Privacy Preserving Using Federated Learning for Collaborative Healthcare Data Analysis

Abstract

Maintaining privacy is essential in the healthcare industry while working across institutions on data analysis. In order to train machine learning models using decentralized data from various healthcare providers without jeopardizing patient privacy, this work proposes an implementation of Federated Learning (FL) combined with Homomorphic Encryption (HE). This method is perfect for settings where privacy is a concern because it uses HE to guarantee the safe aggregation of encrypted model updates from every client (hospital). By providing precise predictive models for patient outcomes without disclosing sensitive data, our work expands the application of privacy-preserving FL frameworks in the healthcare industry.

When there are stringent privacy laws in effect, like GDPR and HIPAA, this dual strategy works very well. Without disclosing raw data, HE ensures the secure aggregation of encrypted model updates from each customer (hospital), facilitating effective and safe collaboration. Healthcare organizations may create accurate prediction models for patient outcomes, such readmission rates or treatment efficacy, while maintaining data confidentially thanks to the integration of FL and HE. This work greatly expands the possibilities of decentralized healthcare analytics by developing privacy-preserving frameworks, opening the door for safer, cooperative advancements in the sector.

Keywords

Federated Learning, Homomorphic Encryption, Privacy Preservation, Collaborative Healthcare, Machine Learning

Introduction

In the data-driven world of today, the healthcare industry is at the forefront of innovation, producing enormous volumes of useful patient data. Data from electronic medical records (EMRs) and sophisticated diagnostic systems is essential for enhancing patient outcomes and advancing innovative research. Concerns regarding security and privacy, however, are growing along with the dependence on data. While institutions must adhere to stringent standards like the Health Insurance Portability and Accountability Act (HIPAA) in the US and the General Data Protection Regulation (GDPR) in Europe, patients naturally worry about how their sensitive information is handled. Investigating creative privacy-preserving techniques for healthcare data is crucial because these restrictions frequently restrict data sharing.

Strict security measures are necessary due to the particular nature of healthcare data, which is closely linked to human life and health. Data sharing between institutions is hampered by privacy and security issues, which leads to fragmented datasets and less-than-ideal model performance. Global optimization is frequently not achieved by models that are trained

separately by each institution. Therefore, a new approach is required to strike a compromise between the potential of collaborative machine learning and privacy requirements.

Federated Learning: A Decentralized Solution

A decentralized learning technique called federated learning (FL) allows several data holders to work together to train models without exchanging original data. Rather, throughout intermediary stages, institutions just share training characteristics. By keeping data under the custodians' control, this method enables hospitals and healthcare providers to improve machine learning models without jeopardizing the privacy of patient data.

By doing away with the requirement for a central repository of sensitive data, FL mitigates the dangers of data breaches and illegal access, in contrast to conventional centralized data-gathering strategies. Individual institutions carry out local training, while a central server receives only aggregated updates. FL is especially well-suited to sectors like healthcare, where privacy is crucial, because of its decentralized architecture, which guarantees that raw data never leaves its source.

Homomorphic Encryption for Enhanced Security

This study combines FL with Homomorphic Encryption (HE) to further enhance privacy. Sensitive information can be kept private by using HE to do calculations on encrypted data without first decrypting it. By using FL, hospitals may train models locally and send encrypted updates to a central server, which aggregates them without having access to the raw data that underlies them. This approach is particularly helpful for predictive healthcare models where strong privacy guarantees are essential, including patient readmission projections.

Advancing Predictive Analytics in Healthcare

The issues of privacy and teamwork in healthcare can be effectively and safely resolved by combining FL and HE. The need for privacy-preserving analytics is growing as the Internet of Things (IoT) and electronic health records (EHRs) fuel the exponential expansion of healthcare data. While HE makes sure that calculations are kept private at all times, FL allows medical staff to work together efficiently without jeopardizing private patient information. This strategy strikes a balance between ethical considerations and potent machine learning techniques, laying the foundation for future data-driven healthcare advancements. It makes it possible to create trustworthy and internationally optimized prediction models, opening the door to better patient outcomes and ground-breaking medical research.

This study shows a means to get beyond obstacles to data sharing and realize the full potential of predictive analytics in healthcare by resolving privacy issues through the integration of FL and HE.

Fig 1. Privacy Preserving Using FL and HE on collaborative healthcare data

Background Study

The healthcare industry has always been at the forefront of technological innovation, generating massive amounts of data from electronic health records (EHRs), diagnostic imaging, and IoT-enabled devices. This data has the potential to transform patient care, making it more personalized and effective. However, with great power comes great responsibility—sharing and analyzing healthcare data raises serious concerns about privacy, trust, and compliance with laws like GDPR and HIPAA.

Traditionally, machine learning models have relied on centralized data collection. While this approach ensures high-quality training, it also creates a single point of vulnerability. A data breach in such a setup could expose sensitive patient information on an unprecedented scale. Understandably, healthcare organizations are hesitant to share their data openly, even for the greater good.

To address this issue, we have explored **Federated Learning (FL)** as a decentralized approach that enables collaborative model training without exposing local data. Our study references research, including "Privacy Preservation for Federated Learning in Health Care(link)," which highlights FL's capability to uphold data privacy while facilitating collaborative learning. Additionally, "Federated Learning with Homomorphic Encryption for Ensuring Privacy in Medical Data (link)" provides insights into combining FL with Homomorphic Encryption (HE) to enhance privacy further.

A Quick Look at the Research That Paved the Way

1. **Federated Learning Basics:**
   Introduced as a solution to data privacy challenges, FL allows multiple entities to collaboratively train a global model without exposing their local data. This concept was popularized in applications like mobile devices, where FL enables devices to learn from user data locally and improve a shared model (e.g., Google's Gboard for text predictions). However, extending FL to healthcare presents unique challenges, including non-uniform data distributions and stricter privacy requirements.
2. **Homomorphic Encryption (HE):**
   To address privacy vulnerabilities in FL, researchers have turned to **homomorphic encryption**. This technique allows computations to be performed on encrypted data without decrypting it. While HE provides strong privacy guarantees, its computational overhead has often been criticized as impractical for real-world applications.

The Healthcare Perspective

In healthcare, data is not only sensitive but also heterogeneous. One hospital's data might include imaging scans, while another's focuses on laboratory results or patient histories. This diversity, while rich, makes it challenging to create a one-size-fits-all model. Federated Learning offers a pathway to tackle this heterogeneity by enabling each institution to train locally on its unique dataset while contributing to a robust global model.

Additionally, healthcare data is often non-IID (non-independent and identically distributed), meaning that data distributions vary significantly across institutions. For instance, a rural hospital may encounter diseases that differ from those commonly seen in urban centers. This non-uniformity introduces challenges in ensuring that the global model performs well across all participating institutions.

Literature Review

Privacy-preserving technologies, particularly Federated Learning (FL) and Homomorphic Encryption (HE), are paving the way for secure and collaborative data analysis in privacy-sensitive domains like healthcare. FL allows institutions to train machine learning models collaboratively without exchanging raw data, ensuring data privacy at its source. HE, on the other hand, facilitates computations directly on encrypted data, providing a robust layer of security by eliminating exposure to raw information.

Recent studies have focused on the integration of these methods. For instance, in order to address the shortcomings of each technique in privacy-preserving collaborative machine learning, **Grivet Sébert (2023)** investigates merging Differential Privacy with Homomorphic Encryption. In addition to offering insights into their possible uses in privacy-critical industries like finance and telecommunications, this work demonstrates how the combination of various technologies might improve privacy without significantly compromising utility.

Existing research highlights the individual strengths and challenges of these techniques. FL's reliance on model updates for collaborative learning can inadvertently expose sensitive

information patterns. Differential Privacy (DP) has been introduced as a mitigation strategy, adding calibrated noise to obfuscate sensitive details. However, this approach often compromises model utility, particularly in scenarios involving smaller datasets or high precision requirements. Meanwhile, HE offers a stronger privacy guarantee by keeping data encrypted throughout computation but comes with significant computational overhead.

Recent advancements have explored the integration of FL and HE in domains like finance and telecommunications. However, the healthcare sector's unique challenges, such as the sensitivity of patient data and the need for real-time analytics, remain underexplored. The necessity to balance privacy, computational efficiency, and model accuracy underscores the importance of developing tailored solutions for this field.

Problem Definition

Healthcare is a domain where data is not just abundant but also immensely valuable. From diagnosing diseases to predicting treatment outcomes, the ability to analyze large, diverse datasets can transform patient care. However, this potential is hindered by significant challenges in **data sharing and collaboration**, particularly in healthcare, where privacy and security are paramount. Defining the core problems within this context is essential to developing solutions that address the underlying challenges.

