FEDERATED LEARNING FOR EYE DISEASE PREDICTION

A PROJECT REPORT

Submitted by

KANNIGA SARASWATHY M(811721243024)

MADHUMITHA P(811721243026)

MONISH VIDYARTHI(811721243035)

SUBALATHA A(811721243054)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

MAY, 2025

K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report titled **"FEDERATED LEARNING FOR EYE DISEASE PREDICTION"** is the bonafide work of **KANNIGA SARASWATHY M (811721243024), MADHUMITHA P(811721243026), MONISH VIDYARTHI R(811721243035), SUBALATHA A(811721243054)** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. T.Avudaiappan M.E., Ph.D.,
HEAD OF THE DEPARTMENT

Department of Artificial Intelligence
K.Ramakrishnan College of Technology
(Autonomous)
Samayapuram – 621 112

SIGNATURE

Mr. P. B. Aravind Prasad M.E.,
SUPERVISOR

ASSISTANT PROFESSOR
Department of Artificial Intelligence
K.Ramakrishnan College of Technology
(Autonomous)
Samayapuram – 621 112

Submitted for the viva-voce examination held on ……

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We jointly declare that the project report on **"FEDERATED LEARNING FOR EYE DISEASE PREDICTION"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfilment of the requirement of the award of Degree of **BACHELOR OF TECHNOLOGY**.

Signature

KANNIGASARASWATHY M

MADHUMITHA P

MONISH VIDYARTHIR

SUBALATHA A

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude and in-debt to our institution "**K.Ramakrishnan College of Technology (Autonomous)**", for providing us with the opportunity to do this project.

We are glad to credit honourable chairman **Dr. K.RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

We would like to thank **Dr. N. VASUDEVAN, M.E., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

We whole heartily thank to **Dr. T.AVUDAIAPPAN**, **M.E., Ph.D.,** Head of the department, **ARTIFICIAL INTELLIGENCE** for providing his encourage pursuing this project.

We express our deep and sincere gratitude to our project guide **Mr.P.B.ARAVIND PRASAD**,**M.E.**, Department of **ARTIFICIAL INTELLIGENCE,** for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

We render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

We wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

ABSTRACT

The advancement of federated learning has transformed ocular disease diagnosis in medical imaging by enabling collaborative model training across multiple local servers while ensuring data privacy. This project presents an innovative approach leveraging Convolutional Neural Networks (CNNs) for the diagnosis of ocular diseases, with model training conducted on distributed local servers. The decentralized nature of federated learning ensures that sensitive retinal and ophthalmic image data remains localized.The system focuses on CNN architecture to enhance hierarchical feature extraction from eye images, improving diagnostic accuracy for conditions such as glaucoma, diabetic retinopathy, and macular degeneration.The federated learning paradigm facilitates collaborative model training on diverse datasets from various ophthalmology institutions, accommodating the heterogeneity inherent in ocular imaging. Each local server independently refines the model based on its unique dataset, adapting to distinct imaging modalities, patient demographics, and pathological variations. To ensure security and prevent unauthorized access, the trained CNN model is encrypted using Elliptic Curve Cryptography (ECC) before being transmitted to the central server for aggregation. ECC provides a robust encryption mechanism, guaranteeing the confidentiality and integrity.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AMD - Age-Related MAacular Degeneration

AUC - Area under curve

DME - Diabetic Macular Edema

DR - Diabetic Retinopathy

FTL - Federated Transfer Learning

OCT - Optical Coherence Tomography

ROP - Retinopathy Of Prematurity

CHAPTER 1

INTRODUCTION

Federated learning is an emerging machine learning technique that enables multiple devices to collaboratively train a model without sharing their data with each other. This approach is particularly useful in medical settings where data privacy and security are of utmost importance. In this context, medical data is often sensitive and protected by law, and traditional machine learning approaches may not be feasible due to privacy concerns. To address this issue, here propose a cross-device federated learning approach that utilizes medical datasets to build a predictive model. We also employ Elliptic Curve Cryptography (ECC) to encrypt the model during training and inference, providing an additional layer of security.

1.1 BACKGROUND

The algorithm begins with the collection of the medical image dataset with train and test samples and distributed across multiple devices. Each device then initializes a local model, and an encryption key is generated using ECC for each device. The devices train their local models on their respective subsets of the dataset, with the updates to the local model being encrypted using the encryption key. The encrypted updates are aggregated across all devices to create a global model, which is then decrypted using the encryption keys. The decrypted global model is evaluated on a hold-out dataset to assess its performance. If the performance of the model is not satisfactory, the algorithm may be retrained using the updated global model. This proposed approach ensures that medical data remains private and secure while still allowing for the development of models that can improve patient outcomes. Additionally, the use of ECC encryption provides an extra layer of security for the model, ensuring that it remains protected during training and inference.

1.2 PROBLEM STATEMENT

The increasing global burden of vision-related disorders—such as diabetic retinopathy, glaucoma, and age-related macular degeneration—demands early detection and timely intervention. Traditional AI-based diagnostic tools, while

powerful, rely heavily on centralized data collection, which poses several limitations. There is an urgent need for a solution that safeguards patient privacy, includes diverse data sources, and adapts to real-world deployment scenarios across various devices and regions.Centralized models often fail to represent the wide variability in patient demographics, imaging conditions, and device hardware, leading to biased outcomes and reduced generalizability in underserved communities. Additionally, sharing sensitive retinal images to centralized cloud servers raises significant concerns around data privacy, security, and compliance with regulations like GDPR and HIPAA.

1.3 AIM AND OBJECTIVES

1.3.1 Aim

To develop a cross-device federated learning with medical image dataset model build and model encryption using ECC Algorithm is to develop a machine learning approach that can improve patient outcomes while preserving the privacy and security of medical data.

1.3.2 Objectives

- Implement ECC encryption techniques to moderate the federated server.
- Encryption to protect the trained model during training and inference.
- Enable inclusive and privacy-preserving eye disease prediction by leveraging diverse data from low-resource and high-resource devices across geographies, reducing diagnostic disparities and ensuring AI learns from all populations.
- Partition a medical image dataset into subsets that can be distributed across multiple devices while preserving the privacy of the data.
- Implement ECC encryption to protect the trained model during training and inference.

CHAPTER 2

LITERATURESURVEY

2.1 SECURE AND ROBUST MACHINE LEARNING FOR HEALTHCARE SURVEY

Qayyum, Adnan, Junaid Qadir, Muhammad Bilal, and Ala Al-Fuqaha

This paper presented an overview of various application areas in healthcare that leverage such techniques from security and privacy point of view and present associated challenges. In addition, we present potential methods to ensure secure and privacy-preserving ML for healthcare applications. Present a comprehensive survey of existing literature on the security and robustness of ML/DL models when used for building healthcare systems with a specific focused on the above-mentioned dimensions. Here note that the aim of this work is to provide an in-depth survey of various security challenges associated with the application of ML/DL in healthcare systems and to provide taxonomy of potential solutions to overcome these issues. Along with discussing security and robustness challenges of using ML/DL models, also briefly elaborate on various general challenges and sources of vulnerabilities that hinder the safe and robust application of ML/DL in healthcare applications. The ML techniques utilizing unlabelled data are known as unsupervised learning methods. Widely used examples of unsupervised learning methods are a clustering of data points using a similarity metric and dimensionality reduction to project high dimensional data to lower dimensional subspaces. The use of machine learning (ML)/deep learning (DL) models for clinical applications has great potential to transform traditional healthcare service delivery.

Advantage:

- Easily detects the objects

Disadvantage:

- Drop - outs algorithm to mitigate this problem is difficult

2.2 A LIGHTWEIGHT CHAOS-BASED MEDICAL IMAGE ENCRYPTION SCHEME USING RANDOM SHUFFLING AND XOR OPERATIONS

Masood, Fawad, Maha Driss, Wadii Boulila, Jawad Ahmad, Sadaqat Ur Rehman, Sana Ullah Jan, Abdullah Qayyum, and William J. Buchanan.

This paper presented a lightweight cryptosystem based on Henon chaotic map, Brownian motion, and Chen's chaotic system to encrypt medical images with elevated security. Here designing an effective multi-stage cryptographic algorithm for medical images encryption using substitution-permutation technique. This multi-stage cryptographic algorithm uses random numbers generated from chaos maps which reduces correlation among the pixels of the digital medical images. Then design a contemporary variant of the chaos-based confusion-diffusion approach that is capable of achieving significant higher entropy and NIST-based randomness results as compared to existing methods. The results demonstrate that the proposed encryption algorithm is able to generate highly secured medical encrypted images. An enhanced image encryption scheme is proposed that combines chaos theory with Brownian motion (BM) and Chen's chaotic system (CCS) to achieve the desired level of security in storage systems of hospitals and medical centers. The proposed system achieves confusion through two-dimensional Henon chaotic map (HCM), whereas difusion is obtained using BM and CCS. Furthermore, the reliability and security of the proposed system are analyzed and compared with existing techniques using the following parameters. The NIST and entropy measures are obtained through randomness test, the consistency and variance through histogram examination, and the pixel similarity using a coeffcient of correlation.

Advantage:

- It can protect data and communication from unauthorised disclosure and access.

Disadvantage:

- High Cost Implementation

2.3 LIGHTWEIGHT ENCRYPTION TECHNIQUE TO ENHANCE MEDICAL IMAGE SECURITY ON INTERNET OF MEDICAL THINGS APPLICATIONS

Hasan, Mohammad Kamrul, Shayla Islam, Aisha-Hassan Abdalla Hashim, Shabana Habib, Mohammad Islam

This paper presented an efficient, lightweight encryption algorithm for providing secure image encryption in healthcare industry. The proposed lightweight encryption technique employs two permutation techniques to secure medical images. Besides, picture-based information requires more exertion during encryption and decryption. A change procedure dependent on the mix of picture stage and a recently evolved encryption calculation called ‘‘Hyper Image Encryption Algorithm (HIEA)’’. From the chose picture, we will utilize the twofold worth squares, which will be reworked into a permuted picture using a change procedure, and afterward, the produced picture will be encoded utilizing the ‘‘Hyper Image Encryption Algorithm (HIEA)’’ calculation. All the current strategies utilizing the reasonable client characterized key is created with a similar goal. Likewise, separate between them with a proposed calculation utilized for encryption and decryption. For entropy esteem, connection worth, and execution time of the known cryptographic calculation with proposed cryptography calculations. The proposed lightweight encryption algorithm focused on the efficiency and security of the medical images on IoMT application. The proposed algorithm considered the performance matric of entropy, as well as correlation. The study has found that the current methods generated key based unsystematic sequence number that creates an enormous computation time. In comparison, it is evident from the result that the proposed algorithm has a small computation.

Advantage:

- Relatively Quick

Disadvantage:

- Increased Damage if Compromised

2.4 A NEW IMAGE ENCRYPTION ALGORITHM FOR GREY AND COLOR MEDICAL IMAGES

Kamal, Sara T., Khalid M. Hosny, Taha M. Elgindy, Mohamed M. Darwish, and Mostafa M. Fouda.

This paper presented a new encryption algorithm for encrypting both grey and color medical images. A new image splitting technique based on image blocks introduced. Then, the image blocks scrambled using a zigzag pattern, rotation, and random permutation. Then, a chaotic logistic map generates a key to diffuse the scrambled image. Different algorithms for securing medical images are introduced, yet they may be liable to attacks. A strong correlation between neighboring pixels characterizes medical images; thus, removing this correlation requires a permutation (scrambling) technique with a higher security level. Here presents a new algorithm for encrypting medical images that include four parts: image splitting, image scrambling, key generation, and diffusion. First, the plain image is divided into blocks and sub-blocks using a new image splitting technique. Second, the pixels' arrangement is changed in the blocks and sub-blocks using a zigzag pattern, rotation at a 90-degree angle, and random permutation between blocks. Third, a key is generated from the logistic map, where the map's initial condition depends on the plain image. Finally, image pixel values are changed using the secret key. A new technique for image splitting is proposed. Random permutation between blocks is applied, and pixels substitution in each block is performed to remove the correlation between pixels. Therefore, the proposed algorithm is robust against differential attacks. Analysis of the results proves that our algorithm gains a high performance in encrypting medical images than other methods.

Advantage:

- It provides more secure and confidentiality

Disadvantage:

- Lost the password unable to decrypt the data

2.5 ROBUST DETECTION OF ADVERSARIAL ATTACKS ON MEDICAL IMAGES

Li, Xin, and Dongxiao Zhu.

This paper present an efficient unsupervised learning approach that helps to detect adversarial attacks on medical images. Proposed approach is capable of detecting a wide range of adversarial attacks without knowing the attackers nor sacrificing the classification performance. More importantly, this approach can be easily embedded into any deep learning-based medical imaging system as a module to improve the system's robustness. This proposes to augment the medical image classification system with an adversarial image detection module. Proposed framework of the chest X-ray disease classification system equipped with our detection module. After training the CNN classifier with all clean images to extract the high-level features for learning the detection module, the lower panel illustrates the process of detection and testing. Given a new (clean or adversarial) image, the system extracts features using the trained CNN classifier as the input of detection module. The input image is rejected if detected as an adversarial image, otherwise, it continues to the loss layer to predict classification labels. This strategy does not need any prior knowledge of attack methods nor modification of the CNN architecture. Finally evaluate the performance of our method under both white-box and black-box settings using a benchmark chest X-ray dataset. This effective strategy can be combined with other defense methods and is sufficiently flexible for many medical imaging applications with diverse image formats.

