layer serves as regularisation to stop overfitting. The final output of the network, which indicates the expected probability that the input image belongs to the target class, is obtained by passing the output of the dropout layer through a dense layer with one unit and a sigmoid activation function. Since ResNet50 is not as complex as its successors ResNet101 and ResNet152, it makes it more suitable for classifying medical images encrypted with partially homomorphic encryption.

4.3.2 DenseNet121

Unlike ResNet, DenseNet is a denser convolutional neural network. Though it performs similar to ResNet, DenseNet does not skip any layer with the help of residual blocks. The output of previous layer is fed as the input to the future layers.

That is, when there are 100 layers, then the 100th layer will receive feature maps as input from all the previous 99 layers [35]. Let us assume a DenseNet with N number of layers, then the number of connections between the layers in DenseNet can be given as $N(N+1)/2$. This presumes that accuracy is achieved with a smaller number of layers itself, during the training process. Because of this property, there may be a problem of collision of feature maps from different layers. In order to overcome this issue, separate dense blocks are created, where each dense block may have a fixed number of layers inside them. Each dense block produces an output that is fed to the transition/convolution layer. Max pooling is performed over the output of the transition layer, which reduces the feature map's size.

Figure 4 gives an overview of DenseNet model Architecture. DenseNet121 is comprised of 4 dense blocks, each of which has 6, 12, 24, and 16 layers, respectively. Four dense blocks are followed by a classification layer, and this layer performs classification based on the feature maps received from all the previous layers. DenseNet121 is less complex than the other three versions—DenseNet169, DenseNet201, and DenseNet264. DenseNet is particularly useful in classifying the gray scale images that are used in the medical industry.

ResNet50 is a deep residual network that effectively performs gradient propagation. This aspect is crucial when training medical images, because a small loss can even result in major drift in encryption. Skip connections help in mitigating the vanishing gradients problem, at the same time ensuring stable training. DenseNet121 facilitates feature reuse, which is an amicable benefit when dealing with encrypted images. Also, DenseNet can propagate features, which is essential in processing medical images, where few losses are unpredictable during encryption. Other architectures such as EfficientNet, VGGNet, and MobileNet are not considered in this work because of the listed reasons. EfficientNet requires extensive tuning to moderate the computational cost and accuracy, which may not be suitable for a federated learning environment, where resource constraints are high. VGGNet is not considered in this study as it works on more parameters than ResNet and DenseNet, resulting in a higher computational cost. MobileNet is more suitable and efficient for a federated edge computing environment, but it may fail to address the intricate details necessary for processing medical images. In addition to this, ResNet20 and DenseNet121 are highly suitable for encrypted data because of their ability to extract features and robust gradient flow for encrypted images. EfficientNet, VGGNet and MobileNet showcases moderate to

low suitability for working with encrypted images. So, this study considers ResNet50 and DenseNet121, as they tend to balance model efficiency and performance, making them suitable for medical image's privacy preservation.

4.4 Model aggregation

Each client systems employ the above-specified deep learning models–ResNet50 and DenseNet121–to train the encrypted image. Once the training is over, the clients send the model parameters to the Server in order to update the global model. There are two methods of model aggregation – parameter-based aggregation and output-based aggregation. Here, parameter-based aggregation is employed, and so the weights and gradients of the local model are transferred to the Server. There are various FL aggregation algorithms used for the purpose of fusing the client models, such as FedAvg, FedProx, FedNova, Scaffold, Zeno, Per-FedAvg, FedMax, FedMin, etc. [36].

Here, FedAvg is used by the Server to aggregate the models into a global model. The parameters passed to the Server by the clients are the values received out of encrypted images. So, the same public key is used by the clients and the server use the additive homomorphic property to add the encrypted values. The secret key is shared among the clients using the Paillier cryptosystem.

4.5 Synthesized framework

The details discussed so far are combined to frame the proposed synthesized architecture, as in Figure 5. The framework takes in the medical images and it is converted to NumPy array that has values between 0 and 1. This is encrypted using Partially Homomorphic Encryption. The encrypted images are kept in the client side itself and two CNN models – ResNet50 and DenseNet121 are used for training. After recursive training, the model parameters are passed to the server. The same procedure is followed by all the clients, thereby following horizontal federated learning. Initially, the client uses the global model parameter sent by the server and trains the data. After that, it iterates the training process. The server collects the updated model parameters from all the connected clients or from specific number of clients and performs FedAvg aggregation. The global model is updated and the same is iterated back to the clients. This procedure is repeated until the epochs and desired accuracy are reached.

The proposed framework effectively secures medical images using PHE while utilizing ResNet50 and DenseNet121 for robust feature extraction. FedAvg reduces the communication overhead as it performs local updates before updating the encrypted gradients to the server. Scalability is ensured even with multiple health centers in different places. The clients can also be selected dynamically and randomly to minimize the computational load and to optimize the training efficiency. Although, PHE is computationally efficient than FHE, PHE incurs approximately 15-25% increase in computational cost for encrypting gradients. The training time of encrypted images is 20-30% higher than plain text training with ResNet50 and DenseNet121. But, training plain images is not secured and PHE is used for encryption.