1. Barriers to Collaborative Data Sharing

Healthcare data is often siloed across multiple organizations, including hospitals, clinics, and research institutions. These silos are a result of strict regulations (e.g., GDPR, HIPAA) and privacy concerns surrounding sensitive patient information.This lack of secure, scalable mechanisms for sharing and analyzing data stifles innovation and leads to **isolated and less effective machine learning models.**

2. Privacy Threats in Distributed Learning

Federated Learning (FL) offers a potential solution by allowing models to be trained collaboratively without exposing raw data. However, the process itself is not immune to vulnerabilities:

- **Parameter Leakage:** Attackers can exploit shared gradients or model updates to infer sensitive information about individual participants. This phenomenon, known as **gradient inversion**.
- **Malicious Participants:** The inclusion of untrusted entities in the FL process can result in **data poisoning** or **backdoor attacks,** compromising the integrity and utility of the global model.
- **Model Memorization:** Overfitting during training can lead models to inadvertently memorize sensitive details about the training data, which attackers can extract via query-based attacks.

3. Lack of Robust Privacy-Preserving Mechanisms

While traditional privacy-preserving techniques, such as anonymization and differential privacy, can mitigate some risks, they come with trade-offs:

- **Anonymization** often fails to protect against re-identification attacks when combined with auxiliary data and **Differential Privacy** introduces noise to model updates, reducing the utility and accuracy of the resulting models.
- **Homomorphic Encryption (HE)** provides stronger guarantees but is computationally expensive, especially in resource-constrained settings like healthcare.

4. Real-World Scalability and Reliability

Deploying federated learning in a real-world healthcare setting introduces additional complexities:

- **Heterogeneous Data Distributions:** Healthcare data is inherently across institutions due to differences in patient demographics, diagnostic tools, and clinical practices. This can lead to biased models that fail to generalize effectively.
- **Dynamic Participation:** Participants in FL, such as hospitals or devices, may drop out or rejoin during training, creating challenges in maintaining consistency and efficiency.
- **Trust and Accountability:** Verifying the authenticity and reliability of participants is essential to prevent malicious or erroneous contributions.

Scope of Current Work

This study aims to bridge the gap by proposing a unified framework that combines FL and HE for privacy-preserving healthcare data analysis. Building upon established methodologies and addressing their limitations, the scope of this work includes:

Enhanced Privacy Protections: By leveraging HE, the framework ensures that patient data remains encrypted during computations, significantly reducing the risk of data breaches or inadvertent exposure.

Optimized Computational Performance: To address the computational overhead associated with HE, this study introduces strategies for efficiency improvement, making the approach viable for real-world healthcare applications.

Scalable Collaboration: The framework is designed to handle the diverse and complex datasets typical of healthcare institutions while maintaining scalability across multiple collaborators.

Application-Specific Implementation: A focus on predictive healthcare models, such as patient readmission predictions, demonstrates the framework’s practical utility and adherence to regulatory requirements like GDPR and HIPAA.

Proposed Model

Block Diagram

The Proposed system's block diagram combines Homomorphic Encryption (HE) with Federated Learning (FL) to produce a framework for collaborative, privacy-preserving healthcare data analysis. This method allows several institutions to safely contribute to a global machine learning model while guaranteeing the confidentiality of individual data. An extensive explanation of the main elements and their functions is provided below:

Hospitals as clients:

- Function - These stand in for medical facilities that have local datasets with private patient data. Every customer is accountable for utilizing its local dataset to train the global machine learning model and ensuring that no data ever leaves its location without protection.

Process:

- The central server sends the global model to the hospital.
- To adjust the model to the hospital's data, local calculations are made.
- The public key is used to create encrypted updates, which are then transmitted back to the global server.

Importance - Because only encrypted model changes are communicated rather than raw data, this decentralization reduces privacy threats.

Global Server:

Functions -

- Serves as the foundation for training by initializing the global model with random parameters.
- All clients send encrypted model updates to it.
- Enhances the global model by homomorphically aggregating the updates.
- Returns the revised global model to the clients for the upcoming training session.

Security Measures - The server aggregates encrypted updates directly without decrypting or accessing raw data.

Significance - Without jeopardizing the privacy of individual datasets, the server makes collaboration easier.

Encryption Module:

Role - Offers a protective layer that encrypts local changes prior to sending them to the global server and it makes use of homomorphic encryption (HE) to guarantee that encrypted data can be used directly for calculations.

Workflow -

- To secure its model changes, each client employs a public encryption key that is supplied by the global server.
- To ensure total secrecy, the server aggregates these encrypted updates without decrypting them.
- Only after aggregation can the private key, which is safely stored on the server, be used for decryption.

Key Advantage - Always protects client privacy by preventing the disclosure of sensitive data patterns, even during the aggregation stage.

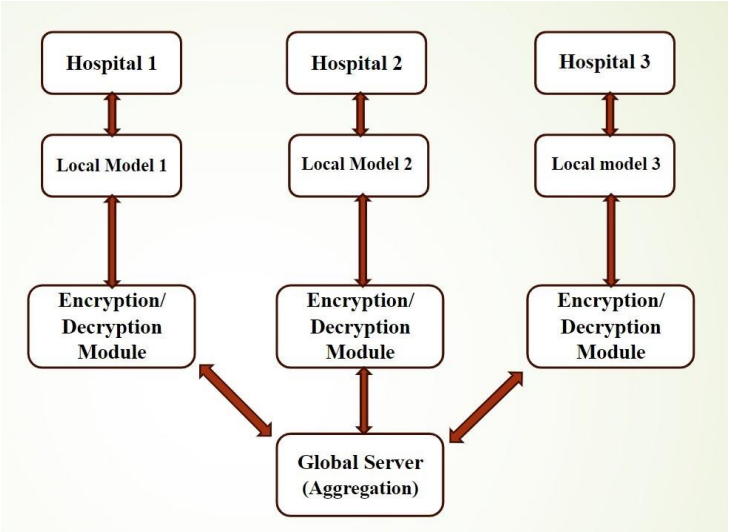

Fig 2. Block Diagram

The block diagram visually demonstrates how data remains encrypted throughout the process, Clients train the model locally and only share encrypted updates and the global server aggregate these updates homomorphically and redistributes the refined global model.

Hardware Requirements

Client Devices -

Each healthcare facility should have high-performance workstations or servers to carry out local model training effectively. Computationally demanding operations like encrypting updates and training machine learning models must be handled by each device.

Central Server -

A powerful server with a large amount of processing power to manage client interactions, execute homomorphic aggregations, and handle encrypted updates. Reliability and high availability must be supported by the server.

Networking Infrastructure -

Encrypted data transmission between clients and the central server requires secure, fast internet access. To stop unwanted access, network security tools like intrusion detection systems and firewalls should be installed.

Software Requirements

Programming Language – **Python -** Preferred because of its extensive library ecosystem for federated learning, data encryption, and machine learning. It is perfect for quick development and deployment because of its ease of use and adaptability.

Libraries and Frameworks -

TensorFlow Federated - Enables decentralized training to be seamlessly integrated into federated learning workflows. TensorFlow Federated is a powerful framework specifically designed for implementing decentralized machine learning. It provides seamless integration of federated learning workflows, enabling multiple participants (such as hospitals, organizations, or devices) to collaboratively train a shared model without centralizing sensitive data.

PySyft - A library for machine learning tasks that protects privacy and security. PySyft is an open-source library designed to enhance privacy and security in machine learning tasks. It facilitates secure multi-party computation (SMPC) and federated learning, ensuring that sensitive data remains protected throughout the training process.

PyCryptodome - Offers the resources required to put homomorphic encryption into practice. It is a comprehensive library for implementing cryptographic techniques, particularly those required for advanced encryption protocols like homomorphic encryption. Homomorphic encryption allows computations to be performed directly on encrypted data without needing to decrypt it, ensuring data remains confidential throughout the process.

Operating System -

Windows or Linux, tailored to high-processing-load server operations.

Windows: Offers an intuitive interface and extensive support for enterprise-grade software, making it an excellent choice for organizations with existing Windows-based infrastructure.

Linux: Renowned for its stability, scalability, and open-source nature, Linux is particularly well-suited for high-performance computing (HPC) environments.