Advantage:

- It requires small amount of memory

Disadvantage:

- Sensitive to handle the noisy dataset

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Designing and implementing FL models with an emphasis on privacy, tailored specifically for healthcare applications, particularly focusing on disease prediction and patient risk profiling. Assessing the real-world implications and potential challenges associated with the deployment of FL models in healthcare settings, along with providing feasible recommendations to circumnavigate identified hurdles. Medical image based object detection has been considered to be an ideal approach to assist medical diagnosis. Doctors can utilize the automatic detection results of medical images to obtain further insights into the patient-specific pathological features and make a more accurate diagnosis. For medical image privacy, current research mostly concentrates on data storage privacy and cannot support online calculations. The problem of the method is that when applying the medical image data into Faster RCNN, we still have to query and download the data to a local server, which can dramatically reduce data availability and computational efficiency. However, the accuracy reduction caused by the introduction of random perturbations is quite considerable. CNN allows multiple healthcare centers to securely share their medical image data and collaborate to build a high-performance Faster CNN model to assist in clinical diagnosis. During the cooperation process, no healthcare center has to worry about their own data revealed to other healthcare centers or the cloud server.

3.1.1 Demerits

Traditional systems typically do not send real-time notifications or alerts when movement or suspicious activity is detected.

- Medical datasets may come from different sources, and they might

have variations in terms of imaging equipment, resolution, and data formats.

- High communication overhead may lead to latency issues, especially in scenarios with slow or unreliable network connections.
- Federated learning systems are susceptible to security threats such as model poisoning attacks or malicious clients.

3.2 PROPOSED SYSTEM

The envisioned system of federated learning-based ocular disease diagnosis is meant to guarantee secure, correct, and decentralized medical image analysis. The workflow starts with the collection of data sets, whereby medical imaging data is collected from different healthcare facilities while adhering to privacy legislation and ethical considerations. The acquired data is processed through preprocessing techniques like standardization and anonymization in order to have consistency between heterogeneous data sets. The module for model building uses Convolutional Neural Networks (CNNs) for diagnosis of disease, utilizing federated learning to train models locally in healthcare facilities without sharing sensitive patient information. Individual local servers fine-tune the CNN model with their dataset, enabling adaptation to particular imaging modalities and demographic differences. For security purposes, the encrypted model storage module encrypts the trained CNN model using Elliptic Curve Cryptography (ECC) prior to transmission and storage to guarantee data confidentiality and integrity. The query processing module receives incoming requests for disease diagnosis, directing queries to suitable local servers depending on patient data privacy preferences as well as geographical locations. When retrieved, the model decryption module decrypts the CNN model with ECC, checking its integrity prior to disease prediction. The disease prediction module predicts diseases from medical images uploaded by patients, returning diagnostic results with confidence scores to support decision-making by healthcare professionals. Lastly, the appointment system schedules consultations according to predicted diagnoses, providing automated

reminders and notifications to facilitate patient care. By combining federated learning with secure encryption and AI-powered predictive diagnostics, this system improves privacy-preserving medical imaging analysis and expands accessibility and accuracy for ocular disease detection.

3.2.1 Merits

- Enhanced security with real-time alerts.
- Better user control and flexibility through remote monitoring.
- Additive learning enhances model performance by integrating new data iteratively, leading to better IDCIS (Intelligent Dynamic Context-Informed System) predictions.
- The VGC (Vision-Guided Convolution) architecture adapts to various input complexities, allowing seamless scalability for real- time environments.
- Additive learning allows the model to update without retraining from scratch, saving computational resources and time.
- IDCIS allows continuous context-aware updates, maintaining system accuracy even when user behavior or environmental conditions change.
- Intelligent prediction using VGC reduces noise and irrelevant detections, lowering the number of false alarms in automated monitoring systems.
- The modularity of additive learning and lightweight VGC models enables easy integration into edge devices with limited processing power.

CHAPTER 4

SYSTEM SPECIFICATIONS

4.1 HARDWARE SPECIFICATION

- Processor : Intel processor 2.6.0 GHZ
- RAM : 1GB
- Hard disk : 160 GB
- Compact Disk : 650 Mb
- Keyboard : Standard keyboard
- Monitor : 15 inch color monitor

4.2 SOFTWARE SPECIFICATION

- Server : Flask, FastAPI ,PySyft
- Python libraries : Tensorflow Federated, HTTP
- Cloud Tools : AWS,GCP,AZURE
- Storage : EC2 ,S3
- Autoscaling : Kubernetes ,Docker
- Evaluation tool : Aequitas

4.2.1 Software Environment

The federated learning system for eye disease prediction operates in a hybrid environment comprising client devices (mobile or edge devices) and a central cloud-based aggregator. Python is the primary language, with frameworks like TensorFlow Federated (TFF) and Flower to handle FL orchestration. Medical image processing is managed using OpenCV and Albumentations, while EfficientNet serves as the core CNN model. Data privacy is ensured with PySyft, implementing differential privacy and secure aggregation. The cloud backend uses Google Cloud Platform for compute and model hosting, while Firebase manages user metadata. Visual experiment tracking is handled by Weights & Biases (wandb). The user interface is developed using Kivy for cross-platform deployment, enabling seamless communication between users and the FL framework. This environment

ensures privacy-preserving, real-time collaborative learning, even in low-resource regions, by training models locally and sharing only secure updates.

4.2.1 Library

- **Federated Averaging** package aggregates local model updates (weights) from clients by computing a weighted average, improving global model

- **Convolutional Neural Network**s is deep learning architecture used to analyze retinal images. Extracts features like blood vessels, optic disc, and lesions through layered convolutions.
- **EfficientNet** is a CNN model optimized for mobile/edge devices. Scales width, depth, and resolution uniformly, making it ideal for resource-constrained FL clients.
- **FedProx** introduces a proximal term in the loss function to stabilize training when clients have non-IID (non-identically distributed) data. This helps reduce performance degradation when data across clients (e.g., retinal images from different hospitals) is highly variable in quality or distribution.

- **Homomorphic Encryption** is a form of encryption that allows computations directly on encrypted data. In FL, it ensures model updates remain secure during transmission—especially important for sensitive medical data—by preventing the central server from accessing individual updates in plaintext.

- **FedMA (Federated Matched Averaging)** matches and merges neural network layers across clients before averaging, improving performance when model architectures are complex.
- **FedBN (Federated Batch Normalization)** allows each client to maintain its own batch normalization parameters while sharing other model parameters, improving performance on non-IID data.

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECHTURE

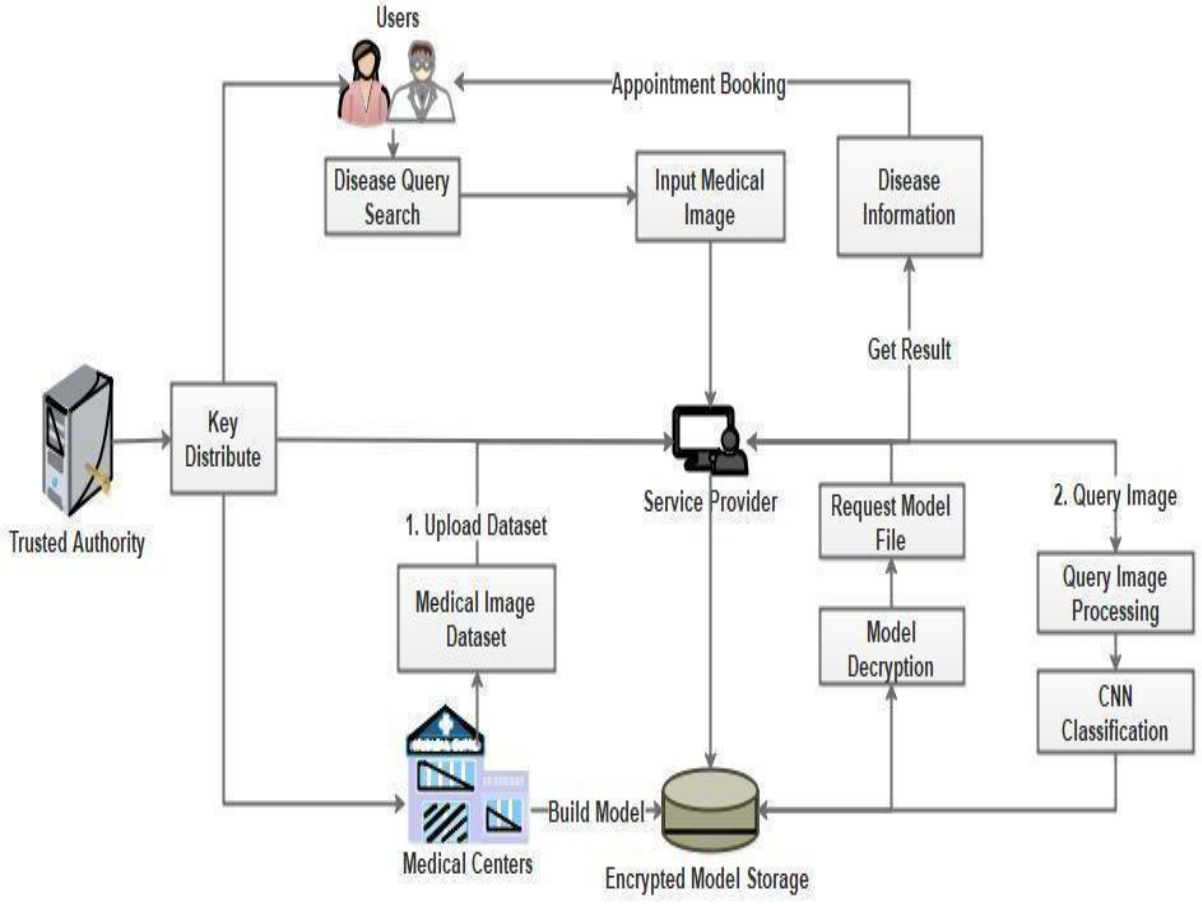

Fig.5.1 System Architecture

The diagram represents the workflow of a federated learning-based ocular disease diagnosis system incorporating Convolutional Neural Networks (CNNs) for medical image analysis. It illustrates interactions between key components, ensuring secure model training, encryption, and disease prediction while maintaining patient data privacy. The process begins with users submitting disease queries and uploading medical images. A trusted authority oversees security by distributing cryptographic keys to healthcare institutions and the service provider. Healthcare centers provide medical image datasets, which are uploaded for CNN model training within a federated learning framework. Instead of sharing raw patient data, each institution independently refines the model based on localized datasets.Once trained, the CNN model is encrypted using Elliptic Curve Cryptography (ECC) and stored securely. Upon receiving a query, the system processes the medical image using CNN-based classification to identify ocular diseases. Encrypted models are requested and decrypted, ensuring data integrity and confidentiality during transmission. The final disease diagnosis is provided to the user along with a confidence score, allowing remote and secure access. Additionally, the appointment system schedules consultations based on predicted diagnoses, enabling timely medical intervention. This workflow ensures a decentralized, secure, and collaborative approach to disease diagnosis while enhancing accuracy and preserving patient privacy.