The encrypted images still possess enough structural patterns that are essential for feature extraction and data confidentiality. However, the encryption performed provides a certain amount of transformation that may result in the loss

of fine details of the image. To counteract this optimistic loss of image quality, our model utilizes robust CNN architecture models, ResNet50 and DenseNet121, to perform an optimum level of feature extraction even with little distortions.

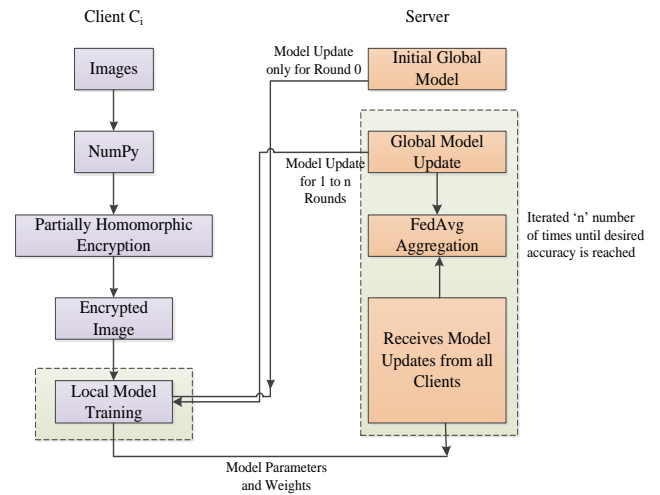


Figure 5. Proposed synthesized framework

5. RESULTS AND DISCUSSIONS

The Alzheimer's dataset [37] consists of MRI images and the dataset is divided into training data and testing data. The training and testing images are classified as mild demented, Moderate demented, Non-demented, and very mild demented. The images are resized to 128×128 and they are converted to NumPy array based on pixel values. Since these are almost grey-scale images, the NumPy values range between 0 and 1. After converting to NumPy, it is encrypted using PHE. Only the encrypted images are trained using the CNN models–ResNet50 and DenseNet121. Since four types of classifications are done in medical image, multi-class classification is done. Let us consider the Alzheimer disease classifications under multiple classes where Class A denotes Non-demented cases, Class B denotes Very mild demented cases, Class C denotes mild demented cases and Class D denotes Moderate demented cases. The metrics considered are model Accuracy, Precision, Recall and F1-Score. These metrics are calculated for each class categorization as in Eqs. (2)-(5), where TP refers to True Positives, TN refers to True Negatives, FP refers to False Positives and FN refers to False Negatives. The same calculation is performed for each of the classes.

$$Accuracy_{Class A} = \frac{TP_{Class A} + TN_{Class A}}{TP_{Class A} + TN_{Class A} + FP_{Class A} + FN_{Class A}} \quad (2)$$

$$Precision_{Class A} = \frac{TP_{Class A}}{TP_{Class A} + FP_{Class A}} \quad (3)$$

$$Recall_{Class A} = \frac{TP_{Class A}}{TP_{Class A} + FN_{Class A}} \quad (4)$$

$$F1 - score_{Class A} = 2 \times \frac{Precision_{Class A} \times Recall_{Class A}}{Precision_{Class A} + Recall_{Class A}} \quad (5)$$

Table 1. Performance comparison of ResNet50 and DenseNet121

Class	ResNet50			DenseNet121		
	Recall	Precision	F1-Score	Recall	Precision	F1-Score
A	93%	91%	92%	93%	88%	90%
B	90%	81%	85%	90%	85%	88%
C	76%	75%	75%	76%	83%	79%
D	72%	75%	80%	77%	83%	75%
Accuracy	84%			86%		