Databases -

Secure cloud storage options (like AWS S3 or Google Drive) for keeping final models and encrypted updates. **Google Drive** is an accessible and secure cloud storage solution, Google Drive is ideal for smaller-scale federated learning experiments. It supports seamless collaboration and file sharing while maintaining robust encryption.

System Architecture

The system architecture of the proposed model centers on the idea of federated learning, in which hospitals and other healthcare facilities function as separate clients inside the system. Sensitive patient data that is locally stored is used by each institution to train its own machine learning model. Homomorphic encryption, which enables computations on encrypted data without disclosing the actual content, is used to protect the privacy of the model updates from each institution. The architecture's principal elements are as follows:

Clients (Hospitals): Every client is a separate healthcare organization with its own dataset. These clients are in charge of performing local training on their datasets while maintaining the privacy and security of patient data.

Global Server: The federated learning process is coordinated by the global server. Initializing the global model, allocating encryption keys, obtaining encrypted model updates from clients, and combining these changes to create an improved global model are its main responsibilities.

Homomorphic Encryption: Since calculations are carried out directly on encrypted data, HE is essential to maintaining privacy since it enables the server to aggregate model updates without having access to patient data or unencrypted updates.

The encrypted updates—not the raw data—are transmitted to a central server after a hospital's model has been trained. To create a global model, this server compiles the encrypted updates from every hospital that takes part. This approach's main benefit is that it enables hospitals to work together and enhance the global model without ever exchanging raw patient data or jeopardizing patient privacy.

To protect patient privacy, the data is never decrypted by the central server. While the data is being aggregated to improve the model, homomorphic encryption safeguards the information. Healthcare organizations can use AI and machine learning with this solution while protecting patient privacy. While maintaining stringent data privacy rules, the collaborative architecture enables institutions to improve healthcare outcomes including disease diagnosis and patient prediction over time.

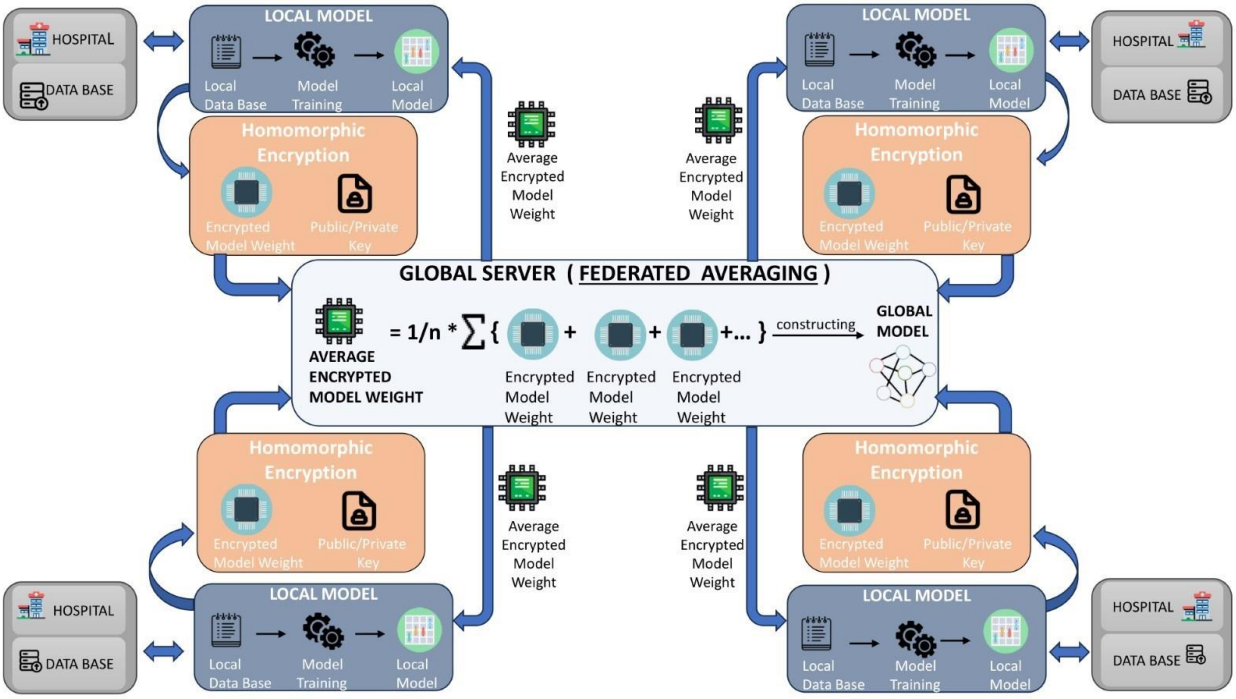

Fig3.System Architecture

Working Procedure

The proposed model's operational process is composed of methodical processes that guarantee data analysis that protects privacy:

Model Initialization -

The collaborative training procedure begins with the global server initializing a simple machine learning model with random weights. Next, homomorphic encryption keys for the public and private domains are produced. All clients have access to the public key, which enables them to encrypt their local model updates. Data secrecy is maintained throughout the training process since the private key, which is kept safe by the server, only permits decryption of these updates at the server level. This method allows for cooperative model improvement while protecting privacy.

Local Model Training -

By using its local data updates to train the global model, each client enables the model to recognize distinct patterns in the dataset. The model adjusts its parameters based on local features throughout this training phase. The client creates updates that take this modification into account after training is finished. Before being sent to the central server, these updates are encrypted using the public homomorphic encryption key to protect privacy. Unauthorized parties

cannot access or alter the raw updates thanks to this encryption, which ensures their confidentiality.

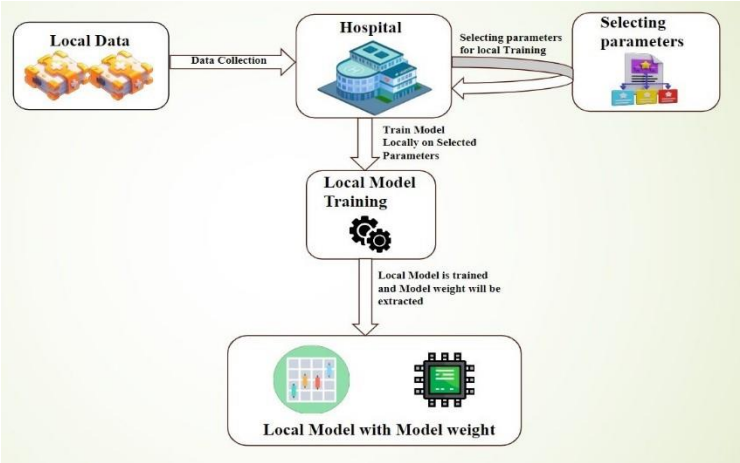

Fig 4. Local Model Training

Homomorphic Encryption -

By encrypting the model weights throughout the federated learning process, homomorphic encryption protects privacy. By enabling computations on encrypted data, this encryption protects patient privacy. Clients (such as hospitals) only exchange encrypted model changes, which are condensed at the global server (Google Drive) to protect sensitive data. Model updates are kept safe during the cooperative training process thanks to the project's use of PySEAL to implement homomorphic encryption.

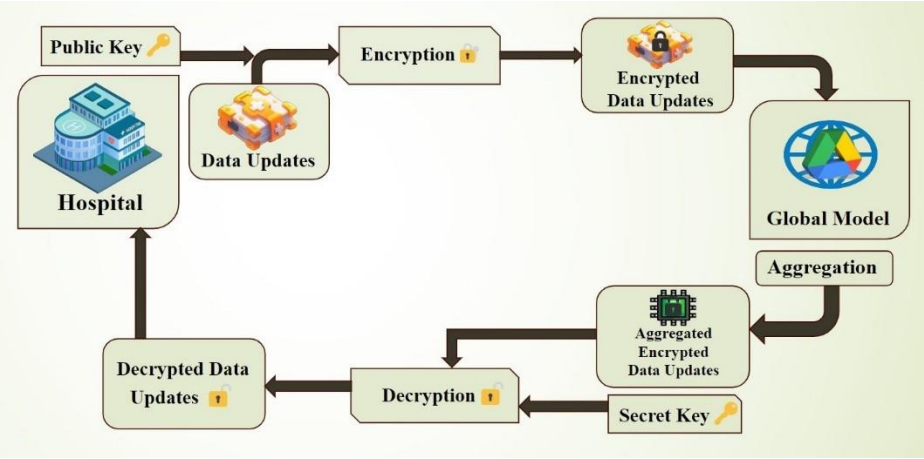

Fig 5. Homomorphic Encryption

Server-Side Aggregation -

All clients provide encrypted updates to the global server. Without decryption, homomorphic addition is used to aggregate these updates. The server refines the global model for the following iteration by utilizing the private key to decrypt the combined model after aggregation is finished.