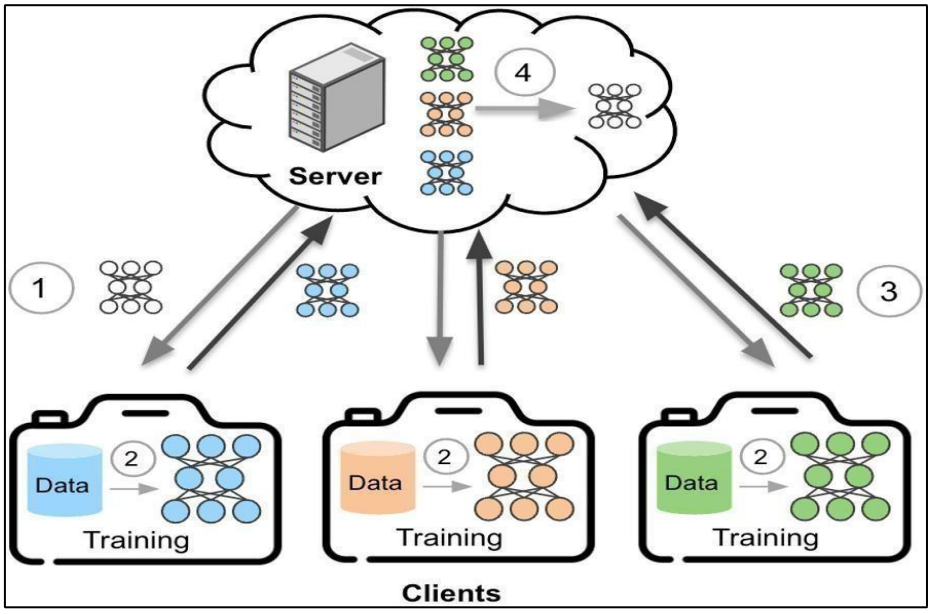

Fig.5.2 Server Client Training

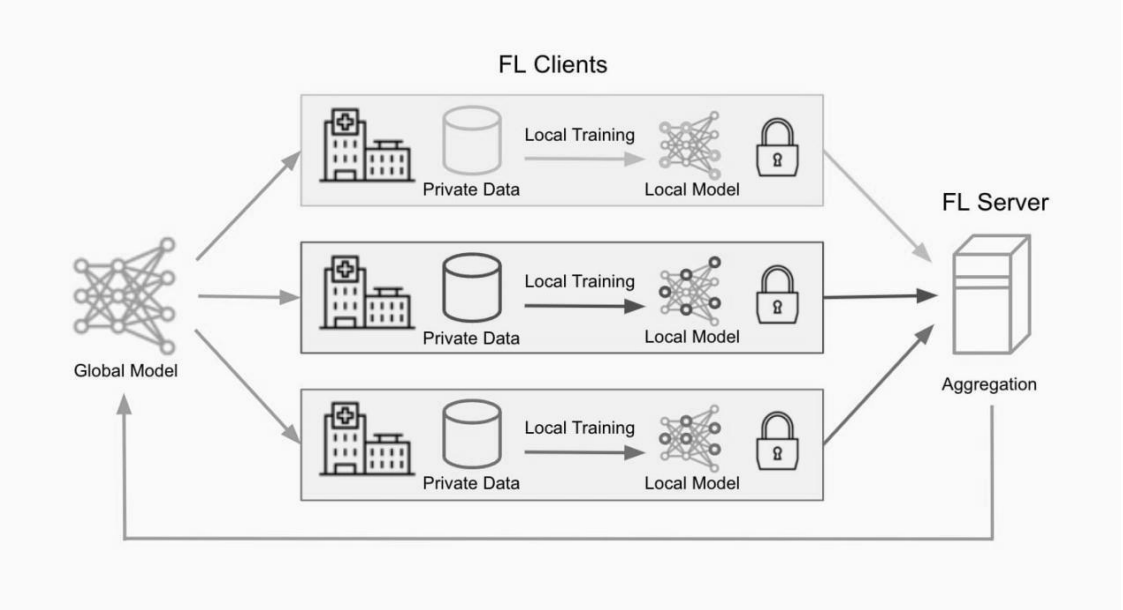

Fig.5.3 Global Model Aggregation

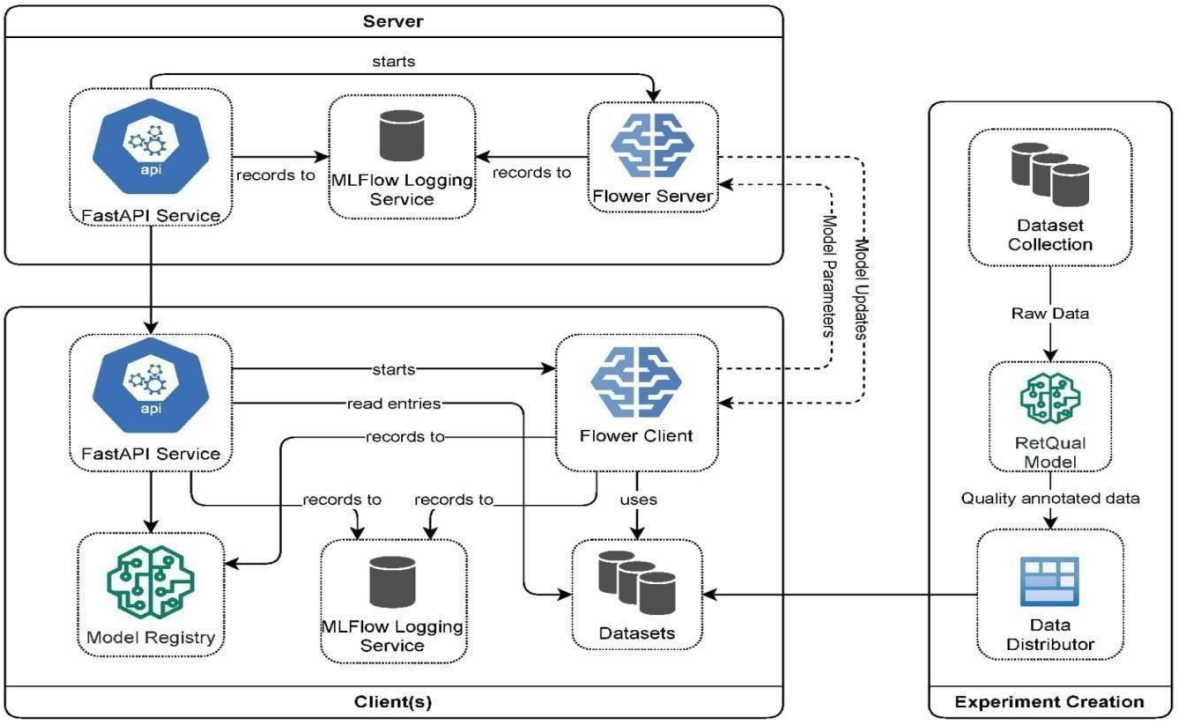

Fig. 5.4 Cloud Integration With Mlop Flow

CHAPTER 6

MODULE DESCRIPTION

LIST OF MODULES

- Dataset Collection
- Model Building
- Encrypted Model Storage
- Query Processing
- Module Decryption
- Disease Prediction Module
- Appointment Booking System

6.1 DATASET COLLECTION

The Dataset Collection Module serves as the foundation for the project, responsible for the acquisition of diverse medical imaging datasets from multiple healthcare institutions. Ensuring adherence to privacy regulations and ethical standards, this module meticulously gathers anonymized patient data while maintaining the integrity and confidentiality of sensitive information. Preprocessing techniques are employed to standardize and anonymize the collected datasets, preparing them for subsequent model training.

The dataset used for detecting diabetic retinopathy and diabetic macular edema has been procured and assembled from two datasets. These datasets are available publicly on the Kaggle repository for machine learning enthusiasts. The Kaggle repository contains many datasets on ocular diseases from which two have been taken. The first one is the Indian Diabetic Retinopathy dataset which contains coloured fundus images of patients' left and right eyes. These images are labelled into three types: diabetic retinopathy, diabetic macular edema and normal. The second is the Aptos blindness detection 2019 dataset from which samples of diabetic retinopathy and normal images of ocular diseases have been taken and pooled. The dataset thus formed contains a total of 838 images, 278 images are of the diabetic

macular edema class and 280 images each of diabetic retinopathy and normal class. In machine learning, data preprocessing entails converting unstructured data into a format that can be utilized to create and enhance machine learning models. The initial stage in machine learning before building a model is data preprocessing. Data preparation is crucial in increasing data reliability to extract valuable information. Actual data is frequently unreliable, inaccurate (contains outliers or errors), incomplete, and devoid of particular attribute values or patterns. In this situation, data preparation is essential because it makes it easier to organize, filter, and present raw data in a format that machine learning models can use. The min-max normalization technique is used for data preprocessing Min-Max Normalization: The low variance, the ambiguous dataset is structured, and data integrity is maintained using min-max scaling for normalizing the features. A model that relies on the magnitude of values has to scale the input attributes. Because of this, normalizing describes the discrete range of real-valued numerical properties between 0 and 1.,is being utilized to normalize the data. After preprocessing, the dataset is split into two sections: a training dataset and a testing dataset, with 25% of the dataset used as testing data to assess the proposed model and 75% used to train the model.

Feature Extraction: The ocular disease recognition dataset is trained after data preprocessing. The VGG16 deep neural network model extracts the critical features without human oversight. Convolutional filters extract features from the training dataset following the benefit of deep learning. The VGG16 deep neural network model is used in this study to classify different ocular types and extract the finer details from an image. The extracted features are then sent to a VGG16 model with a fully connected (FC) layer in the deep neural network model.

- This module is responsible for collecting medical imaging datasets from various healthcare institutions.
- It ensures compliance with data privacy regulations and ethical guidelines.
- Data preprocessing techniques may be applied to standardize and anonymize the collected datasets.

6.2 MODELBUILDING

At the heart of the project lies the Model Building Module, this harnesses the power of Convolutional Neural Networks (CNNs) to construct a robust disease diagnosis model. Employing federated learning methodologies, this module facilitates collaborative model training across decentralized local servers. Each server autonomously refines the model using its unique dataset, thereby accommodating local variations and enhancing the model's adaptability to diverse healthcare settings and imaging modalities.

A federated learning system comprises a client(s) and a server. The cloud-based federated learning server analyses key data types for the target application and trains hyper-parameters like learning rate, number of epochs, activation function, and training the VGG16 deep neural network model at the second level is required.

Every client starts by acquiring updated information and modifying the (Myx) local model parameter, which depends on the global model (Gy), where y is the index for the next iteration. Each client seeks for the optimal situation to minimize the loss. Ultimately, provide the federated learning server with the new parameters regularly. The global model’s integration is the third level. Once the results from various clients have been combined on the server side, send the updated parameters to each client.

The ederated learning system comprises a client(s) and a server. The cloud-based federated learning server analyses key data types for the target application and trains hyper-parameters like learning rate, number of epochs, activation function, and Adam optimizer. Three crucial steps are included in the federated learning paradigm. The initialization of training is the first step. Additionally, a global model is first developed by the federated learning server.

It is important to note that the federated learning server determines the model’s epoch and learning rate. Training the VGG16 deep neural network model at the second level is required.

The VGG16 deep neural network model has specified client requirements and multiple hyper-parameters. It is important to note that the federated learning server determines the model's epoch and learning rate. Training the VGG16 deep neural network model at the second level is required. Every client starts by acquiring updated information and modifying the (Myx) local model parameter, which depends on the global model (Gy), where y is the index for the next iteration. Each client seeks for the optimal situation to minimize the loss. Ultimately, provide the federated learning server with the new parameters regularly. The global model's integration is the third level. Once the results from various clients have been combined on the server side, send the updated parameters to each client. The global mean loss function is the primary objective of the federated learning server, which uses Eq. (2).

$$Loss(Gy)=1M \sum x=Mx=1 Loss(Myx) \quad (2)$$

In Algorithm 1, firstly, take the ocular disease recognition dataset Ds as input and predict the cataract. In the second step, the dataset is preprocessed Dp and partitioned as the training dataset for two clients. Then extract the features fe using the VGG16 model and normalize the data using the min-max normalization technique. The dataset is divided between training and testing, with 75% of the data used to train the models and 25% used to test the models. Update the global model and initialize the model weight w0. ti denotes the current round of the model, T is the total round of the local model. ci is the current client, and C is the total clients. Update the local model for each client according to the current iteration/round.

Calculate the current iteration weight by the sum of the weight of the client's dataset and the current client iteration. The following model parameters, such as an epoch value, activation function, and batch size, are used to calculate the loss of the local model of each client. Update the local model by calculating the loss function Fi(w). The procedure is repeated until the requisite accuracy is attained or the loss function is constantly minimized.The VGG16 deep neural network model is used for cataract disease prediction. When simulating the

structure of neural networks, the neurons and the number of layers are essential. The number of neurons utilized as input and output in a deep neural network model depends on the training data size. A deep convolution neural network model with much success in computer vision is the VGG16. The architecture of the VGG16 model is provided in Fig. 3. In general, the VGG16 model has three layers: an input layer, several hidden layers (such as dropout, dense, flatten, etc.), and the output layer. The sequential VGG16 model used in this study has a single input layer. The input layer has a shape of 224 and uses the relu activation function.

The next layer is the hidden layer, consisting of four dense layers and three dropout layers. The three dropout layers are employed to prevent the overfitting of the model. The values of the dropout layers are 0.5, 0.2, and 0.1, respectively. The four dense layers comprise 256 and 128, and 1 unit and the activation functions are the relu and sigmoid. The flattened layer, typically used in the transition from the convolution layer to the fully connected layer, is the next layer to reduce the multidimensional input to one dimension. The output layer comes next; it is the fully connected layer utilized for binary classification problems. The VGG16 model uses binary cross-entropy and Adam as an optimizer to calculate and reduce the loss. To address the binary classification problem, every dense layer uses relu and sigmoid activation functions along with a fully connected layer.

- Utilizes Convolutional Neural Networks (CNNs) to build the disease diagnosis model.
- Implements federated learning techniques to enable collaborative model training across local servers.
- Each local server independently trains the model using its dataset, fostering.

6.3 ENCRYPTED MODEL STORAGE

The Encrypted Model Storage Module plays a pivotal role in ensuring the security and confidentiality of the trained CNN model. Utilizing Elliptic Curve Cryptography (ECC), this module encrypts the model parameters before storage and transmission, safeguarding against unauthorized access and maintaining data integrity. By employing robust encryption mechanisms, sensitive patient information remains protected, mitigating the risk of breaches and preserving privacy.

In FL, only the model weights or gradients are exchanged between client devices and a central aggregator. However, even these gradients can be reverse-engineered to extract sensitive information about the training data. Therefore, robust encryption mechanisms are necessary to secure model updates and prevent data leakage.