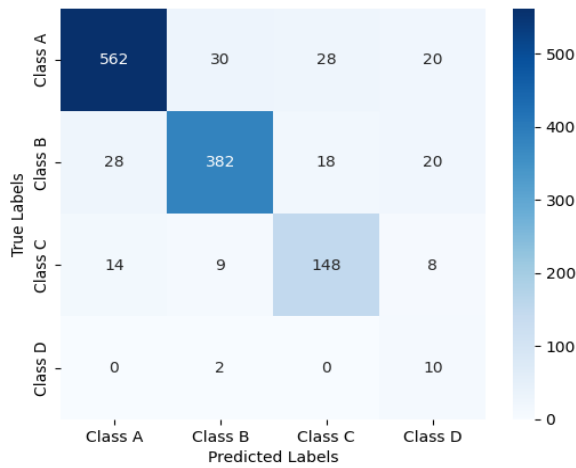


Figure 6. Confusion matrix of ResNet50

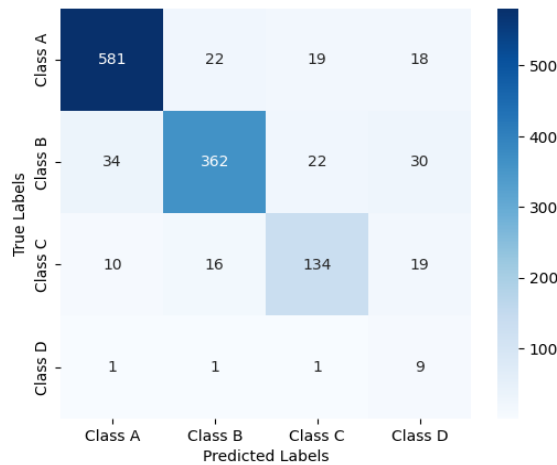
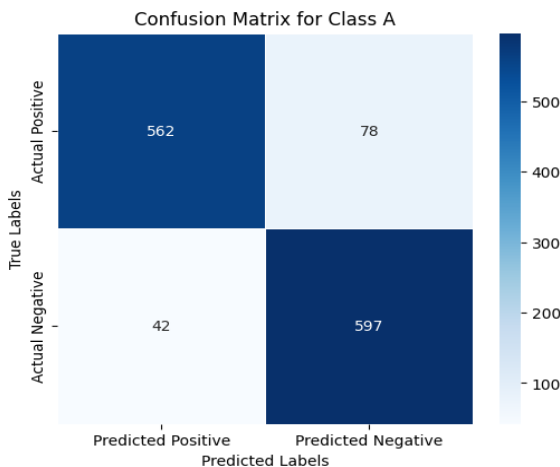
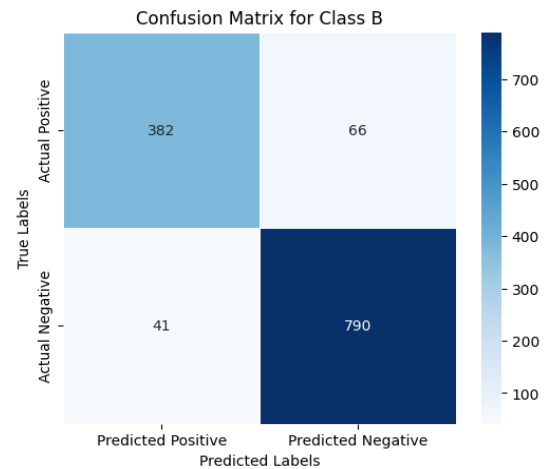


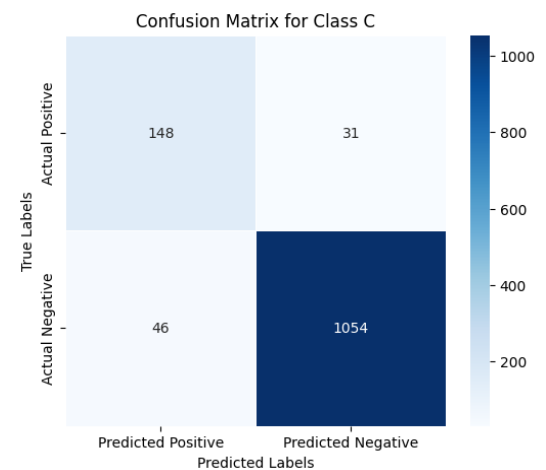
Figure 7. Confusion matrix of DenseNet121