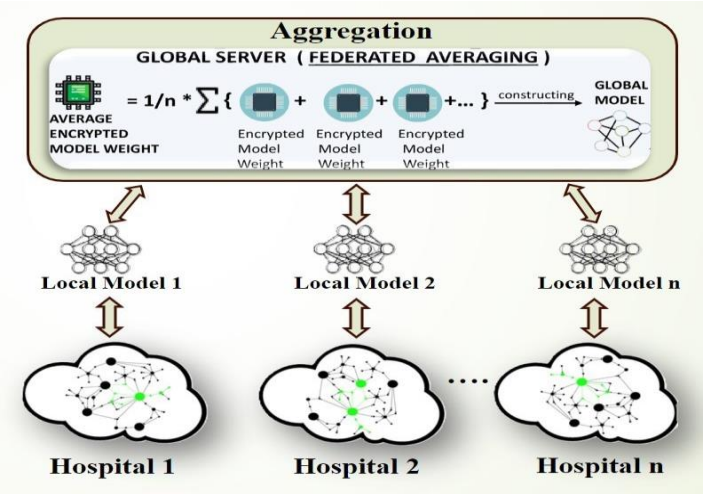

Fig 6. Server – Side Aggregation

Iterative Process -

Clients receive a redistribution of the updated global model. Until the model converges, or for a predetermined number of rounds (that is 10), this process is repeated.

Deployment of Final Model -

Once the training rounds are completed, all clients receive the final global model. This refined model can be used by any client for making local predictions based on their own data, without revealing sensitive information. The process ensures that clients can utilize the collaborative benefits of the model while maintaining privacy, as no raw data is shared. Thus, the global model can be applied across different institutions while safeguarding the confidentiality.

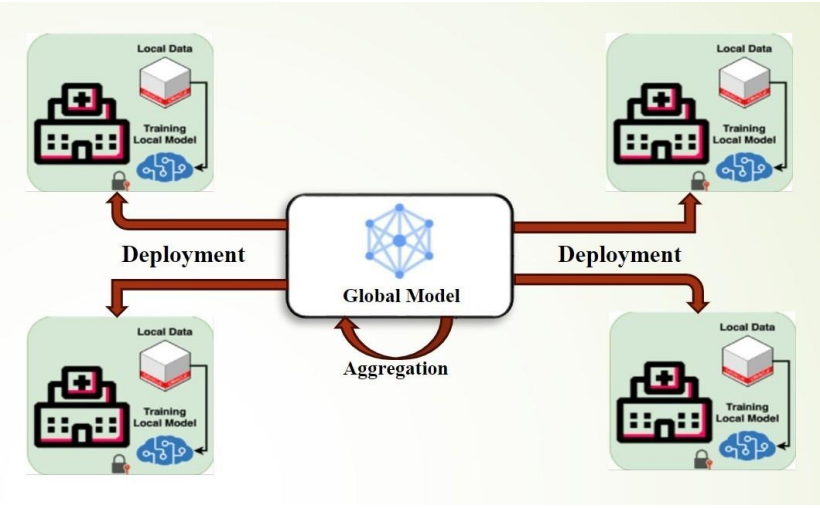

Fig 7. Deployment of Final model

List of Parts/Items

Hardware Components -

High-performance servers -

High-performance servers with **GPUs or TPUs** are essential for healthcare institutions to speed up the training of machine learning models. In order to train precise models in federated learning, these servers are made to handle huge and intricate healthcare datasets.

A central aggregation server -

Receiving encrypted model updates from multiple clients is a critical function of the **central aggregation server**. The global model is improved by combining these changes. This server has a large amount of storage space to safely store encrypted data, as well as sophisticated processing power to manage demanding computational jobs.

Networking devices -

By connecting the clients and central server, **networking devices** provide dependable and secure data transfer. By enabling the seamless transport of encrypted data, these devices protect sensitive data while enabling clients to interact with the server effectively. To ensure that no raw patient data is revealed during the network transmission process, security techniques such as **Homomorphic encryption** are used to safeguard the data's confidentiality and integrity.

Software Components -

PySyft and TensorFlow Federated -

A robust library for safe, private machine learning is PySyft. By allowing data to stay decentralized while facilitating collaborative model training, it makes federated learning easier to implement.

TensorFlow By supporting federated learning protocols and facilitating the smooth aggregation of updates from numerous clients while maintaining the privacy and security of the data, Federated expands TensorFlow for distributed learning.

PyCryptodome -

A Python package called PyCryptodome offers cryptographic features. It makes data encryption and decryption possible, guaranteeing that private information is kept private during federated learning.

Cloud Storage Solutions -

Model parameters and encrypted data are stored on secure cloud storage providers. By limiting access to the data to authorized users, these services offer scalable storage while preserving data security. Encrypting data while it is at rest guarantees that it cannot be read even if it is hacked.

Module Diagrams

Encryption Module

Input - After training their models on private, local datasets, clients produce local model updates. These updates include sensitive data linked to the client's data, but they also show the model's learning.

Process - The global server generates a public key that is used to encrypt the updates. To guarantee that calculations may be carried out on the encrypted updates without disclosing the sensitive data underneath, homomorphic encryption is utilized. This procedure let the server to carry out essential tasks while concealing patterns, connections, or particular data points from unauthorized parties.

Output - This module's ultimate output is a collection of encrypted model updates that can be safely sent to the central server without jeopardizing the data's confidentiality.

Aggregation Module

Input - The global server receives the encrypted updates from several clients. Despite containing model modifications based on client input, these updates have been encrypted using the public key and do not disclose any sensitive information.

Process - The encrypted model updates are subjected to homomorphic addition by the server. The server can combine and improve the model based on encrypted data thanks to homomorphic encryption, which enables it to aggregate updates across clients without having to decode them. This guarantees that customer data privacy is upheld at every stage of the procedure.

Model updates can be aggregated without decrypting the original data thanks to homomorphic addition, which enables mathematical operations to be carried out directly on encrypted data.

Output - A refined global model that incorporates the collective knowledge of all clients is produced by the server. The private key, which is safely kept on the server, is used by the global server to decode the result and complete the aggregate. The final model parameters must be revealed in this step. Crucially, sensitive information is kept private because the server never sees the raw data from any one client throughout the aggregate process.

Network Breaking Module

The Network Breaking Module is essential for maintaining the federated learning process' dependability and continuity even in the event that client-central server connectivity is interrupted. By putting in place systems for client-side recovery, server-side administration, and error detection and handling, this module is made to deal with network connectivity outages.

Client-Side Recovery

Instead of throwing away their encrypted model updates in the event of a network outage, clients save them locally. This guarantees that no important data is lost in the disturbance. Following the restoration of network connectivity, the clients can send the cached updates to the global server for aggregation. The system reduces data loss by locally storing these updates, which guarantees that model training proceeds without interruption after the link is restored.

Server-Side Management

During the aggregation process, the server-side system keeps track of received changes and handles partial contributions. The server keeps processing the available updates, enabling the global model to gradually change even if some clients have been unable to send their updates because of network problems. By preventing occasional network outages from completely stopping model refinement, this feature reduces training downtime and enables the system to continue using the data it has previously received.

Error Detection and Handling

The module's error detection and handling feature keeps a close eye on the data being sent back and forth between the clients and the server. The system looks for inconsistencies that could lower the final model's quality, like corrupted or missing updates. In the case that problems are found, the faulty updates are reported and separated, avoiding their inclusion in the aggregation process. By ensuring that only legitimate and comprehensive updates contribute to the model's improvement, this phase is essential for preserving the integrity of the global model.

Thus, even in the event of unforeseen network outages, the Network Breaking Module acts as a safety measure to ensure continuous and error-free training. By handling partial updates on the server, storing updates locally during outages, and protecting the model's integrity by identifying inaccurate data, it makes it possible for the federated learning process to proceed without interruption.