Types of threats addressed by encryption in FL:

- Gradient leakage attacks: Where adversaries try to reconstruct private data from gradients.
- Eavesdropping: Where attackers intercept data during transmission.
- Model poisoning: Malicious clients may inject harmful updates to manipulate the global model.

a) Homographic Encryption(HE)

Homomorphic encryption allows mathematical operations to be performed on encrypted data without decrypting it. This means that clients can encrypt their model updates before sending them to the server, which can then perform aggregation on the ciphertexts.

Use in Eye Disease Prediction: Each hospital encrypts its model parameters using HE and sends them to a central server. The server performs federated averaging on the encrypted weights and sends back an encrypted global model.

Advantages: Strong security; data never exposed in plaintext.

Disadvantages: High computational overhead; slower training cycles.

b) Secure Multi-Party Computation (SMPC)

SMPC enables multiple parties to jointly compute a function over their inputs while keeping those inputs private. In FL, this can be used to ensure that no single party learns the full model update from others.

Use in Eye Disease Prediction: Client devices split their model updates into shares and distribute them among other clients or secure computation servers. These shares are used to compute the aggregated model without revealing individual updates.

Advantages: High privacy guarantee.

Disadvantages: Requires a complex infrastructure; synchronization needed.

c) Differential Privacy (DP)

DP ensures that the removal or addition of a single data point does not significantly affect the outcome of the computation. In FL, noise is added to model updates to obfuscate specific data characteristics.

Use in Eye Disease Prediction: Before transmitting the model gradients, each client adds calibrated noise to protect individual patient information.

Advantages: Lightweight compared to HE and SMPC.

Disadvantages: Potential trade-off with model accuracy.

The model storage module in an FL system is critical for managing versions of local models, encrypted updates, and the global model. This module ensures synchronization, rollback capability, and consistency between clients and the central server.

a) Local Storage at Clients

Each client stores:

- Local model weights before and after training.
- Encrypted gradients or model updates.
- Logs and training metadata for audit and reproducibility.

For resource-constrained devices (e.g., medical IoT systems), lightweight storage formats like TensorFlow Lite or ONNX are preferred. Models are often compressed using quantization techniques to reduce memory overhead.

b) Central Aggregator Storage

The server or aggregator maintains:

- The global model state across training rounds.
- Encrypted update logs from each client (useful for debugging or anomaly detection).
- A versioning system to manage rollback in case of corrupted updates or model poisoning.
- Storage systems often use secure cloud infrastructure with end-to-end encryption and access controls (e.g., AWS S3 with encryption keys managed via AWS KMS or Azure Blob Storage with RBAC).

During each training round:

- Clients download the latest global model.
- They train locally on private data and apply encryption (HE/DP).
- Encrypted updates are transmitted back to the server.
- The server aggregates the updates (e.g., using Federated Averaging).
- The new global model is encrypted and stored securely for distribution in the next round.
- To enhance performance, asynchronous FL is also being explored, where clients can update the model without waiting for all others, improving efficiency in real-time diagnosis

6.4 QUERY PROCESSING

The Query Processing Module acts as the interface between users and the distributed network of local servers. It handles incoming diagnostic queries, routing them to the appropriate local server based on user preferences and geographical considerations. This module ensures efficient allocation of computational resources while prioritizing patient privacy and data locality, facilitating seamless interaction with the federated learning framework.

1. Query Generation

The query begins when a user (a healthcare practitioner, radiologist, or diagnostic technician) inputs data into the system. This input is typically a high-resolution retinal or fundus image, possibly accompanied by metadata like patient age,

sex, and previous conditions. In real-world deployments, this data is captured using imaging devices connected to an edge device (e.g., laptop or local server) in a hospital or clinic. Instead of sending the entire image to a remote server, the system queries the locally stored global model, which has been updated through previous federated learning cycles. This model is capable of detecting diseases such as cataracts, diabetic retinopathy, glaucoma, and other ocular conditions. The query is now passed to the preprocessing pipeline.

2. Input Preprocessing

To ensure consistency and accuracy, the input query (image and metadata) must be preprocessed to match the training data format. This includes:

- Resizing the image to fit the input dimensions required by the neural network (e.g., 224×224 for VGG16).
- Min-max normalization is applied to scale pixel intensity values between 0 and 1, which improves model convergence and reduces computational cost.
- Noise filtering and enhancement methods (such as histogram equalization) may be applied to enhance clarity, especially in poor lighting conditions.
- Augmentation (optional): If the model supports uncertainty estimation, the image may be rotated, flipped, or color-shifted slightly to generate multiple variants for robust prediction.
- Once preprocessed, the query is fed into the locally stored federated model.

3. Local Model Inference

At this point, the query processing engine loads the VGG16-based deep neural network that was trained using federated learning. This model has been previously downloaded from the aggregator and stored locally in a lightweight format (e.g., TensorFlow Lite, ONNX, or PyTorch Mobile). Inference begins by passing the preprocessed image through multiple convolutional layers. The system extracts hierarchical features like edges, textures, and structural patterns of the retina and lens. These features are compared against learned patterns corresponding to various diseases:

- In the case of cataracts, the model identifies opacities or cloudiness.

- For diabetic retinopathy, it looks for microaneurysms or hemorrhages.
- For glaucoma, optic nerve cupping and other changes are examined.

The output layer of the model provides a multi-class classification score (e.g., probability of presence of each disease class). In some cases, Grad-CAM or other explainability tools may highlight regions of interest on the image to assist the medical professional in understanding the model's focus.

4. Encryption and Privacy Checks

Even though inference is done locally, federated learning models often include privacy-preserving features during deployment to ensure that no data leaks occur, even during query response handling. The results are encrypted using symmetric or asymmetric cryptography before being transmitted (if required) to central dashboards or cloud archives. Differential privacy techniques may be used to obfuscate output metadata if the results are aggregated later for retraining or performance monitoring. If model improvements are scheduled via continual FL learning, the gradients or updates computed from the query are temporarily stored in encrypted form and shared securely in the next federated training round.

5. Result Interpretation and Response Generation

Once the inference is complete, the system interprets the prediction output and formats the result for delivery. This step involves:

- Mapping the prediction scores to disease categories (e.g., Cataract: 95.2%, Normal: 3.0%, DR: 1.8%).
- Generating a textual or graphical report for the physician.
- Optionally overlaying heatmaps or attention visualizations on the input image using saliency maps or LIME explanations.
- Storing the result locally or in a secure cloud-based EHR (Electronic Health Record) system with appropriate encryption.
- The clinician now views the result and uses it as a support tool for diagnosis, referral, or treatment.

6. Continuous Learning and Feedback Loop

Post-query, the system may allow clinicians to provide feedback—whether the prediction was accurate, uncertain, or incorrect. These flags are recorded and used in subsequent federated learning cycles to improve model performance. The feedback mechanism includes:

- Tagging misclassified images for further review.
- Flagging edge cases for retraining.
- Storing metadata securely for future model refinement.
- This feedback is optional but instrumental in boosting the model's reliability and generalization capability across diverse populations.

6.5 MODEL DECRYPTION

Upon retrieval of the encrypted CNN model from the central server, the Model Decryption Module decrypts the model parameters using ECC encryption. In federated learning systems designed for sensitive healthcare applications like eye disease prediction, model decryption is a critical component that bridges the gap between secure training and effective deployment. Since federated learning avoids centralizing raw patient data, it relies on a system where each client or hospital downloads a globally trained encrypted model to perform local inference or continue the training process. The encrypted model, often shared by a central aggregator after aggregating the parameters from multiple clients, must undergo secure decryption before it can be utilized locally. This is particularly important in the diagnosis of eye-related conditions like cataracts, diabetic retinopathy, or glaucoma, where the confidentiality of patient records is paramount and even model misuse could potentially leak sensitive patterns.

Decryption ensures that the received model is usable, trustworthy, and matches the structure of the local system while maintaining compliance with privacy protocols. In this project, after using federated learning to train a VGG16 deep convolutional neural network across multiple clients with the ODIR dataset, the final global model is encrypted using cryptographic techniques such as homomorphic encryption or secure multiparty computation (SMPC)before being broadcasted. These encryption techniques allow training to occur on encrypted weights without ever revealing raw

model parameters during aggregation. However, for a client to use the model for real-time predictions or continued local training, it must decrypt the model using a private key or a secure decryption protocol. The decryption is performed locally on the client-side system using pre-shared keys or secure hardware modules like Trusted Platform Modules (TPMs) or Hardware Security Modules (HSMs). In scenarios where asymmetric encryption is used (e.g., RSA or ECC), the aggregator encrypts the model with a client's public key, and only the respective client can decrypt it using its private key. This approach ensures that even if an unauthorized user intercepts the model in transit, it remains unreadable and unusable. Once decrypted, the model parameters are restored to their usable form—floating-point weight matrices and activation thresholds—which are then loaded into a local execution environment such as TensorFlow Lite, PyTorch Mobile, or ONNX Runtime. This decryption process must be both secure and efficient, especially because clients may be edge devices with limited computing power, such as diagnostic workstations or mobile retinal scanners. To facilitate this, lightweight cryptographic libraries and optimized decryption pipelines are used that reduce overhead while still ensuring high security. Furthermore, during decryption, model integrity is verified using hash checks, digital signatures, or blockchain verification (if used).

This is to confirm that the model has not been tampered with and corresponds exactly to the aggregated version intended by the server. If the hash does not match, the decryption process halts, and a security alert is triggered, which is vital in medical contexts where wrong predictions could affect lives. Once decrypted and verified, the model is temporarily cached in encrypted local memory or sandboxed environments to prevent unauthorized access by background applications. The decrypted model then serves two primary purposes: real-time inference for patient diagnosis and continued local training (optional) in adaptive federated learning setups. Inference involves loading the decrypted VGG16 model into memory, feeding in preprocessed patient eye images (normalized using min-max scaling), and generating classification scores indicating the probability of specific ocular diseases. If adaptive training is enabled, the decrypted model also accepts feedback labels from the practitioner and continues refining on the local dataset before encrypting updates and sending them back to the aggregator. This ensures the global model evolves without ever accessing

raw data. It is also essential to understand the data-flow control during and after model decryption. The decrypted model must not persist indefinitely in memory or on disk; hence, automatic cleanup scripts or ephemeral containers are deployed to delete the decrypted weights once the session ends. In high-risk deployments, zero-trust policies are enforced, where even administrators cannot view decrypted models without secure authentication.

Additionally, differential privacy algorithms are embedded into the decrypted model's architecture to ensure that inferences made from sensitive patient images do not accidentally leak information through side channels like model inversion or membership inference attacks. This means even after decryption, privacy remains a first-class citizen. In some advanced setups, model decryption and execution happen simultaneously using secure enclaves or confidential computing frameworks like Intel SGX or ARM TrustZone, which create a hardware-isolated region where both decryption and inference can occur securely, further minimizing exposure. These modules allow the decrypted model to run entirely inside a protected memory space, invisible to the host operating system or external users. Given that the model achieved over 95% accuracy across multiple rounds in this project, preserving the fidelity of the model during decryption is crucial. Incorrect decryption could not only fail to load the model but also degrade its prediction accuracy due to floating-point mismatches or corrupted weights. Therefore, numerical integrity and reproducibility checks are also part of the decryption process, comparing model outputs on benchmark samples with expected results to ensure consistent behavior. In conclusion, model decryption in federated learning-based eye disease prediction systems is not just a technical step—it is a security-critical gateway that enables protected, ethical, and efficient use of AI models in healthcare environments. By integrating strong cryptography, hardware security, and smart data lifecycle policies, the system ensures that decrypted models empower real-time diagnosis without ever compromising patient trust or data protection regulations. It verifies he integrity of the decrypted model, ensuring that the information remains intact and unaltered. By securely decrypting the model, this module prepares it for utilization in disease prediction, maintaining the confidentiality and integrity of sensitive medical information throughout the process.

- Decrypts the encrypted CNN model using ECC upon retrieval from the central server.
- Verifies the integrity of the decrypted model parameters to ensure data integrity.
- Prepares the decrypted model for disease prediction.

6.6 DISEASE PREDICTION

The Disease Prediction Module serves as the core component for medical imaging diagnosis, leveraging the decrypted CNN model to analyze patient images and provide accurate diagnostic predictions. By harnessing the hierarchical feature extraction capabilities of CNNs, this module identifies patterns indicative of various diseases, enabling precise and reliable diagnoses. Results are delivered with associated confidence scores, empowering healthcare professionals with actionable insights for patient care.

In federated learning systems designed for sensitive healthcare applications like eye disease prediction, model decryption is a critical component that bridges the gap between secure training and effective deployment. Since federated learning avoids centralizing raw patient data, it relies on a system where each client or hospital downloads a globally trained encrypted model to perform local inference or continue the training process. The encrypted model, often shared by a central aggregator after aggregating the parameters from multiple clients, must undergo secure decryption before it can be utilized locally. This is particularly important in the diagnosis of eye-related conditions like cataracts, diabetic retinopathy, or glaucoma, where the confidentiality of patient records is paramount and even model misuse could potentially leak sensitive patterns.