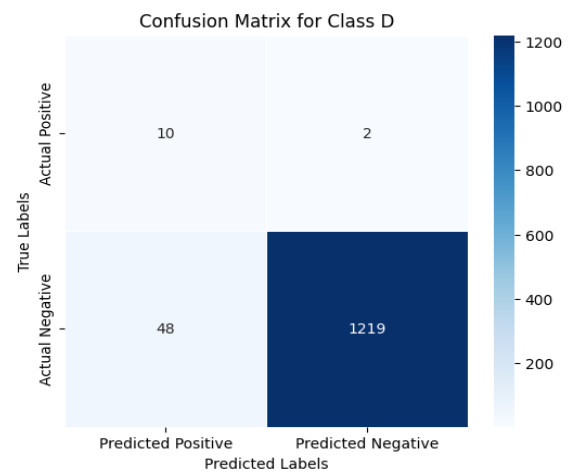
(a) Confusion matrix for class A



(b) Confusion matrix for class B



(c) Confusion matrix for class C

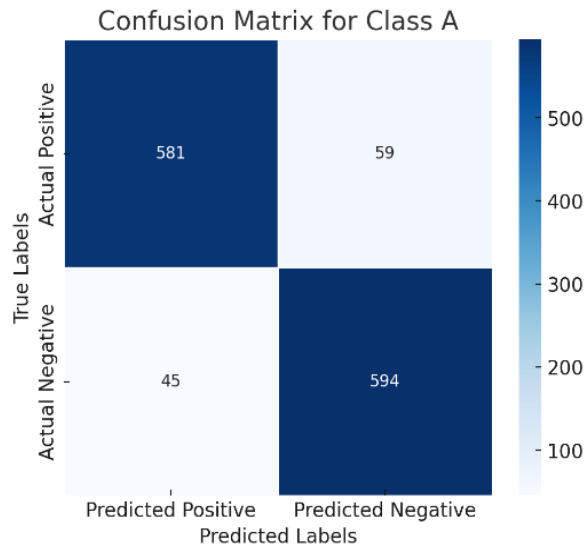


(d) Confusion matrix for class D

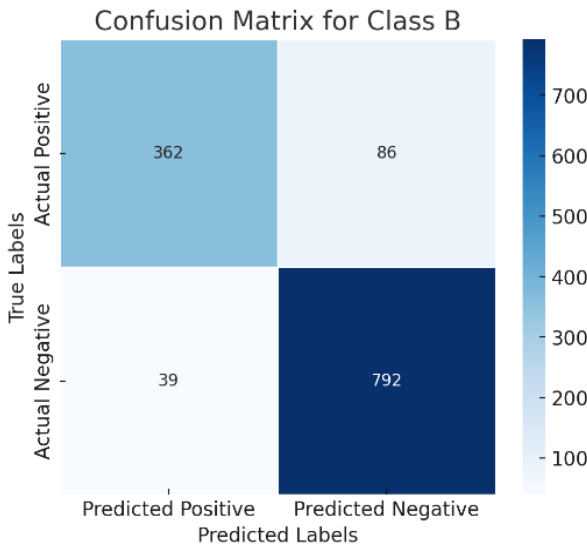
Figure 8. Confusion matrices of ResNet50 with binary classification of classes

The confusion matrix of ResNet50 and DenseNet121 are in Figure 6 and Figure 7, respectively. It can be evidenced that 562 Non – demented cases are identified correctly in ResNet50, while DenseNet121 identified 581, providing more value of True Positives. Accuracy is the ratio of correctly identified cases to the total of all cases. Here, DenseNet121 showed up an accuracy rate of 86% while the accuracy of ResNet50 is

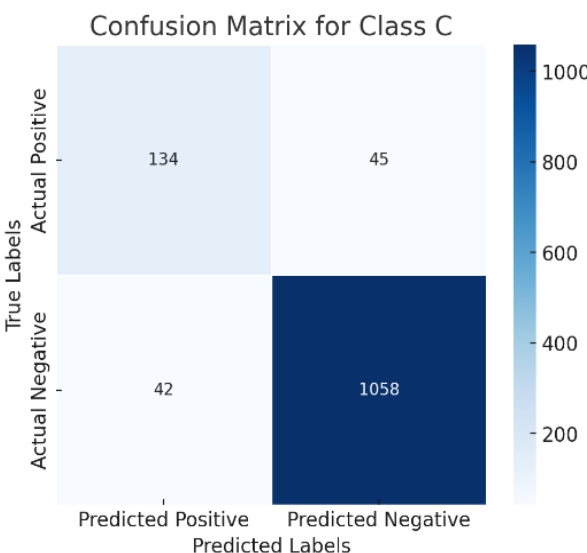
84%. There may be huge difference when unencrypted images are used in the process. Table 1 shows the accuracy of DenseNet121 is greater than Resnet50.



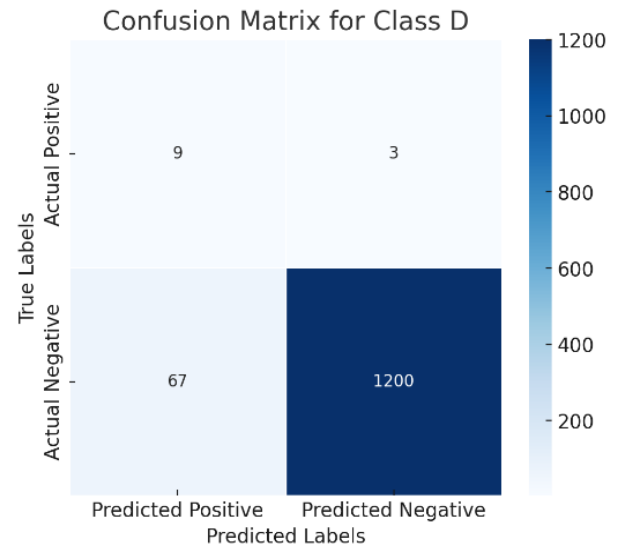
(a) Confusion matrix for class A



(b) Confusion matrix for class B



(c) Confusion matrix for class C



(d) Confusion matrix for class D

Figure 9. Confusion matrices of DenseNet121 with binary classification of classes

The confusion matrix generated can be redefined by following a one-vs-all approach, where each class is considered separately, that is, one at a time. This makes the selected class positive and all other classes are considered to be negative. A binary confusion matrix can be derived for all classes discussed. Figures 8 and 9 show the confusion matrix of all classes for ResNet50 and DenseNet121, respectively, for a more comprehensive evaluation.