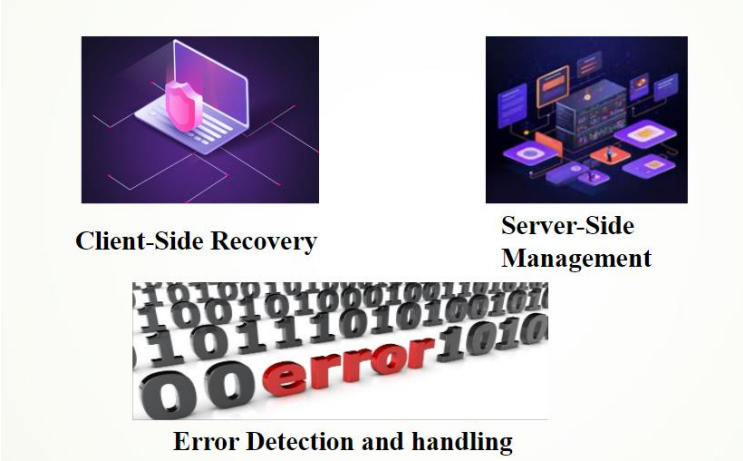

Fig 8. Network Breaking Module

Implementation

The implementation of the proposed framework for privacy-preserving federated learning (FL) using homomorphic encryption (HE) in collaborative healthcare data analysis is designed to ensure data privacy, efficient computation, and scalability. Below, the full implementation is outlined, divided into key modules that signify its major components -

Global Model Setup: Random weights are used to initialize the global model □0. This acts as the starting point for upcoming training cycles.

Key Generation: Two encryption keys are created: a private key □□ for decrypting aggregated updates and a public key □□ for encrypting model updates. A homomorphic encryption system is used in this process to give the security guarantees required for medical data.

Public Key Distribution: All clients are safely given the public key □□, which guarantees that they can encrypt their changes before transmitting them to the server.

Federated Learning Process: The model updates are iteratively improved through client-server interactions during the ten (□ =10) communication rounds that make up the federated learning process.

For every round □:

Model Distribution: Every client receives the most recent global model □□ from the global server. This model is the foundation for local training and is essential for coordinating client models toward a shared goal.

Client-Side Training: Every client obtains its local dataset, □□, and uses the received model, □□, to conduct local training. This is calculating the model updates Δ□□□ by implementing federated learning algorithm to the local data. This training guarantees that the model adjusts to the distinct features of the data from every client. Following training, clients use the public key

□□ to encrypt their model updates, producing encrypted updates Encrypted (Δ□□□). Since the clients never exchange raw updates, homomorphic encryption enables them to protect their data. The global server receives these encrypted updates for aggregation.

Server-Side Aggregation:

a) The aggregation procedure starts when the global server receives the encrypted updates from every client:

i) It uses the current encrypted global model to initialize the aggregation:

Encrypted(□□+1) = Encrypted(□□)

ii)Without ever decrypting the updates, the server can calculate the aggregated model by performing homomorphic addition on the encrypted updates. This step is essential because it keeps sensitive data safe:

Encrypted(□□+1) = Encrypted(□□+1) + Encrypted(Δ□□□)

b) The updated global model □□+1 is obtained by the server decrypting the result once all modifications have been gathered. Using the private key □□, this decryption is necessary to get a workable model for further iterations.

Final Model Distribution: Each client receives the final global model □□ once all communication rounds are finished. This makes it possible for every client to gain from the group training effort, enhancing their local model without ever disclosing private information.

Client-Side Verification: This process ensures no discrepancies indicating a privacy violation. If the updates match, the client confirms privacy protection; otherwise, the client halts the process and flags potential issues. This step is crucial for maintaining trust in FL environments.

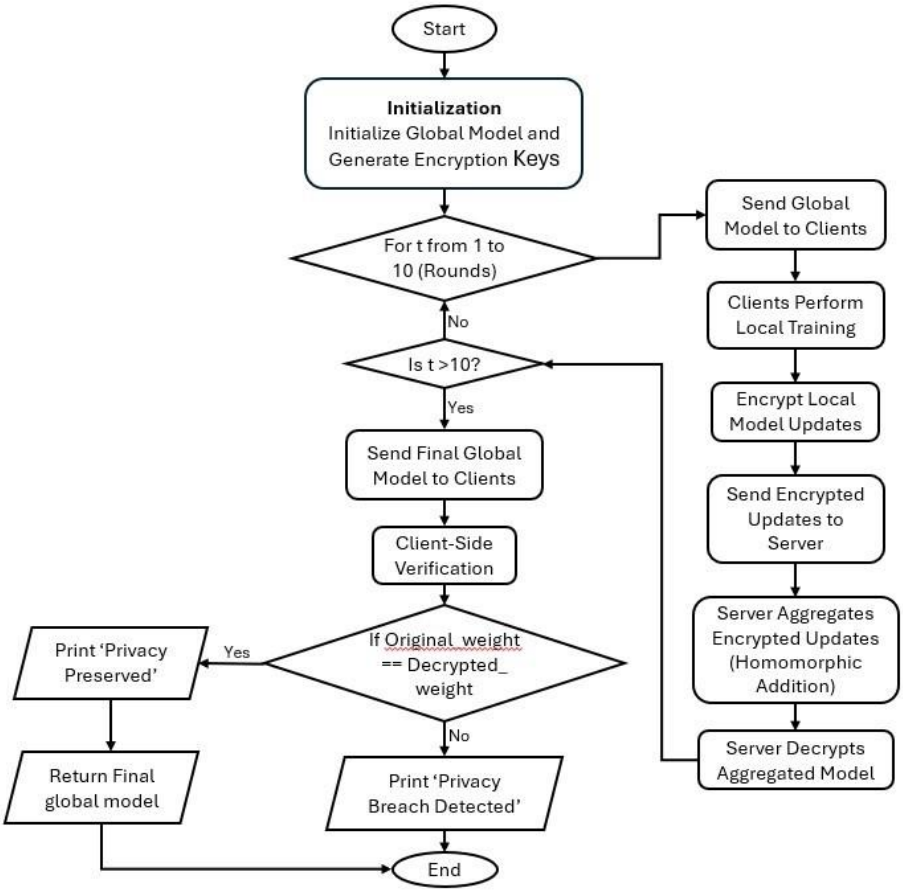

Fig 9. Schematic Illustration of the Proposed Model

ALGORITHM: Privacy Preserving Using Federated Learning for Collaborative Healthcare Data Analysis

Input: Healthcare datasets D0, D1,...., Dn containing patient details and labels for readmission, Initial global model parameters 'M0', Number of FL rounds = 10, public and private encryption keys ('pk' and 'sk') for Homomorphic Encryption.

Output: Final global model MT after 't' rounds of FL, Model updates securely stored on Google Drive.

Function FederatedLearning():

1: M0 ← InitializeGlobalModel() - Server Initialization.

2: (pk, sk) ← GenerateEncryptionKeys() - Generating homomorphic encryption keys.

3: DistributeKey(pk) - Sends public key to all participating clients.

4: For t from 1 to 10 do: - Performs ten rounds of Federated Learning.

5: SendModelToClients(Mt) - Server sends current global model to all participating clients.

6: Each client performs local training on their dataset, calculating local model updates.

7: For each client i in 1 to N do: - client i performs local training.

8: Di ← GetClientDataset(i) - Loading and preprocessing Client 'i' healthcare dataset 'D'.

9: ΔMi^t ← LocalTraining(Di, M_t) - Compute local model updates using their dataset.

10: EncryptedΔMi^t ← Encrypt(ΔMi^t, pk) - client i encrypt model updates using HE.

11: Server-Side Aggregation – client i send their encrypted updates to the Central Server.

12: SendEncryptedUpdateToServer(i, EncryptedΔMi^t) - (Homomorphic Aggregation)

13: Encrypted(M_{t+1}) ← Encrypted(M_t) - Start aggregation with current model.

14: For each client i in 1 to N do: - Perform homomorphic addition on encrypted updates.

15: Encrypted(M_{t+1}) ← Encrypted(M_{t+1}) + Encrypted(ΔMi^t) - Aggregate updates.

16: M_{t+1} ← Decrypt (Encrypted(M_{t+1}), sk) - Decrypt the aggregated result.

17: SendModelToClients(M_T) - Send final global model back to clients.

18: Client-Side Verification (AFTER all rounds)

19: Client decrypts their original update and compares with the final global model:

20: If ΔMi^T == Decrypt(EncryptedΔMi^T, sk) then Print ("Privacy Preserved for Client", i)

21: Else Print("Privacy Breach Detected for Client",i) and then Abort() - if there's a mismatch

22: Return M_T - Returning the Final global model after 10 rounds.

23: Load and store the encrypted model updates on Google Drive.