Decryption ensures that the received model is usable, trustworthy, and matches the structure of the local system while maintaining compliance with privacy protocols. In this project, after using federated learning to train a VGG16 deep convolutional neural network across multiple clients with the ODIR dataset, the final global model is encrypted using cryptographic techniques such as homomorphic encryption or secure multiparty computation (SMPC) before being broadcasted. These encryption techniques allow training to occur on encrypted weights without ever revealing raw

model parameters during aggregation. However, for a client to use the model for real-time predictions or continued local training, it must decrypt the model using a private key or a secure decryption protocol. The decryption is performed locally on the client-side system using pre-shared keys or secure hardware modules like Trusted Platform Modules (TPMs) or Hardware Security Modules (HSMs). In scenarios where asymmetric encryption is used (e.g., RSA or ECC), the aggregator encrypts the model with a client's public key, and only the respective client can decrypt it using its private key. This approach ensures that even if an unauthorized user intercepts the model in transit, it remains unreadable and unusable. Once decrypted, the model parameters are restored to their usable form—floating-point weight matrices and activation thresholds—which are then loaded into a local execution environment such as TensorFlow Lite, PyTorch Mobile, or ONNX Runtime.

This decryption process must be both secure and efficient, especially because clients may be edge devices with limited computing power, such as diagnostic workstations or mobile retinal scanners. To facilitate this, lightweight cryptographic libraries and optimized decryption pipelines are used that reduce overhead while still ensuring high security. Furthermore, during decryption, model integrity is verified using hash checks, digital signatures, or blockchain verification (if used). This is to confirm that the model has not been tampered with and corresponds exactly to the aggregated version intended by the server. If the hash does not match, the decryption process halts, and a security alert is triggered, which is vital in medical contexts where wrong predictions could affect lives. Once decrypted and verified, the model is temporarily cached in encrypted local memory or sandboxed environments to prevent unauthorized access by background applications.

The decrypted model then serves two primary purposes: real-time inference for patient diagnosis and continued local training (optional) in adaptive federated learning setups. Inference involves loading the decrypted VGG16 model into memory, feeding in preprocessed patient eye images (normalized using min-max scaling), and generating classification scores indicating the probability of specific ocular diseases. If adaptive training is enabled, the decrypted model also accepts feedback labels from the practitioner and continues refining on the local dataset before encrypting updates and sending them back to the aggregator. This ensures the global model evolves without

ever accessing raw data. It is also essential to understand the data-flow control during and after model decryption. The decrypted model must not persist indefinitely in memory or on disk; hence, automatic cleanup scripts or ephemeral containers are deployed to delete the decrypted weights once the session ends. In high-risk deployments, zero-trust policies are enforced, where even administrators cannot view decrypted models without secure authentication.

Additionally, differential privacy algorithms are embedded into the decrypted model's architecture to ensure that inferences made from sensitive patient images do not accidentally leak information through side channels like model inversion or membership inference attacks. This means even after decryption, privacy remains a first-class citizen. In some advanced setups, model decryption and execution happen simultaneously using secure enclaves or confidential computing frameworks like Intel SGX or ARM TrustZone, which create a hardware-isolated region where both decryption and inference can occur securely, further minimizing exposure. These modules allow the decrypted model to run entirely inside a protected memory space, invisible to the host operating system or external users. Given that the model achieved over 95% accuracy across multiple rounds in this project, preserving the fidelity of the model during decryption is crucial. Incorrect decryption could not only fail to load the model but also degrade its prediction accuracy due to floating-point mismatches or corrupted weights.

Therefore, numerical integrity and reproducibility checks are also part of the decryption process, comparing model outputs on benchmark samples with expected results to ensure consistent behavior. In conclusion, model decryption in federated learning-based eye disease prediction systems is not just a technical step—it is a security-critical gateway that enables protected, ethical, and efficient use of AI models in healthcare environments. By integrating strong cryptography, hardware security, and smart data lifecycle policies, the system ensures that decrypted models empower real-time diagnosis without ever compromising patient trust or data protection regulations.

- Performs disease prediction using the decrypted CNN model.
- Analyzes medical images provided by the user to make accurate diagnostic predictions.

- Returns the predicted diagnosis along with confidence scores or probabilities.

6.7 APPOINTMENT SYSTEM

Finally, the Appointment System Module orchestrates patient management based on predicted diagnoses. It schedules appointments with healthcare professionals for further consultation or treatment, ensuring timely intervention and continuity of care. By integrating with existing healthcare systems, this module enhances patient engagement and streamlines the healthcare delivery process, ultimately improving patient outcomes and satisfaction.

The Appointment Booking Module plays a vital role in the proposed federated learning-based healthcare framework, streamlining the interaction between patients and healthcare providers. Designed with an emphasis on automation, user accessibility, and intelligent scheduling, this module ensures that patients receive timely consultations based on disease prediction outcomes. It bridges the gap between AI-based diagnosis and human-led treatment by allowing patients to seamlessly book appointments with ophthalmologists, general practitioners, or teleconsultants.

1. Purpose and Functionality

The Appointment Booking Module is developed to enhance healthcare accessibility by enabling patients to schedule eye checkups, follow-up consultations, or emergency visits directly through a digital interface. It is especially useful in cases where the disease prediction module detects signs of eye-related conditions that require medical intervention. The booking system can automatically suggest appointments based on:

- Disease severity level
- Doctor availability
- Nearest hospital or clinic location
- Patient preferences (e.g., date, time, and mode of consultation)

The system supports both manual and automated booking. For high-risk cases (e.g., advanced diabetic retinopathy), the module can trigger an auto-scheduling feature, notifying both the patient and the doctor of an urgent appointment suggestion.

2. User Interface and Experience

- The module is accessible through multiple platforms, including:
- Web portals for hospitals or clinics
- Mobile applications for patients
- Embedded systems in kiosks at rural health centers or screening vans

The user interface is designed to be simple and multilingual, ensuring usability across various age groups and literacy levels. Upon logging in, users can:

- View their diagnosis summary from the prediction module
- Choose their preferred doctor or hospital from a list
- Select available dates and times
- Opt for either physical visits or teleconsultation options (video call)
- Receive confirmation messages and reminders via SMS, email, or app notifications.

3. Integration with Disease Prediction Module

A key innovation in this system is its tight integration with the eye disease prediction module. Once a retinal image is analyzed and a disease is detected:

- The patient receives a diagnostic report with a risk level (e.g., mild, moderate, severe).
- Based on this risk level, the system recommends doctors who specialize in the relevant condition.
- Priority slots are automatically opened for high-risk patients to ensure fast-track care.
- This intelligent handoff between prediction and booking ensures minimal delay in diagnosis-to-treatment flow, which is critical for progressive diseases like glaucoma and AMD.

4. Doctor and Hospital Dashboard

On the provider side, doctors and hospital admins have access to a dedicated dashboard where they can:

- Set their availability (time slots, consultation mode)
- Accept or reject appointments

- View patient diagnostic summaries in advance
- Upload prescriptions and consultation notes post-visit

This dashboard allows easy calendar management, especially for specialists who serve in multiple locations or conduct both in-person and online sessions.

5. Backend Logic and Scheduling Algorithms

The scheduling engine of the module is supported by rule-based and AI-driven algorithms that handle:

- Time slot optimization (avoiding overlaps or overbooking)
- Load balancing across doctors in the same facility
- Geolocation-based filtering of nearby clinics
- Priority queues for emergency or high-risk cases
- Conflict resolution and rescheduling suggestions

For example, if a patient's preferred slot is unavailable, the system suggests the nearest available alternative while considering travel distance and urgency.

6. Teleconsultation Support

Recognizing the importance of remote care, especially in rural or underserved regions, the module fully supports teleconsultation. Features include:

- In-app video calling for doctor-patient interaction
- Secure file sharing (reports, prescriptions)
- Session recording and timestamping (optional)
- Integration with EMR (Electronic Medical Records) systems

Patients with internet access can consult with ophthalmologists from home, reducing travel cost and time, especially for follow-ups and non-critical care.

7. Notification and Reminder System

The module includes an automated communication system that keeps all parties informed:

- Patients receive confirmations, appointment reminders, and follow-up prompts.
- Doctors get alerts for upcoming appointments or cancellations.

- Administrators are notified about appointment load and resource utilization.
- Notifications are customizable and can be sent via:
- SMS for non-smartphone users
- App notifications for real-time alerts
- Email for record-keeping and digital confirmations

8. Benefits and Impact

The Appointment Booking Module brings several benefits to the overall healthcare system:

- Time Efficiency: Reduces patient wait times and eliminates manual scheduling errors
- Improved Access: Enables remote users to consult top specialists without traveling
- Resource Optimization: Ensures optimal use of doctor time and infrastructure
- Continuity of Care: Supports follow-ups and longitudinal tracking of patient health.
- Emergency Readiness: Fast-tracks high-risk patients to priority consultation slots