Accuracy curves of ResNet50 and DenseNet121 over the encrypted image dataset and the unencrypted image dataset are shown in Figure 10. It can be noted that DenseNet121 gives better accuracy than ResNet50 because of the fact that the number of datasets considered is relatively smaller. This suggests that in the case of a small dataset, some specially created small-scale networks might be more appropriate for medical image classification than heavyweight networks. Since client-side machines handle a limited dataset, DenseNet121 is more appropriate for image classification than ResNet50. Although PHE provides a stronger encryption mechanism that eventually results in higher privacy, it can marginally reduce the overall performance this is depicted in Figure 10.

In addition to this, Receiver Operating Characteristic–Area Under Curve (ROC-AUC) is used for evaluating the trade-off between True Positive Rate (TPR) and False Positive Rate (FPR). TPR, also called Recall or Sensitivity, is defined in Eq. (4). FPR for class A can be defined as in Eq. (6). The same is applied for other classes as well.

$$FPR_{Class A} = \frac{FP_{Class A}}{FP_{Class A} + TN_{Class A}} \quad (6)$$

Figure 11 and Figure 12 give the One-vs-All Receiver Operating Characteristic Curve (OvA ROC) of ResNet50 and DenseNet121, respectively. While ROC is used for evaluating the performance of binary classification models, OvA ROC is used for evaluating multi-class classification models. ROC curves are plotted between the true positive rate (TPR) and false positive rate (FPR) for different threshold values. Each point on the ROC curve represents a TPR-FPR pair corresponding to a particular decision threshold. In the OvA

strategy, each class is treated as the positive class while the other classes are treated as negative. This way, a separate ROC curve is generated for each class, resulting in multiple ROC curves (one for each class).

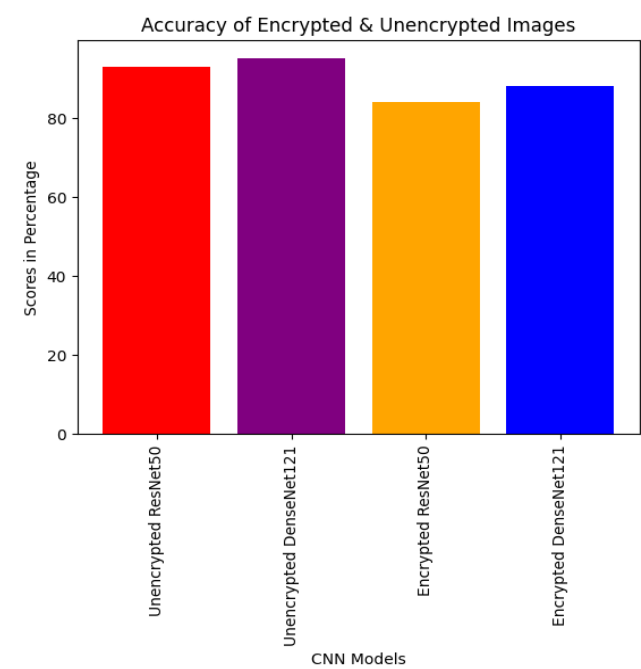


Figure 10. Performance of ResNet50 and DenseNet121 over encrypted and unencrypted data

Our dataset contains four classes and Class A is given as Class 0, B as 1, C as 3 and D as 4. For each class in the dataset, a separate binary classifier distinguishes that class from all other classes combined. Here, we have 4 classes (A, B, C, D), so 4 binary classifiers are trained as A vs (B+C+D), B vs (A+C+D), C vs (A+B+D) and D vs (A+B+C). Once the binary classifiers are trained, ROC curve for each class is computed individually. Area under the ROC curve (AUC) is calculated for each class separately. AUC represents the performance of the classifier at distinguishing the positive class from the negative class. A higher AUC indicates better performance.