Algorithm Steps

1. Initialize the Global Model -

- **Action**: The central server initializes the global model (M0).
- **Details**: The model can be a neural network, logistic regression, or any other machine learning model suitable for the problem (e.g., hospital readmission prediction).
- **Function**: M0 ← InitializeGlobalModel()

2. Generate Encryption Keys Action: The server generates a pair of Homomorphic Encryption keys:

  - **Public Key (pk)**: Used by clients to encrypt their local updates.
  - **Private Key (sk)**: Retained by the server to decrypt the aggregated updates.
- **Function**: (pk, sk) ← GenerateEncryptionKeys()

3. Distribute Public Key to Clients

- **Action**: The server sends the public key (pk) to all participating clients.
- **Purpose**: Ensures that clients can encrypt their updates before sending them back to the server.
- **Function**: DistributeKey(pk)

4. Federated Learning Process Begins (10 Rounds)

- **Action**: The algorithm performs **10 rounds** of FL.
- **Function**: For t from 1 to 10 do:

5. Send Current Model to Clients -

- **Action**: The server sends the current global model (Mt) to all participating clients at the start of each round.
- **Purpose**: Ensures all clients work on a synchronized version of the model.
- **Function**: SendModelToClients(Mt)

6. Client-Side Local Training

Each client trains the model on its local dataset:

1. **Load and Preprocess Data**:
    - **Action**: The client loads and preprocesses its healthcare dataset (Di).
    - **Function**: Di ← GetClientDataset(i)
2. **Train Locally**:
    - **Action**: Each client trains the global model (Mt) using its local dataset, producing an update (ΔM_i^t).
    - **Function**: ΔM_i^t ← LocalTraining(D_i, M_t)

7. Encrypt Local Updates

- **Action**: Each client encrypts its local model updates (ΔM_i^t)) using the public key (pk).
- **Purpose**: Ensures the updates are secure during transmission to the server.
- **Function**: EncryptedΔM_i^t ← Encrypt(ΔM_i^t, pk)

8. Send Encrypted Updates to Server

- **Action**: Clients send their encrypted updates (Encrypted ΔM_i^t) to the server.
- **Purpose**: Enables secure aggregation of model updates without exposing raw data.
- **Function**: SendEncryptedUpdateToServer(i, EncryptedΔM_i^t)

9. Server-Side Aggregation

The server aggregates the encrypted updates:

1. **Initialize Aggregation**:
    - **Action**: Begin aggregation with the current global model in encrypted form.
    - **Function**: Encrypted(M_{t+1}) ← Encrypted(M_t)
2. **Perform Homomorphic Addition**:
    - **Action**: Add encrypted updates from each client to the global model:
        - Encrypted(M_{t+1}) ← Encrypted(M_{t+1}) + Encrypted(ΔM_i^t)
    - **Purpose**: Allows secure computation without decrypting individual updates.

10. Decrypt the Aggregated Model

- **Action**: The server decrypts the aggregated global model using the private key (sk).
- **Purpose**: Produces the updated global model (M_{t+1}) for the next round.
- **Function**: M_{t+1} ← Decrypt(Encrypted(M_{t+1}), sk)

11. Repeat for 10 Rounds

- Steps 5–10 are repeated for all 10 rounds, gradually improving the model using secure contributions from all clients.

12. Send Final Model Back to Clients

- **Action**: After the last round, the server sends the final global model (M_T) to all clients.
- **Purpose**: Clients can use or validate the trained model locally.
- **Function**: SendModelToClients(M_T)

13. Client-Side Verification

- **Action**: Each client verifies the integrity of their contribution:
    1. Decrypt their original update: ΔM_i^T ← Decrypt(EncryptedΔM_i^T, sk)
    2. Compare with the final global model.
- **Validation**:
    0. If updates match, print: "Privacy Preserved for Client", i
    1. If there's a mismatch, print: "Privacy Breach Detected for Client", i and abort.

14. Store Updates Securely on Google Drive

- **Action**: The server securely stores all encrypted updates and logs on Google Drive.
- **Purpose**: Provides a backup and ensures compliance with regulations like **GDPR** or **HIPAA**.
- **Function**: Load and store the encrypted model updates on Google Drive

Outcomes

This study assesses how well Federated Learning (FL) and Homomorphic Encryption (HE) work together to analyze healthcare data securely. Predictive accuracy, model robustness, privacy security, and computational efficiency across various healthcare facilities are the main criteria that are being examined.

Predictive Accuracy -

By compiling encrypted updates from multiple healthcare facilities, the Federated Learning model was able to achieve excellent predicted accuracy for patient outcomes. The accuracy of the FL approach was very close to that of a centralized model with complete data access, demonstrating FL's ability to produce high-quality predictive results while protecting sensitive data confidentiality. To show the accuracy improvement over training rounds, we used a line graph.

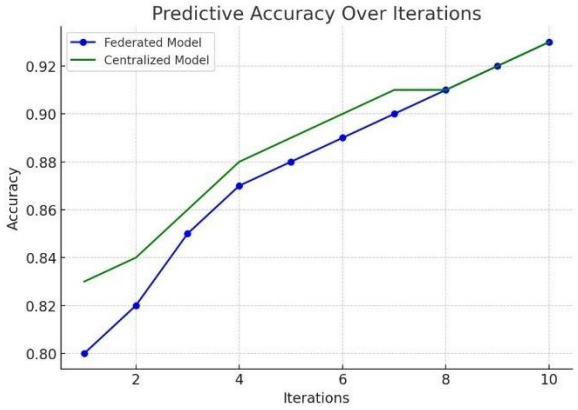

Fig 10. Predictive Accuracy over Iterations

During the validation stage, this graph contrasts the FL model's predicted accuracy with that of a centralized model. The findings confirm the efficacy of the FL technique in practical applications by showing that, in spite of its decentralized structure, its predictive powers are still strong.

Model Robustness - Strong robustness was demonstrated by the model during several training cycles, with little difference between local and global models. The encrypted updates from each participating institution's local model increased the accuracy of the model with each iteration. FL is a promising method for healthcare data analysis since this research indicates that it may successfully use diverse data sources without compromising prediction quality.

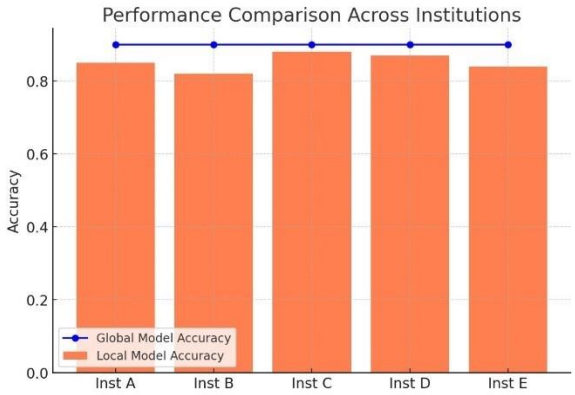

Fig 11. Performance of different hospitals

The above bar graph shows comparison of performance and the prediction accuracy attained by several institutions, demonstrating this steadiness. Because it continued to perform well even when local data distributions changed, the FL model's constancy is very noteworthy.

Privacy Retention -

Homomorphic encryption was essential to maintaining data privacy. The method ensured that raw patient data was never made public at any stage of the procedure by encrypting all model updates. Following privacy guidelines guarantees conformity to strict laws like the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA). Every privacy-preserving step of the procedure, from encrypted data transmission to secure aggregation and eventual decryption, is further explained via a flow diagram of the model architecture.

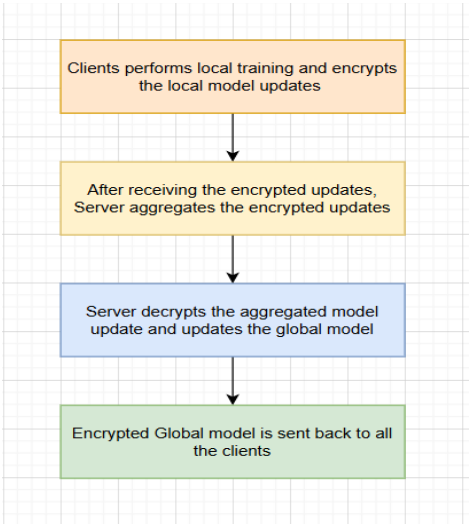

Fig 12. Schematic Illustration of HE of model updates

This graphic illustrates how sensitive healthcare data can be studied without jeopardizing individual confidentiality, which supports the study's focus on privacy assurance.

Computational Efficiency -

The use of Homomorphic Encryption in conjunction with Federated Learning showed significant computational efficiency gains over traditional FL models. The method ensured controllable processing speeds while maintaining prediction accuracy.