CHAPTER 7

RESULTS AND PERFORMANCE COMPARISON

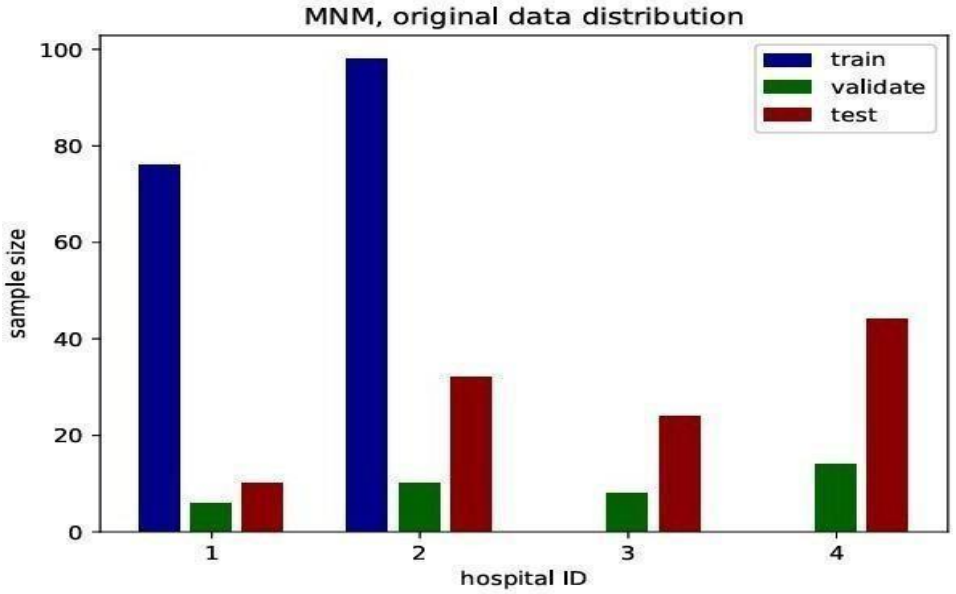

Fig.7.1 Trained Model graph

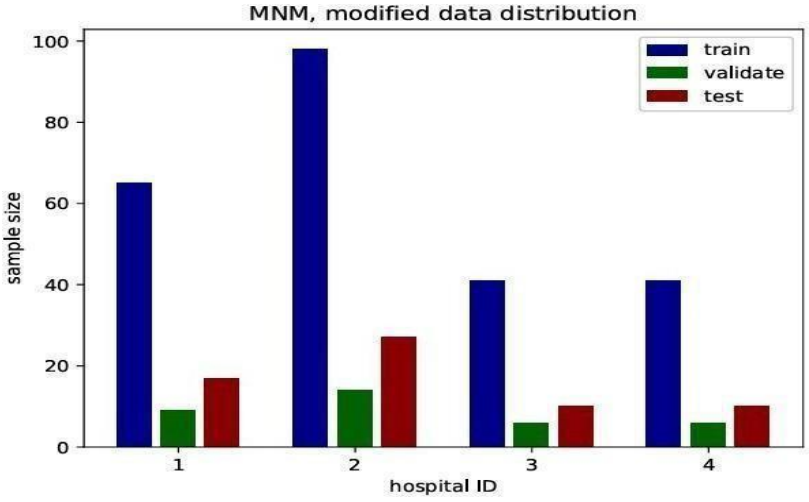

Fig.7.2 Modified Data Graph Results

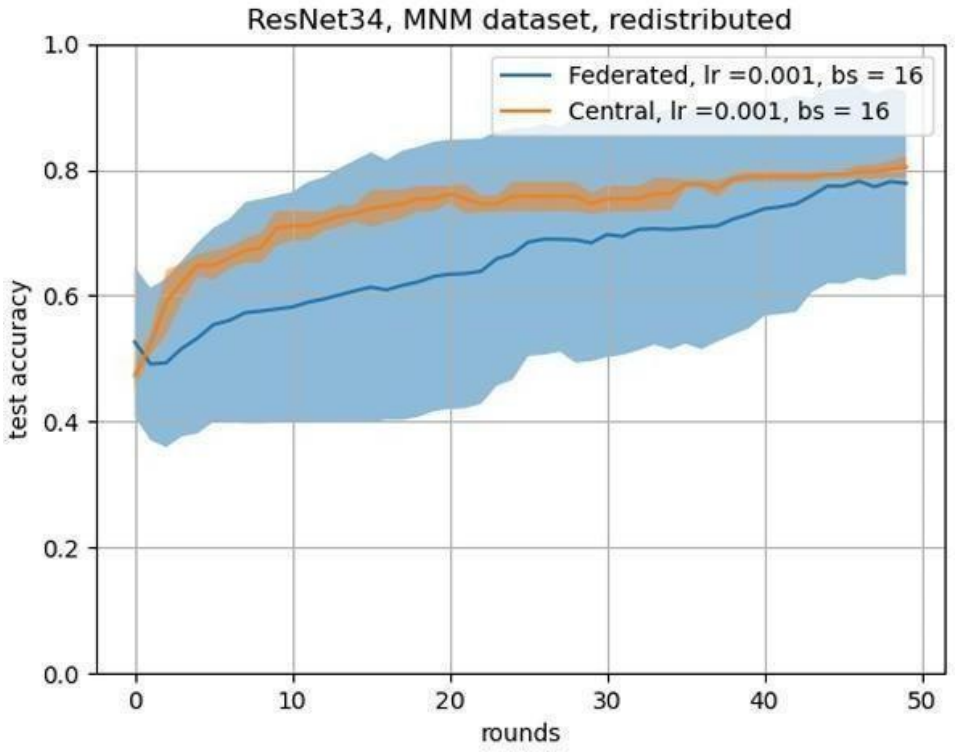

Fig.7.3 Central Data Distribution

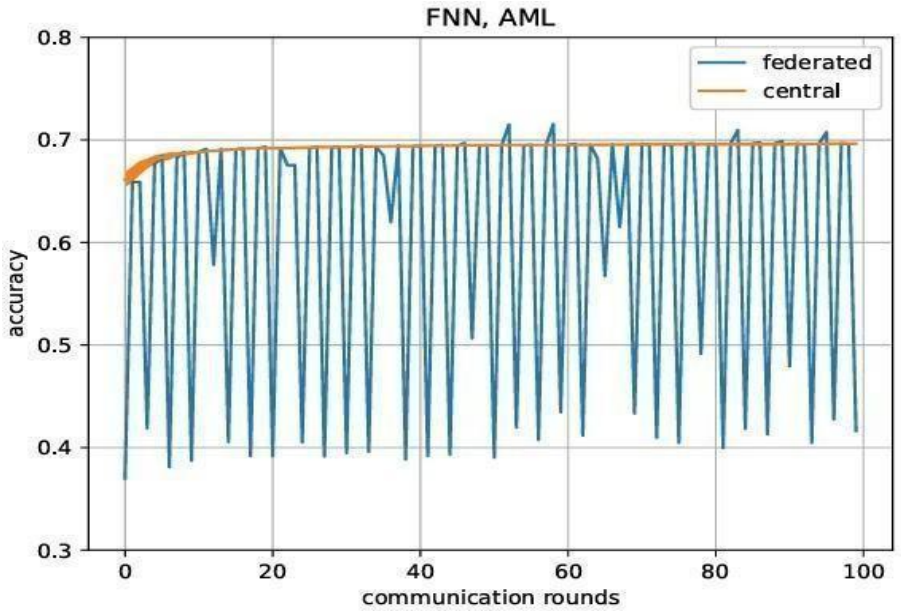

Fig.7.4 Resnet Accuracy Model

Table 1.1 Model Efficiency

Model	Parameters (Millions)	Edge Compatibility	Accuracy on Retinal Data	Inference Speed
ResNet50	25.6	Moderate	85%	Medium
EfficientNet-B0	5.3	High	88%	High
MobileNetV2	3.4	Very High	84%	Very High
VGG16	138	Low	87%	Low

Table 1.2 Privacy Technique Comparison

Privacy Technique	Data Exposure Risk	Computation Overhead	FL Integration Ease	Best Use Case
Differential Privacy	Very Low	Medium	Easy	General patient data privacy
Secure Multiparty Computation	None	High	Medium	Highly sensitive clinical data
Homomorphic Encryption	None	Very High	Complex	Cloud-based secure aggregation
Local Differential Privacy	Very Low	Medium	High	Edge device privacy assurance

Table 1. 3 Communication Efficiency

Technique	Data Size per Round	Accuracy Retention	Client Suitability	Bandwidth Usage
Full Weight Sharing	High	High	High-end clients	High
Federated Distillation	Very Low	Medium	All types	Very Low
Sparse Updates	Low	Medium	Low-resource environments	Low

Table 1.4 Standard Model Comparisons

S.No	Model	Standard Test Accuracy %	Federated Test Accuracy %
1	ResNet fully frozen	88.38	90.52
2	ResNet partially frozen	92.18	95.73
3	VGG fully frozen	88.62	84.59
4	VGG partially frozen	91.70	26

Table 1.5 Accuracy Metrics

Model	Precision %	Recall %	F1-Score %	AUC %
ResNet fully frozen	89.12	88.90	89.01	91.34
ResNet partially frozen	94.87	95.10	94.98	96.45
VGG fully frozen	86.75	85.90	86.32	87.41
VGG partially frozen	24.15	30.40	26.85	28.62

CHAPTER 8
CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION

In conclusion, this project introduces a pioneering framework that revolutionizes disease diagnosis in medical imaging by integrating federated learning, Convolutional Neural Networks (CNNs), and Elliptic Curve Cryptography (ECC) to address critical challenges in healthcare. The collaborative nature of federated learning ensures that model training occurs on local servers, preserving the privacy of sensitive medical image data. The CNN architecture enhances diagnostic accuracy through hierarchical feature extraction, accommodating the heterogeneity present in medical imaging datasets across diverse healthcare institutions. The use of ECC for model encryption provides a robust layer of security, safeguarding both patient data privacy and the intellectual property embedded in the trained model during transmission and storage.

8.2 FUTURE ENHANCEMENT

As a future work the proposed technique can practically be included within the medical information systems to provide medical data integrity, and also implement different access control mechanism. Other revertible watermarking methods can be proposed to increase the amount of embedded data, and other lossless compression methods can be proposed to enhance the ability of the proposed technique to embed larger amount of data.Reversible watermarking techniques can be introduced to embed authentication codes or medical metadata directly into retinal images without compromising their quality. This would allow for traceable and tamper-proof data provenance. Integration with blockchain-based audit trails may also be explored to ensure transparency, accountability, and traceability in federated updates. These enhancements will push the framework closer to deployment in sensitive, high-stakes medical environments.

APPENDIX A

SOURCE CODE

```
from collections import OrderedDict
from typing import Dict, List, Optional, Tuple

import flwr as fl
import numpy as np
import torch
import torch.nn as nn
import torch.nn.functional as F
import torchvision.transforms as transforms
from torch.utils.data import DataLoader, random_split
from torchvision import datasets, utils
torch.manual_seed(42)
DEVICE = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print(f"Training on {DEVICE}")class EyeImageDataset(torch.utils.data.Dataset):
    def __init__(self, root_dir, transform=None):
        self.dataset = datasets.ImageFolder(root=root_dir, transform=transform)
        self.classes = self.dataset.classes

    def __len__(self):
        return len(self.dataset)

    def _getitem_(self, idx):
        return self.dataset[idx]
root_directory = '/kaggle/input/dataset/'

data_transform =
    transforms.Compose([ transforms.ToTensor(),
```

```
    transforms.Resize(size = (224, 224)),
    transforms.RandomRotation(degrees=5),
    transforms.RandomHorizontalFlip(p=0.5),
])

dataset = EyeImageDataset(root_dir=root_directory, transform=data_transform)
train_dataset, test_dataset = tor
num_clients = NUM_CLIENTS

# Split training set into `num_clients` partitions to simulate different local datasets
partition_size = len(train_dataset) // num_clients
lengths = [partition_size]  num_clients
datasets = random_split(train_dataset, lengths, torch.Generator().manual_seed(42))

# Split each partition into train/val and create DataLoader
train_loaders = []
val_loaders = []
for ds in datasets:
    len_val = len(ds) // 10 # 10 % validation set
    len_train = len(ds) - len_val
    lengths = [len_train, len_val]
    ds_train, ds_val = random_split(ds, lengths, torch.Generator().manual_seed(42))
    train_loaders.append(DataLoader(ds_train, batch_size=8, shuffle=True))
    val_loaders.append(DataLoader(ds_val, batch_size=8,shuffle=True))
test_loader = DataLoader(test_dataset, batch_size=8)
import torch
import torchvision.models as models
from torchvision.models import resnet50, ResNet50_Weights, resnet18, ResNet18_Weights
import torch.nn as nn
import tqdm
model = models.resnet50(weights = ResNet50_Weights.IMAGENET1K_V1)
```

```
ct = 0
for child in model.children():
    ct += 1
    if ct < 7:
        for param in child.parameters():
            param.requires_grad = False
def evaluate(model, data_loader, criterion):
    """Evaluate the model on the given dataset."""
    # Set the model to evaluation mode.
    model.eval()
    correct = 0
    val_loss = 0
    count = 0
    # The `torch.no_grad()` context will turn off gradients for efficiency.
    with torch.no_grad():
        for images, labels in (data_loader):
            images, labels = images.to(DEVICE), labels.to(DEVICE)
            output = model(images)
            pred = output.argmax(dim=1)
            loss = criterion(output, labels)
            correct += (pred == labels).sum().item()
            val_loss += loss.item()
            count += 1
    return correct / len(data_loader.dataset), val_loss/count


def train(model, n_epoch, optimizer, scheduler, criterion, train_loader, valid_loader):
    """Train the model on the given dataset."""
    loss_ref = float('inf')
    for epoch in range(n_epoch):
        # Set the model to training mode.
        model.train()
```

```
        for step, (images, labels) in enumerate(train_loader):
            images, labels = images.to(DEVICE), labels.to(DEVICE)
            acc, val_loss = evaluate(model, valid_loader, criterion)
        scheduler.step(val_loss)
        print(f"Epoch {epoch}, Valid Accuracy {acc   100:.2f}%")

        if val_loss < loss_ref:
            patience = 5
            loss_ref = val_loss
        else:
            if patience == 0:
                print(f"[Early Stopping] Epoch {epoch}, Valid Accuracy {acc   100:.2f}%, Valid
Loss {val_loss:.4f}")
                return
            print(f"[INFO] Patience {patience} remaining")
            patience-=1

f get_parameters(net) -> List[np.ndarray]:
    return [val.cpu().numpy() for _, val in net.state_dict().items()]



def set_parameters(net, parameters: List[np.ndarray]):
    params_dict = zip(net.state_dict().keys(), parameters)
    state_dict = OrderedDict(
            {
                k: torch.Tensor(v) if v.shape != torch.Size([]) else torch.Tensor([0])
                for k, v in params_dict
            }
        )
    net.load_state_dict(state_dict, strict=True)

def test(net, testloader):
```

```
    """Evaluate the network on the entire test set."""
    criterion = torch.nn.CrossEntropyLoss() correct,
    total, loss = 0, 0, 0.0
    net.eval()
    with torch.no_grad():
        for images, labels in testloader:
            images, labels = images.to(DEVICE), labels.to(DEVICE)
            outputs = net(images)
            loss += criterion(outputs, labels).item()
            _, predicted = torch.max(outputs.data, 1)
            total += labels.size(0)
            correct += (predicted == labels).sum().item()
    loss /= len(testloader.dataset)
    accuracy = correct / total
    return loss, accuracy
def get_evaluate_fn():

    def evaluate(
        parameters: List[np.ndarray],
    ) -> Optional[Tuple[float, Dict[str, fl.common.Scalar]]]:
        params_dict = zip(model.state_dict().keys(), parameters)
        state_dict = OrderedDict({k: torch.tensor(v) for k, v in params_dict})
        model.load_state_dict(state_dict, strict=True)
        val_loss, val_accuracy = test(model, val_loaders[0])
        test_loss, test_accuracy = test(model, test_loader)
        return val_loss, {"val_accuracy": val_accuracy, "test_accuracy": test_accuracy}

    return evaluate
# The `evaluate` function will be by Flower called after every round
def evaluate_server(
    parameters: List[np.ndarray],
) -> Optional[Tuple[float, Dict[str, fl.common.Scalar]]]:
```

```
    net = model.to(DEVICE)
    valloader = val_loaders[0]
    set_parameters(net, parameters) # Update model with the latest parameters
    loss, accuracy = test(net, valloader)
    print(f"Server-side evaluation loss {loss} / accuracy {accuracy}")
    return loss, {"accuracy": accuracy}
model_params = get_parameters(model.to(DEVICE))
strategy = fl.server.strategy.FedAdagrad(
    fraction_fit=0.3,
    fraction_eval=0.3,
    min_fit_clients=2,
    min_eval_clients=2,
    min_available_clients=NUM_CLIENTS,

initial_parameters=fl.common.weights_to_parameters(get_parameters(model.to(DEVICE))),
    eval_fn=get_evaluate_fn(), # Pass the evaluation function


)
res =
    fl.simulation.start_simulation( cli
    ent_fn=client_fn,
    num_clients=NUM_CLIENTS,
    num_rounds=10,
    strategy=strategy,
    client_resources = {'num_cpus': 4, 'num_gpus': 2}
)
import matplotlib.pyplot as plt
rounds = [x for x in range(len(res.metrics_centralized['val_accuracy']))]
val_accuracies = [x[1] for x in res.metrics_centralized['val_accuracy']]
test_accuracies = [x[1] for x in res.metrics_centralized['test_accuracy']]

# Plotting
```

```
plt.figure(figsize=(8, 6))
# plt.ylim([-1,2])
plt.plot(rounds, val_accuracies, 'b',marker="o", label='Validation Accuracy')
plt.plot(rounds, test_accuracies, 'r',marker="o", label='Test Accuracy')
# plt.plot(epochs, val_losses, 'r', label='Validation Loss')
plt.title('Validation and Test Accuracy') plt.xlabel('Rounds')
plt.ylabel('Accuracy')
plt.legend()
plt.grid(True)
plt.show()
```