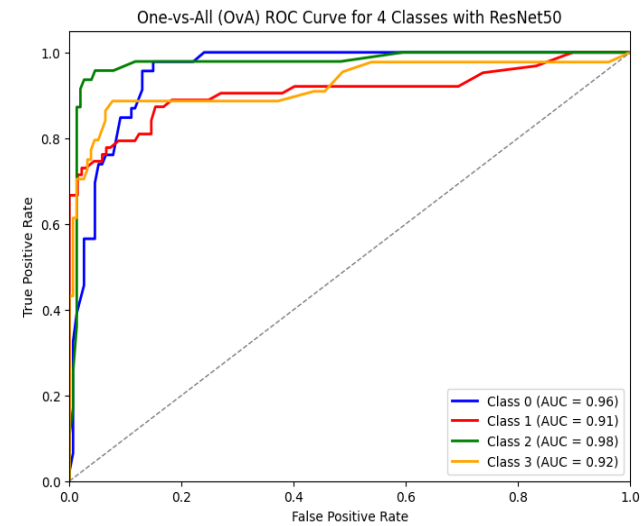


Figure 11. Receiver operating characteristic of ResNet50

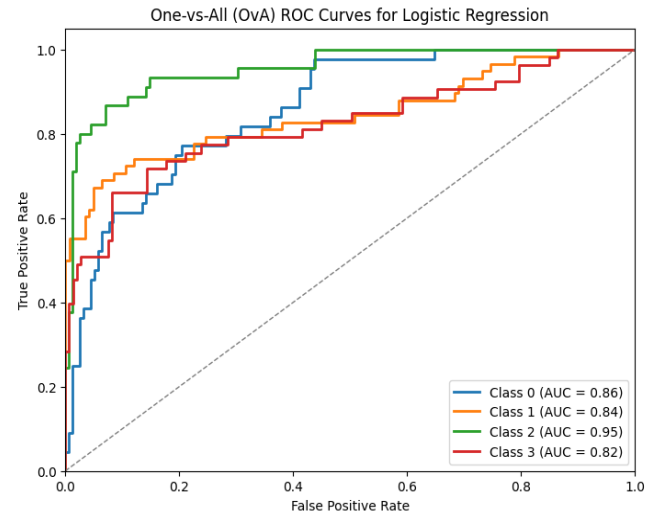


Figure 12. Receiver operating characteristic of DenseNet121

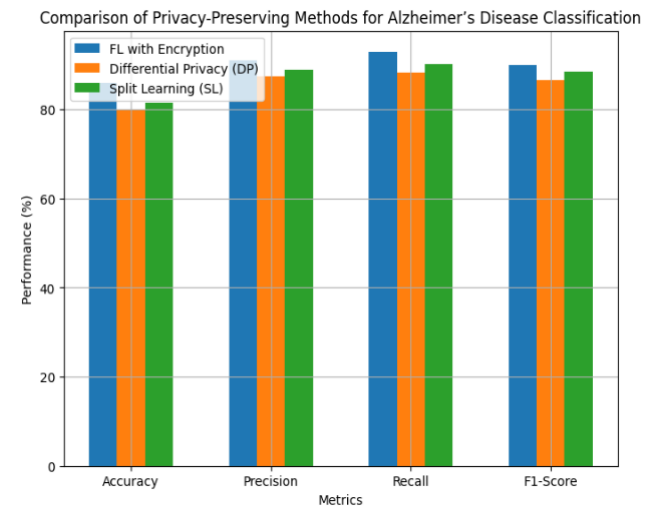


Figure 13. Comparison of performance metrics of different privacy-preserving methods

Table 2. Performance comparison of the proposed encrypted model with DP and SL

Method	Accuracy	Precision	Recall	F1-Score
FL with Encryption	86%	91%	93%	90%
DP	80%	87.5%	88.2%	86.6%
SL	81.5%	89%	90.1%	88.4%

The proposed FL with encryption model considered now is the encrypted DenseNet121 model and it can be compared with Differential Privacy (DP) and Split Learning (SL). When evaluating the performance metrics, the proposed method achieves better model accuracy without degrading the performance. When DP is used, the performance of the model slightly degrades due to noise addition in images. when compared with split learning, the encryption process in FL is computationally efficient because SL requires complex data exchanges between the peers. Table 2 shows the performance metrics of the proposed model in comparison with DP and SL. Figure 13 gives the performance comparison of the three different privacy-preserving model for better clarification.

6. CONCLUSION

The work demonstrates a synthesized framework that uses federated learning and PHE for medical image encryption. The model allows the medical images to reside on the client end itself, without the need to move to a central location. Two pre-trained CNN models, ResNet50 and DenseNet121, are used for training the encrypted images on the client side. Using FL and PHE ensures data integrity, confidentiality, privacy, and availability. The results show that DenseNet121 performs better than ResNet50. Despite several advantages, there are a few setbacks also. ResNet-50 is more suitable when the framework is used for large datasets with a large number of classes, while DenseNet-121 is more parameter-efficient and effective in dealing with limited training data or smaller datasets as in our case. The proposed framework demonstrates improved model accuracy and privacy preservation, with results showing that the model consumes less computation and communication overhead.

FedAvg aggregation can be replaced with Secure Multiparty Computation. In the future, the results of these two models can be compared with a few more advanced deep learning models. The dataset considered in this work is images, but the medical practitioner predicts the diseases not only based on images but also using clinical records, genetic information, and lab reports. So, in the future, FL can be trained to handle these multi-modal sources for more accuracy.