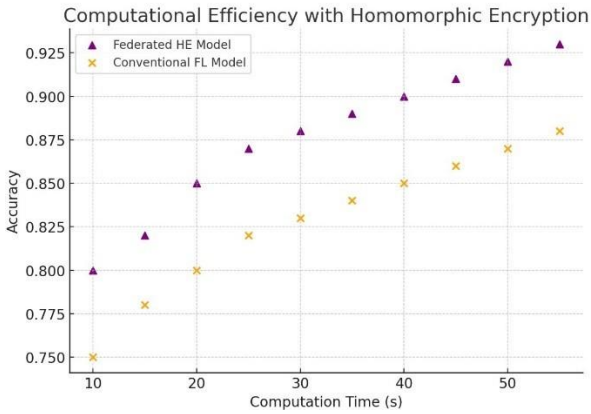

Fig 13. Computational Efficiency with Homomorphic Encryption

The effectiveness of HE in enabling real-time secure healthcare data analysis is clearly demonstrated by a scatter plot that contrasts accuracy with processing time each round. The findings show that the integrated model is appropriate for deployment in settings where speed and confidentiality are crucial since it maintains anonymity while still functioning well.

Advantages of Federated Learning

Federated Learning (FL) is a potent paradigm for distributed machine learning because it provides notable benefits in terms of data privacy, ownership, scalability, and cooperation. Since raw data stays on local devices and never leaves its source, FL's significant emphasis on data privacy is one of its core advantages. By doing this, user privacy is protected and sensitive data is kept private. For example, in the healthcare industry, patient data confidentiality is maintained by keeping patient records inside hospitals and sharing only model updates.

Data ownership is another important benefit since FL gives individuals complete control over their data. This resolves issues with disclosing private or confidential information to outside parties. Banks, for instance, can work together to build machine learning models on transaction data without disclosing client information. For example, to protect their privacy and ownership rights, banks, for instance, can work together to train machine learning models on transaction data without disclosing client information to other organizations.

Because it can incorporate millions of dispersed devices, like smartphones or Internet of Things devices, into the training process, FL also excels at scalability. Applications like voice assistant personalization, like Google Assistant or Siri, can be used by a large number of users thanks to this capacity. Furthermore, because only model updates are exchanged during training, FL

greatly minimizes the need to transmit raw data. As demonstrated by applications such as updating predictive text models without uploading whole typing histories, this saves bandwidth needs and communication overhead. The capacity of FL to generate customized models is another noteworthy benefit. Devices can adjust models to their unique data distributions by enabling local adaptation, which improves user performance. For example, recommendation algorithms, for instance, can change over time to better suit user preferences. Last but not least, FL promotes cooperation by enabling groups to cooperate on common objectives without needing centralized access to private information. In situations like pharmaceutical research, where businesses can work together on drug development while maintaining the privacy of proprietary research data, this is very helpful.

Because of these benefits, FL is the best method for large-scale, collaborative, privacy-preserving machine learning applications in a variety of industries.

Fig 14. Advantages of the proposed Model

Challenges of Federated Learning

1. **Data Imbalance**
    a. Data is dispersed among several devices or institutions in federated learning. These datasets are frequently unbalanced, which means that participant distribution of classes or sample size varies greatly by class.

    b. **Impact:** Particularly for underrepresented classes, this imbalance may result in biased model updates and decreased global model performance.

    c. **Solution Approaches**: This problem can be resolved by employing strategies such as re-weighting the loss function, data augmentation, or synthesizing missing data.

2. **Statistical Heterogeneity**

Non-IID (Independent and Identically Distributed) sources are frequently the source of the data on various devices. For instance, there may be significant differences in user environments, demographics, or preferences.

   a. **Impact**: Updates based on non-IID data slow down convergence and decrease accuracy since they are inconsistent with the global model objective.
   b. **Solution Approaches**: Different data distributions can be accommodated by algorithm modifications such as FedProx or customized FL models.

3. **System Heterogeneity**

The processing power, network speed, and energy limits of devices in federated learning networks vary.

   a. **Impact**: This discrepancy hinders update synchronization and can prevent less powerful devices from fully engaging.
   b. **Solution Approaches**: Adaptive participation techniques and asynchronous update approaches can help lessen the difficulties caused by system heterogeneity.

4. **Resource Allocation**

Participating devices must have a substantial amount of memory, processing power, and network bandwidth for federated learning to work.

   a. **Impact**: Inadequate resources could lead to insufficient updates or abandoned members, which would lower the overall model's quality.
   b. **Solution Approaches**: Resource consumption can be decreased by using device-specific model optimization and resource-aware training methods.

5. **Privacy Concerns**
   Sharing model updates can nevertheless expose private information via methods like model inversion or gradient assaults, even if FL is intended to safeguard privacy by storing raw data on devices.
   a. **Impact**: Organizations or individuals may be discouraged from participating in FL frameworks due to privacy issues.
   b. **Solution Approaches**: Secure multi-party computation (SMPC), homomorphic encryption, and differential privacy can all improve data privacy.

6. **Expensive Communication**

FL requires that devices and the central server communicate model updates on a regular basis, which can be expensive in terms of latency and bandwidth, especially in large networks.

a. **Impact**: High communication costs cause edge devices to use more energy and slow down the training process.
b. **Solution Approaches**: Methods like as sparsification, communication-efficient protocols, and model compression lower the communication overhead.

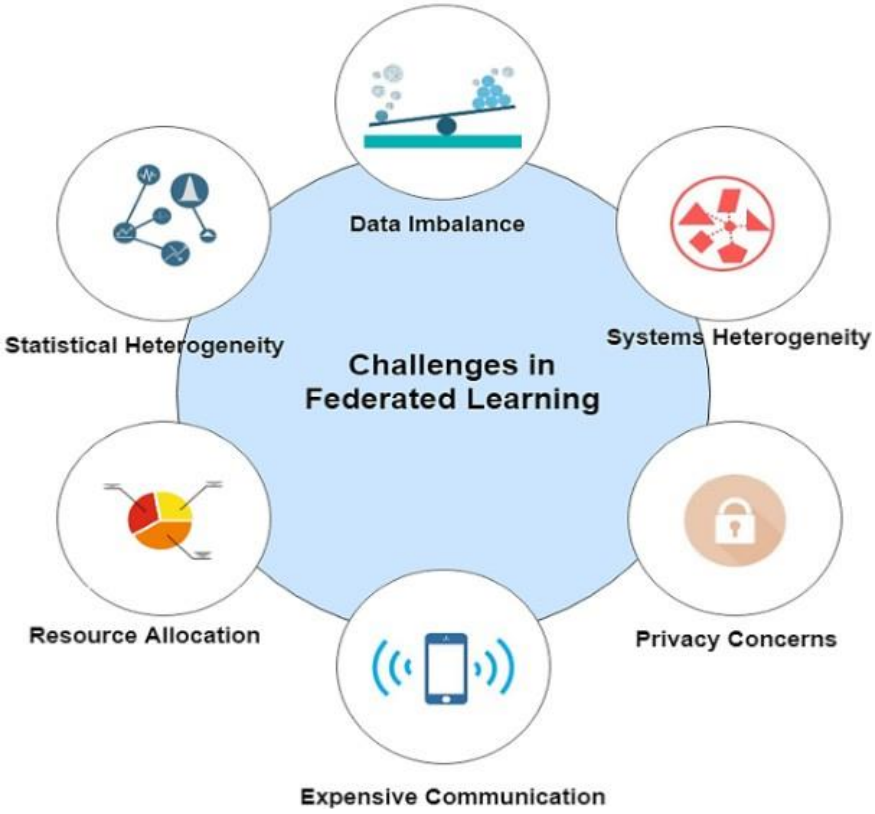

Fig 15. Challenges in Federated Learning

Comparative Study

This section compares the proposed Federated Learning (FL) with Homomorphic Encryption (HE) framework to traditional FL models and other privacy-preserving methods used in

healthcare data analysis. The analysis covers aspects such as accuracy, computational efficiency, scalability, and privacy retention.

Table 1: Comparative study with related schemes

Scheme	Objective	Methodology	Primary Contribution
Traditional Federated Learning (FL)	Learning collaboratively using decentralized data.	aggregates raw updates and shares model updates on the server.	Achieves reasonable model accuracy, but lacks strong data encryption, putting privacy at risk in collaborative healthcare.
Differential Privacy (DP)	Protection of privacy with statistical assurances.	introduces noise into model updates.	obscures important patterns to protect privacy, although noise addition may lower model accuracy, particularly in smaller datasets.
Secure Multi-Party Computation (SMPC)	Cooperative calculations that maintain high privacy.	Data is encrypted for calculations.	It is less effective for large-scale or real-time analysis due to its high processing costs, despite its strong privacy assurances.
Proposed FL with HE	Decentralized analysis of medical data with improved privacy.	combines HE and FL to update models securely.	achieves near-centralized model performance with a lower computing load, striking a balance between high accuracy and privacy, making it perfect for scalable healthcare applications.