APPENDIX B

SCREENSHOTS

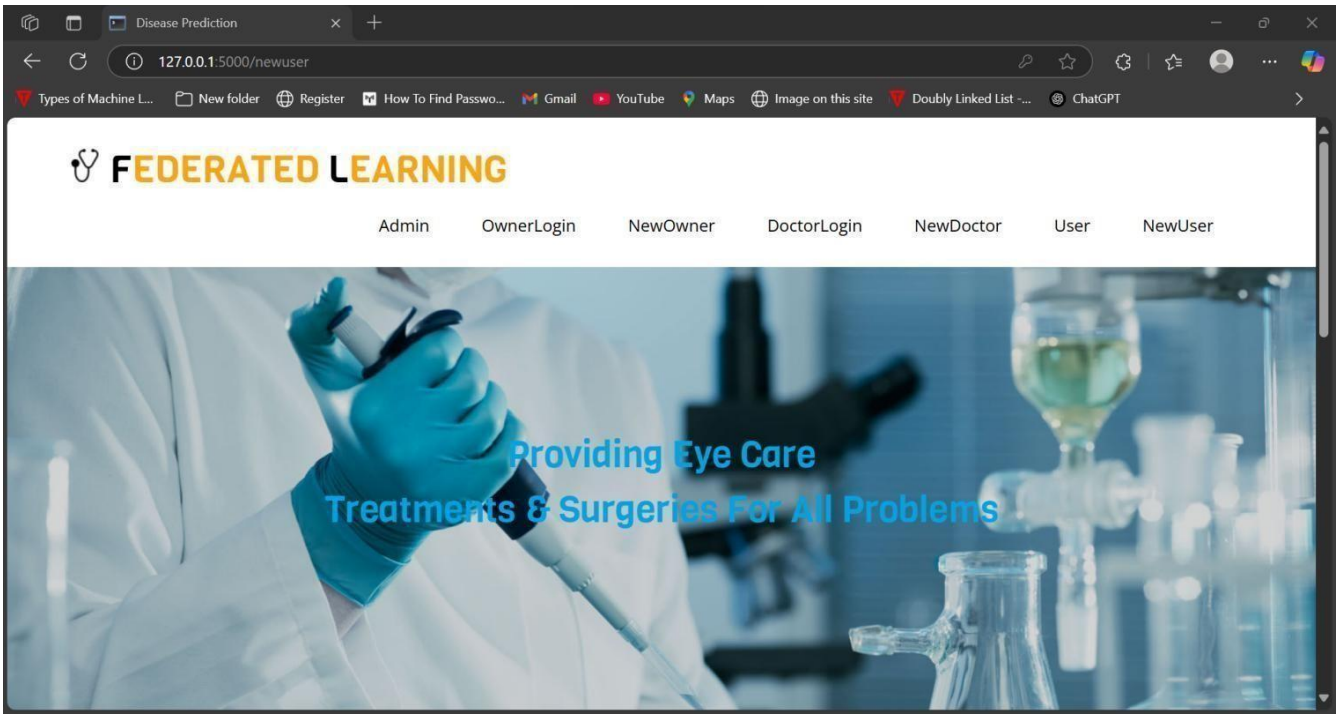

Fig.B.1 Home Page Dashboard

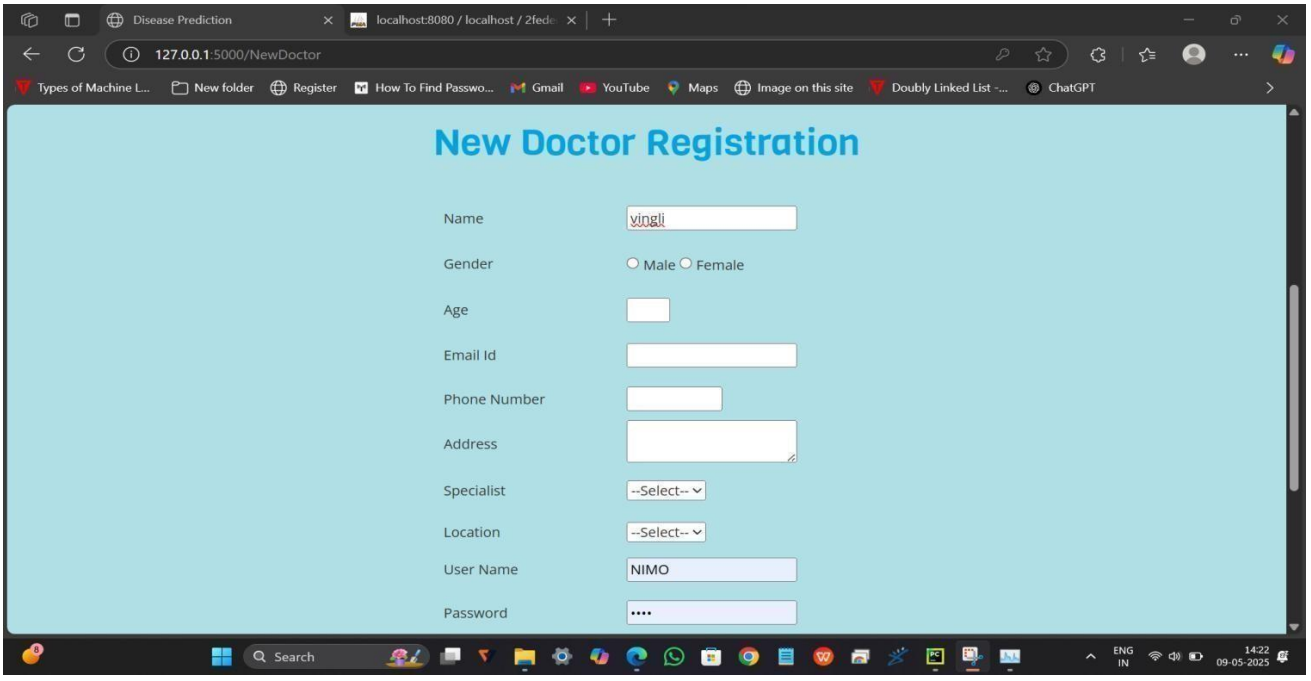

FIG.B.2 Doctor Registration Page

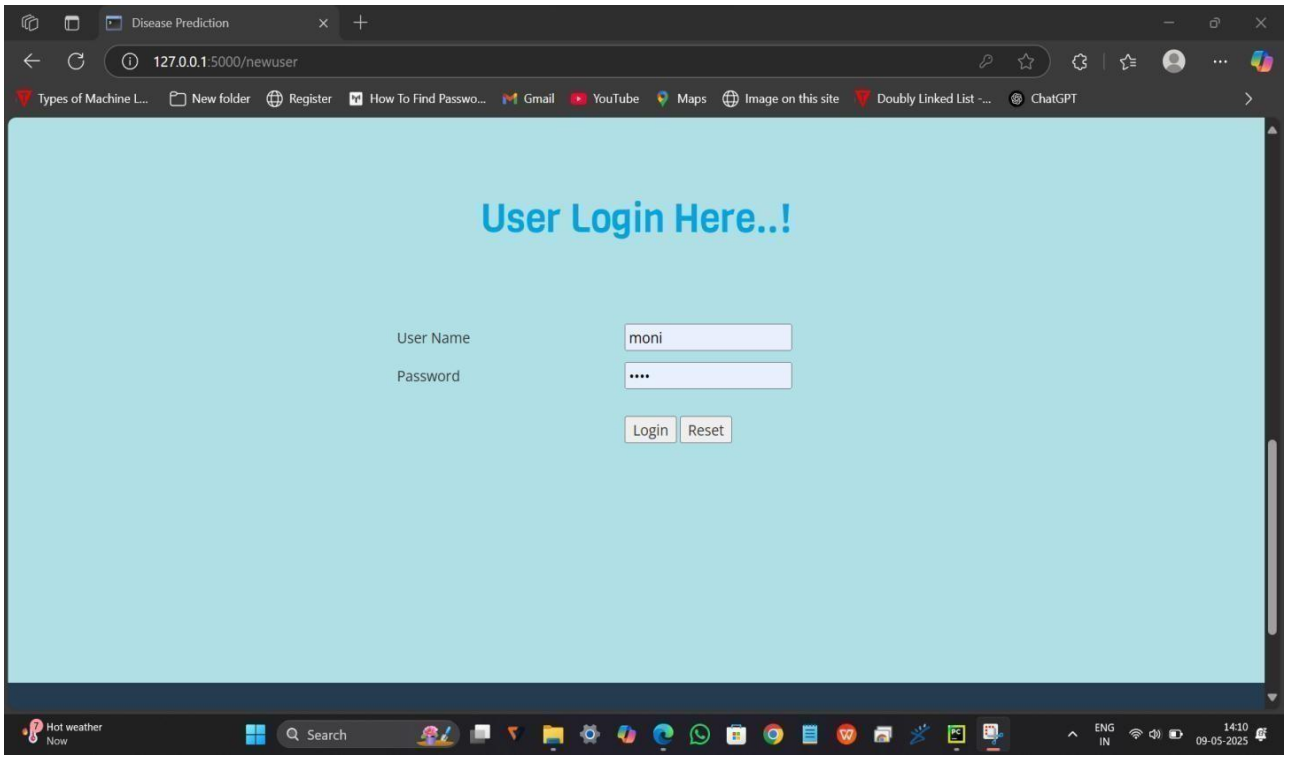

Fig.B.3 User Login Page

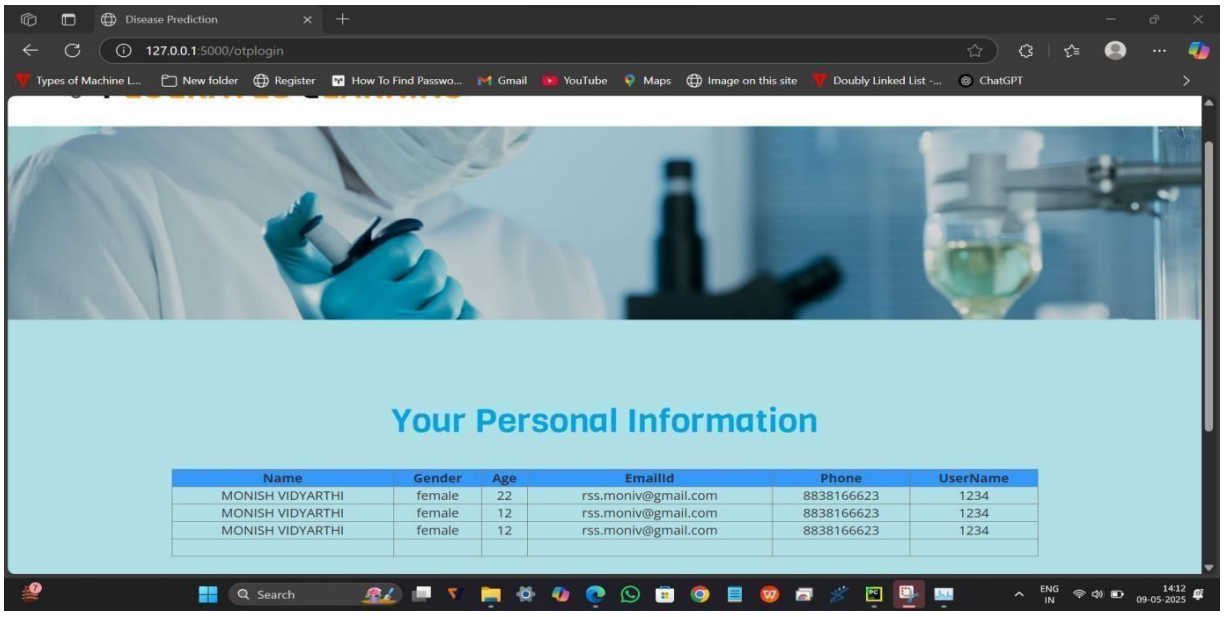

Fig.B.4 User Information

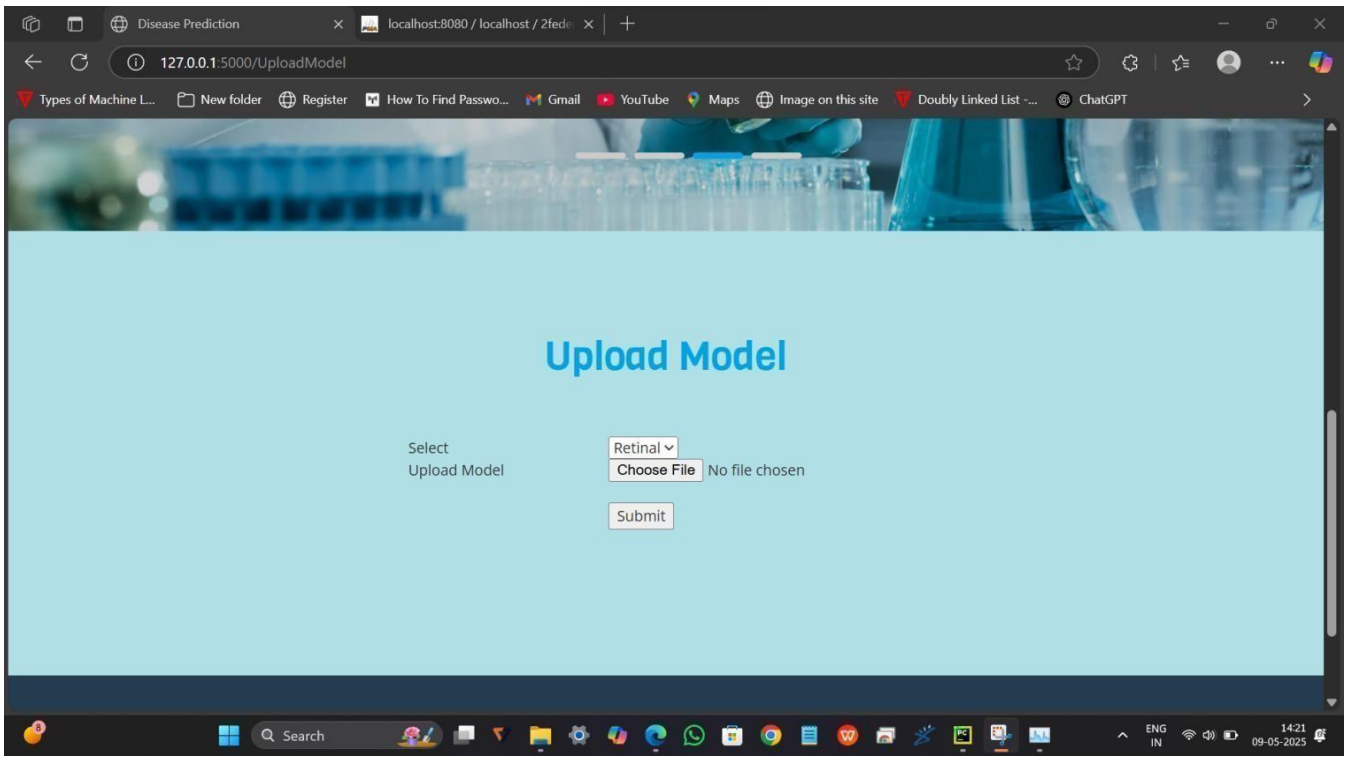

Fig.B.5 Model Information

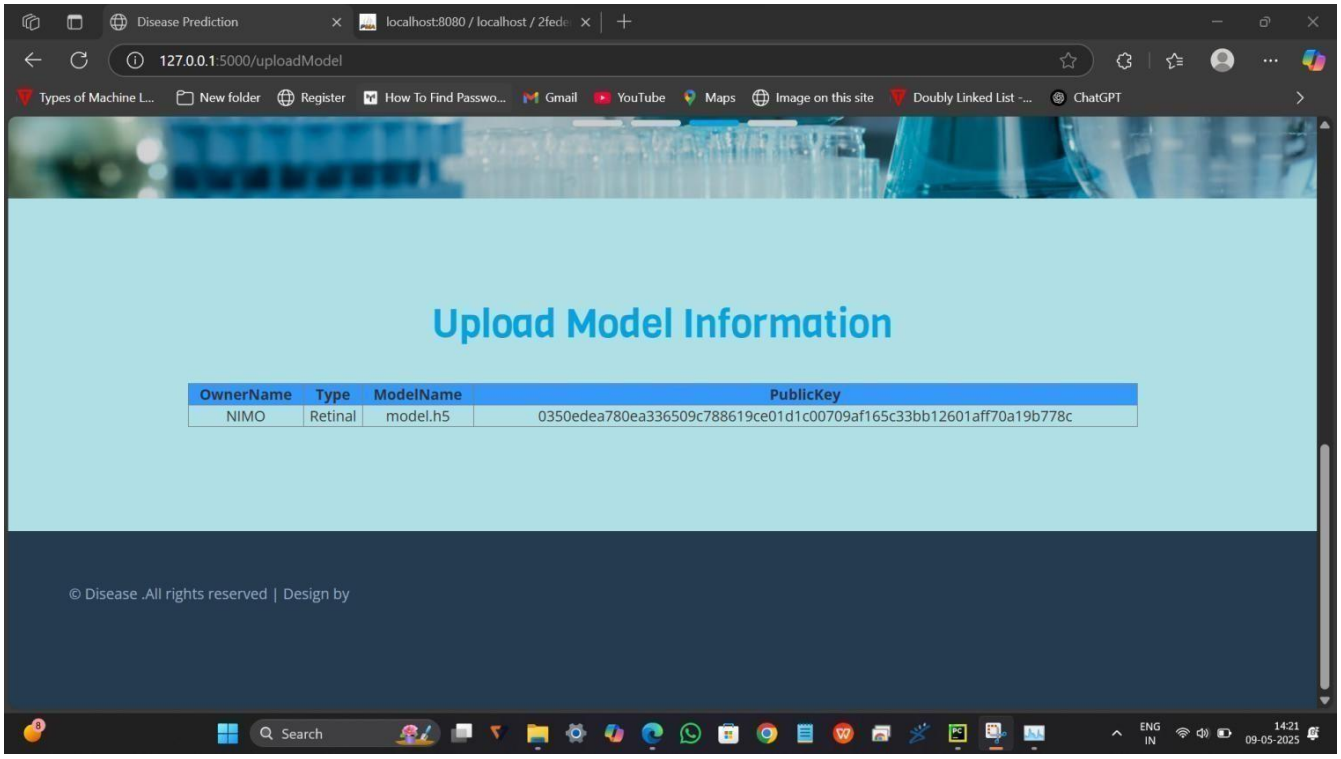

Fig.B.6 Public Key Generation

REFERENCES

1. Qayyum, Adnan, Junaid Qadir, Muhammad Bilal, and Ala Al-Fuqaha. "Secure and robust machine learning for healthcare: A survey." IEEE Reviews in Biomedical Engineering 14 : 156-180:2024.
2. Masood, Fawad, Maha Driss, Wadii Boulila, Jawad Ahmad, Sadaqat Ur Rehman, Sana Ullah Jan, Abdullah Qayyum, and William J. Buchanan. "A lightweight chaos-based medical image encryption scheme using random shuffling and XOR operations." Wireless Personal Communications 127, no. 2 : 1405-1432:2024.
3. Hasan, Mohammad Kamrul, Shayla Islam, Rossilawati Sulaiman, Sheroz Khan,Aisha-Hassan Abdalla Hashim, Shabana Habib, Mohammad Islam et al. "Lightweight encryption technique to enhance medical image security on internet of medical things applications." IEEE Access 9 : 47731-47742.2023.
4. Kamal, Sara T., Khalid M. Hosny, Taha M. Elgindy, Mohamed M. Darwish, and Mostafa M. Fouda. "A new image encryption algorithm for grey and color medical images." IEEE Access 9 : 37855-37865:2023.
5. Li, Xin, and Dongxiao Zhu. "Robust detection of adversarial attacks on medical images." In 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI), pp. 1154-1158. IEEE, 2022.
6. Peng, Zhe, Jianliang Xu, Xiaowen Chu, Shang Gao, Yuan Yao, Rong Gu, and Yuzhe Tang. "Vfchain: Enabling verifiable and auditable federated learning via blockchain systems." IEEE Transactions on Network Science and Engineering 9, no. 1 : 173-186:2021.
7. Lakshmi, C., Karuppusamy Thenmozhi, John Bosco Balaguru Rayappan, Sundararaman Rajagopalan, Rengarajan Amirtharajan, and Nithya Chidambaram. "Neural-assisted image-dependent encryption scheme for medical image cloud storage." Neural Computing and Applications 33 : 6671-668.2021.

8. Sheller, Micah J., Brandon Edwards, G. Anthony Reina, Jason Martin, Sarthak Pati, Aikaterini Kotrotsou, Mikhail Milchenko et al. "Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data." Scientific reports 10, no. 1 : 12598.2020.
9. Ibrahim, Saleh, Hesham Alhumyani, Mehedi Masud, Sultan S. Alshamrani, Omar Cheikhrouhou, Ghulam Muhammad, M. Shamim Hossain, and Alaa M. Abbas. "Framework for efficient medical image encryption using dynamic S-boxes and chaotic maps." Ieee Access :2019.
10. Liu, Quande, Hongzheng Yang, Qi Dou, and Pheng-Ann Heng. "Federated semi-supervised medical image classification via inter-client relation matching." In Medical Image Computing and Computer Assisted Intervention– MICCAI :2018.

SARANATHAN COLLEGE OF ENGINEERING

(AN AUTONOMOUS INSTITUTION)

Approved by AICTE, New Delhi, Affiliated to Anna University, Chennai.

VENKATESWARA NAGAR, PANJAPPUR, TRICHY-620 012.
Website: www.saranathan.ac.in

Accredited by NBA
NATIONAL BOARD OF ACCREDITATION
CSE ECE EEE ICE IT MECH

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
(Accredited by NBA)