REFERENCES

- [1] Secinaro, S., Calandra, D., Secinaro, A., Muthurangu, V., Biancone, P. (2021). The role of artificial intelligence in healthcare: A structured literature review. *BMC Medical Informatics and Decision Making*, 21: 1-23. <https://doi.org/10.1186/s12911-021-01488-9>
- [2] Karatas, M., Eriskin, L., Deveci, M., Pamucar, D., Garg, H. (2022). Big data for healthcare industry 4.0: Applications, challenges and future perspectives. *Expert Systems with Applications*, 200: 116912. <https://doi.org/10.1016/j.eswa.2022.116912>
- [3] Thapa, C., Camtepe, S. (2021). Precision health data: Requirements, challenges and existing techniques for data security and privacy. *Computers in Biology and Medicine*, 129: 104130. <https://doi.org/10.1016/j.combiomed.2020.104130>
- [4] Khan, L.U., Saad, W., Han, Z., Hossain, E., Hong, C.S. (2021). Federated learning for internet of things: Recent advances, taxonomy, and open challenges. *IEEE Communications Surveys & Tutorials*, 2(3): 1759-1799. <https://doi.org/10.48550/arXiv.2009.13012>
- [5] Acar, A., Aksu, H., Uluagac, A.S., Conti, M. (2018). A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys (Csur)*, 51(4): 1-35. <https://doi.org/10.1145/3214303>
- [6] Ogburn, M., Turner, C., Dahal, P. (2013). Homomorphic encryption. *Procedia Computer Science*, 20: 502-509. <https://doi.org/10.1016/j.procs.2013.09.310>
- [7] Li, X., Huang, K., Yang, W., Wang, S., Zhang, Z. (2019). On the convergence of fedavg on non-iid data. *arXiv Preprint arXiv: 1907.02189*. <https://doi.org/10.48550/arXiv.1907.02189>
- [8] Coelho, K.K., Nogueira, M., Vieira, A.B., Silva, E.F., Nacif, J.A.M. (2023). A survey on federated learning for

security and privacy in healthcare applications. *Computer Communications*, 207: 113-127. <https://doi.org/10.1016/j.comcom.2023.05.012>

- [9] KhoKhar, F.A., Shah, J.H., Khan, M.A., Sharif, M., Tariq, U., Kadry, S. (2022). A review on federated learning towards image processing. *Computers and Electrical Engineering*, 99: 107818. <https://doi.org/10.1016/j.compeleceng.2022.107818>
- [10] Mothukuri, V., Parizi, R.M., Pouriye, S., Huang, Y., Dehghantanha, A., Srivastava, G. (2021). A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115: 619-640. <https://doi.org/10.1016/j.future.2020.10.007>
- [11] Pouriye, S., Shahid, O., Parizi, R.M., Sheng, Q.Z., Srivastava, G., Zhao, L., Nasajpour, M. (2022). Secure smart communication efficiency in federated learning: Achievements and challenges. *Applied Sciences*, 12(18): 8980. <https://doi.org/10.3390/app12188980>
- [12] Rahman, K.J., Ahmed, F., Akhter, N., Hasan, M., Amin, R., Aziz, K.E., Islam, A.M., Mukta, M.S.H., Islam, A.N. (2021). Challenges, applications and design aspects of federated learning: A survey. *IEEE Access*, 9: 124682-124700. <https://doi.org/10.1109/ACCESS.2021.3111118>
- [13] Moshawrab, M., Adda, M., Bouzouane, A., Ibrahim, H., Raad, A. (2023). Reviewing federated learning aggregation algorithms; strategies, contributions, limitations and future perspectives. *Electronics*, 12(10): 2287. <https://doi.org/10.3390/electronics12102287>
- [14] Fang, H., Qian, Q. (2021). Privacy preserving machine learning with homomorphic encryption and federated learning. *Future Internet*, 13(4): 94. <https://doi.org/10.3390/fi13040094>
- [15] Ma, J., Naas, S.A., Sigg, S., Lyu, X. (2022). Privacy-preserving federated learning based on multi-key homomorphic encryption. *International Journal of Intelligent Systems*, 37(9): 5880-5901. <https://doi.org/10.1002/int.22818>
- [16] Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on The Theory and Applications of Cryptographic Techniques*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 223-238. https://doi.org/10.1007/3-540-48910-X_16
- [17] McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, PMLR, pp. 1273-1282.
- [18] Qu, Z.N., Lin, K.X., Li, Z.J., Zhou, J.Y. (2021). Federated learning's blessing: Fedavg has linear speedup. In *ICLR 2021-Workshop on Distributed and Private Machine Learning (DPML)*, pp. 1-47.
- [19] Sun, T., Li, D.S., Wang, B. (2022). Decentralized federated averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4): 4289-4301. <https://doi.org/10.1109/TPAMI.2022.3196503>
- [20] Sultana, M., Hossain, A., Laila, F., Taher, K.A., Islam, M.N. (2020). Towards developing a secure medical image sharing system based on zero trust principles and blockchain technology. *BMC Medical Informatics and Decision Making*, 20: 1-10. <https://doi.org/10.1186/s12911-020-01275-y>
- [21] Feki, I., Ammar, S., Kessentini, Y., Muhammad, K.