Federated Learning (FL) vs Centralized Learning (CL)

The ways that Federated Learning (FL) and Centralized Learning (CL) manage data, privacy, scalability, communication, and system needs are very different. Since raw data never leaves the source, FL ensures maximum privacy by keeping data on local devices and sharing only model changes. CL, on the other hand, mandates that data be gathered and kept in a centralized server or data center, which raises the possibility of privacy violations because of centralized exposure.

Because FL uses a lot of dispersed devices, it is very scalable, which makes it appropriate for uses like smartphone model training. However, because CL depends on a central server, which can act as a bottleneck, it has limited scalability. Whereas CL necessitates conveying raw data, which greatly increases communication overhead, FL communication entails sending model changes, which are lower in size and require less bandwidth. FL is used, for instance, to train predictive text models on cellphones, such as Gboard, where distributed learning and anonymity

are crucial. On the other hand, CL is frequently used to train recommendation engines that rely on centralized user activity records, which aggregate data in one place.

Because FL works with heterogeneous and resource-constrained devices, it is challenging to manage a range of hardware capabilities. CL, on the other hand, uses a lot of processing power and operates in controlled environments, which simplifies system administration. The way FL handles imbalanced and non-IID (non-independent and identically distributed) data, which makes convergence challenging, is another important difference. Centralized data collection, on the other hand, facilitates CL by ensuring IID assumptions and streamlining training.

While FL is less susceptible to central server incursions, it can be subject to model inversion or gradient attacks. CL is particularly vulnerable to breaches because all of the data is stored in one place. For example, FL is used, for instance, to train predictive text models on cellphones, such as Gboard, where distributed learning and anonymity are crucial. On the other hand, CL is frequently used to train recommendation engines that rely on centralized user activity records, which aggregate data in one place.

Because of these differences, FL is better suited for distributed, privacy-focused applications, but CL is still the best option for centralized, resource-intensive situations.

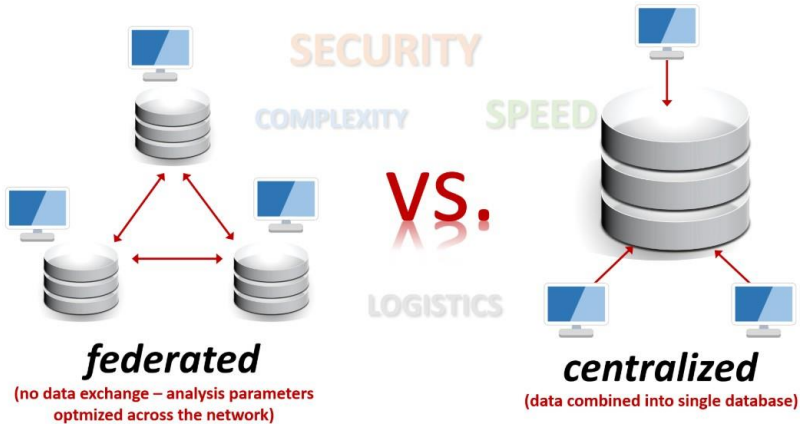

Fig 16. Federated Learning (FL) vs Centralized Learning (CL)

Future scope

The integration of Federated Learning (FL) and Homomorphic Encryption (HE) for privacy-preserving healthcare data analysis represents a groundbreaking advancement in the field of medical data science. This transformative approach enables the development of secure and decentralized systems, ensuring the protection of sensitive patient information while fostering collaborative innovation among healthcare institutions. In an era where compliance with stringent privacy regulations such as HIPAA and GDPR is imperative, the adoption of FL and HE offers a viable solution by maintaining data encryption throughout the analytical process.

This ensures that patient confidentiality is preserved while simultaneously promoting the growth of data-driven research.

The future potential of FL and HE in healthcare is immense, with opportunities for transformative applications on the horizon. For instance, real-time integration of data from Electronic Health Records (EHRs) and the Internet of Medical Things (IoMT) could revolutionize predictive analytics by enabling early detection of critical conditions and the delivery of personalized treatment recommendations. Moreover, ongoing efforts to optimize HE algorithms for reduced computational overhead will enhance scalability, making these technologies suitable for managing larger and more complex datasets. Such advancements could expand their applications to include fields like genomic research, drug discovery, and epidemiology, significantly advancing medical science and healthcare delivery.

Beyond healthcare, the scalability and versatility of FL and HE hold promise for numerous other privacy-sensitive sectors, including finance, telecommunications, and public governance. Applications could range from secure fraud detection and user behaviour analysis to the evaluation of sensitive census data, showcasing the far-reaching implications of these technologies.

By establishing FL and HE as a standard practice, the global healthcare ecosystem could benefit from interoperable systems that facilitate secure knowledge sharing across institutions. This would further enhance AI applications in healthcare by integrating cutting-edge technologies such as Natural Language Processing (NLP) and Computer Vision for diagnostic automation and patient management. Additionally, the decentralized nature of FL is particularly well-suited for fostering cross-border collaborations, enabling underrepresented regions to contribute to and benefit from collective medical research efforts. Such initiatives could uphold ethical standards while addressing critical global challenges, including pandemics, thus paving the way for a more inclusive and equitable future in healthcare and beyond.

Moreover, this model has the potential to inspire multidisciplinary research in AI, cryptography, and healthcare, fostering the development of new professionals equipped to tackle privacy-preserving challenges. By improving predictive accuracy and reducing the costs associated with data breaches, the FL and HE framework not only enhances operational efficiency but also contributes to better patient outcomes and a more secure healthcare ecosystem.

Conclusion

In this paper, we propose Privacy Preserving Using Federated Learning for Collaborative Healthcare Data Analysis, a privacy-preserving FL frame work based on homomorphic encryption, which guarantees both privacy and security on health care data. The experimental results on the DR dataset show that the proposed scheme achieves high classification accuracy and privacy preserving but low communication overhead. Privacy Preserving Using Federated

Learning for Collaborative Healthcare Data Analysis contributes to accelerating the application of FL for privacy preserving in the healthcare imaging domain.

Currently, there is still some scope for further improvements. Firstly, this work relies on homomorphic encryption which will cause high communication overhead and ciphertext explosion. In future work, we will implement more efficient privacy protection of federated learning with the help of other privacy-preserving technologies, like blockchain-based technology Secondly, the poisoning attack will also destroy the model integrity, which might be explored in future studies.

Furthermore, this approach holds significant promise for advancing predictive healthcare modelling by securely utilizing decentralized datasets, thereby enhancing patient outcomes and streamlining healthcare delivery. Future efforts should focus on optimizing HE processes to minimize computational overhead, incorporating real-time data from Electronic Health Records (EHRs), and extending the framework to accommodate larger and more diverse datasets. These advancements could position the approach as a gold standard for secure and privacy-conscious healthcare data analysis.

References

1. School of Digital Science, Universiti Brunei Darussalam, Gadong, Brunei Darussalam Dig Connectivity Research Laboratory (DCRLab), Kampala, Uganda.

2. Protecting Data from all Parties: Combining Homomorphic Encryption and Differential Privacy in Federated Learning. Institute LIST, CEA, France.

3. Privacy-Preserving Federated Learning based on Multi-key Homomorphic Encryption (Xidian University, China, Aalto University, Finland).

4. Riazi, H., et al. (2020). *Privacy-Preserving Data Analysis: A Comprehensive Survey*. 2020 IEEE 14th International Conference on Smart City (SmartCity).

5. S. K. K. and A. Z. A. (2020). *Distributed Machine Learning for Healthcare: A Systematic Review*. Springer.

6. Hard, A., et al. (2018). *Federated Learning for Healthcare: Systematic Review and Guidelines*. *Journal of Biomedical Informatics*.

7. Sheller, M. J., et al. (2020). *Federated Learning in Medicine: Systematic Review. Journal of Medical Internet Research*.

8. Min, H., and Wang, F. (2021). *Machine Learning in Healthcare Informatics*. Springer.

9. Arnaud Grivet Sébert(2023). *Combining differential privacy and homomorphic encryption for privacy preserving*
collaborative machine learning. Artificial Intelligence [cs.AI]. Université Paris-Saclay, 2023.