Certificate

This is to Certify that MONISH VIDYARTHI R
of K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY
has presented a paper titled FEDERATED LEARNING USING EYE DISEASE PREDICTION

in the International Conference on Advances in VLSI, Communication and Navigation Systems (ICAVCNS-2025) held from 11.04.2025 to 12.04.2025 organized by the Department of Electronics and Communication Engineering, Saranathan College of Engineering, Tiruchirappalli, Tamil Nadu, India-620012.

Convener
Dr. M. Santhi

Principal
Dr. D. Valavan

SARANATHAN COLLEGE OF ENGINEERING

(AN AUTONOMOUS INSTITUTION)

Approved by AICTE, New Delhi, Affiliated to Anna University, Chennai.

VENKATESWARA NAGAR, PANJAPPUR, TRICHY-620 012.
Website: www.saranathan.ac.in

Accredited by NBA
NATIONAL BOARD OF ACCREDITATION
CSE ECE EEE ICE IT MECH

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
(Accredited by NBA)

Certificate

This is to Certify that KANNIGA SARASWATHY M
of K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY
has presented a paper titled FEDERATED LEARNING USING EYE DISEASE PREDICTION

in the International Conference on Advances in VLSI, Communication and Navigation Systems (ICAVCNS-2025) held from 11.04.2025 to 12.04.2025 organized by the Department of Electronics and Communication Engineering, Saranathan College of Engineering, Tiruchirappalli, Tamil Nadu, India-620012.

Convener
Dr. M. Santhi

Principal
Dr. D. Valavan

SARANATHAN COLLEGE OF ENGINEERING

(AN AUTONOMOUS INSTITUTION)

Approved by AICTE, New Delhi, Affiliated to Anna University, Chennai.

VENKATESWARA NAGAR, PANJAPPUR, TRICHY-620 012.
Website: www.saranathan.ac.in

Accredited by NBA
NATIONAL BOARD OF ACCREDITATION
CSE ECE EEE ICE IT MECH

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
(Accredited by NBA)

Certificate

This is to Certify that MADHUMITHA P
of K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY
has presented a paper titled FEDERATED LEARNING USING EYE DISEASE PREDICTION

in the International Conference on Advances in VLSI, Communication and Navigation Systems (ICAVCNS-2025) held from 11.04.2025 to 12.04.2025 organized by the Department of Electronics and Communication Engineering, Saranathan College of Engineering, Tiruchirappalli, Tamil Nadu, India-620012.

Convener
Dr. M. Santhi

Principal
Dr. D. Valavan

SARANATHAN COLLEGE OF ENGINEERING

(AN AUTONOMOUS INSTITUTION)

Approved by AICTE, New Delhi, Affiliated to Anna University, Chennai.

VENKATESWARA NAGAR, PANJAPPUR, TRICHY-620 012.
Website: www.saranathan.ac.in

Accredited by NBA
NATIONAL BOARD OF ACCREDITATION
CSE ECE EEE ICE IT MECH

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
(Accredited by NBA)

Certificate

This is to Certify that SUBALATHA A
of K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY
has presented a paper titled FEDERATED LEARNING USING EYE DISEASE PREDICTION

in the International Conference on Advances in VLSI, Communication and Navigation Systems (ICAVCNS-2025) held from 11.04.2025 to 12.04.2025 organized by the Department of Electronics and Communication Engineering, Saranathan College of Engineering, Tiruchirappalli, Tamil Nadu, India-620012.

Convener
Dr. M. Santhi

Principal
Dr. D. Valavan