- (2021). Federated learning for COVID-19 screening from Chest X-ray images. *Applied Soft Computing*, 106: 107330. <https://doi.org/10.1016/j.asoc.2021.107330>
- [22] Adnan, M., Kalra, S., Cresswell, J.C., Taylor, G.W., Tizhoosh, H.R. (2022). Federated learning and differential privacy for medical image analysis. *Scientific Reports*, 12(1): 1953. <https://doi.org/10.1038/s41598-022-05539-7>
- [23] Giuseppe, A., Manfredi, S., Menegatti, D., Poli, C., Pietrabissa, A. (2022). Decentralised federated learning for hospital networks with application to COVID-19 detection. *IEEE Access*, 10: 92681-92691. <https://doi.org/10.1109/ACCESS.2022.3202922>
- [24] Shin, Y.A., Noh, G., Jeong, I.R., Chun, J.Y. (2022). Securing a local training dataset size in federated learning. *IEEE Access*, 10: 104135-104143. <https://doi.org/10.1109/ACCESS.2022.3210702>
- [25] Makkar, A., Santosh, K.C. (2023). SecureFed: federated learning empowered medical imaging technique to analyze lung abnormalities in chest X-rays. *International Journal of Machine Learning and Cybernetics*, 14(8): 2659-2670. <https://doi.org/10.1007/s13042-023-01789-7>
- [26] Han, B., Jhaveri, R.H., Wang, H., Qiao, D., Du, J. (2021). Application of robust zero-watermarking scheme based on federated learning for securing the healthcare data. *IEEE Journal of Biomedical and Health Informatics*, 27(2): 804-813. <https://doi.org/10.1109/JBHI.2021.3123936>
- [27] Tan, Y.N., Tinh, V.P., Lam, P.D., Nam, N.H., Khoa, T.A. (2023). A transfer learning approach to breast cancer classification in a federated learning framework. *IEEE Access*, 11: 27462-27476. <https://doi.org/10.1109/ACCESS.2023.3257562>
- [28] Castro, F., Impedovo, D., Pirlo, G. (2023). A medical image encryption scheme for secure fingerprint-based authenticated transmission. *Applied Sciences*, 13(10): 6099. <https://doi.org/10.3390/app13106099>
- [29] Shen, Z., Ding, F., Yao, Y., Bhardwaj, A., Guo, Z., Yu, K. (2022). A privacy-preserving social computing framework for health management using federated learning. *IEEE Transactions on Computational Social Systems*, 10(4): 1666-1678. <https://doi.org/10.1109/TCSS.2022.3222682>
- [30] Kundu, D., Rahman, M.M., Rahman, A., Das, D., Siddiqi, U.R., Alam, M.G.R., Dey, S.K., Muhammad, G., Ali, Z. (2024). Federated deep learning for monkeypox disease detection on GAN-Augmented dataset. *IEEE Access*, 12: 32819-32829. <https://doi.org/10.1109/ACCESS.2024.3370838>
- [31] Kiruthika, M., Kumar, A., Krishnasamy, L., Sarveshwaran, V. (2024). Investigation on preserving privacy of electronic medical record using split learning. *Procedia Computer Science*, 233: 614-622. <https://doi.org/10.1016/j.procs.2024.03.251>
- [32] Dandu, R.V. (2008). Storage media for computers in radiology. *Indian Journal of Radiology and Imaging*, 18(04): 287-289. <https://doi.org/10.4103/0971-3026.43838>
- [33] Wu, J.X. (2017). Introduction to convolutional neural networks. *Nanjing University*, 5(23): 495. <https://cs.nju.edu.cn/wujx/paper/CNN.pdf>
- [34] Mascarenhas, S., Agarwal, M. (2021). A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification. In *2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*, Bengaluru, India, pp. 96-99. <https://doi.org/10.1109/CENTCON52345.2021.9687944>
- [35] Chhabra, M., Kumar, R. (2022). A smart healthcare system based on classifier DenseNet 121 model to detect multiple diseases. In *Mobile Radio Communications and 5G Networks: Proceedings of Second MRCN 2021*, pp. 297-312. https://doi.org/10.1007/978-981-16-7018-3_23
- [36] Qi, P., Chiaro, D., Guzzo, A., Ianni, M., Fortino, G., Piccialli, F. (2024). Model aggregation techniques in federated learning: A comprehensive survey. *Future Generation Computer Systems*, 150: 272-293. <https://doi.org/10.1016/j.future.2023.09.008>
- [37] Chugh, L. (2023). Best Alzheimer's MRI Dataset 99% Accuracy. <https://www.kaggle.com/datasets/lukechugh/best-alzheimer-mri-dataset-99-accuracy>