ACCSCIENCE PUBLISHING

Artificial Intelligence in Health

ORIGINAL RESEARCH ARTICLE

A hierarchical federated learning-based health stack for future pandemic preparedness

Rojalini Tripathy[1], Asmit Balabantaray[2], Nisarg Shah[2], Prashant Kumar Jha[3], Ajay Kumar Gogineni[1], Atri Mukhopadhyay[1], Kisor Kumar Sahu[3,4]*, and Padmalochan Bera[1]*

[1]School of Electrical and Computer Sciences, Indian Institute of Technology Bhubaneswar, Odisha, India

[2]School of Mechanical Sciences, Indian Institute of Technology Bhubaneswar, Odisha, India

[3]School of Minerals, Metallurgical and Materials Engineering, Indian Institute of Technology Bhubaneswar, Odisha, India

[4]Virtual and Augmented Reality Center of Excellence, Indian Institute of Technology Bhubaneswar, Odisha, India

Abstract

The COVID-19 pandemic, one of the most disruptive global health crises in recent history, exposed critical vulnerabilities in existing healthcare infrastructure. Given the likelihood of future pandemics, it is essential to build a resilient, collaborative, synergistic, data-driven, and intelligent digital healthcare software. It should be meticulously designed and selectively curated to enhance early detection, rapid response, and efficient containment of outbreaks. In this article, we propose a federated learning (FL)-based health stack that prioritizes privacy while fostering collaborative intelligence among hospitals or client nodes. Our framework incorporates hierarchical FL, Byzantine-resilient information-theoretic FL (ByITFL), homomorphic encryption, and blockchain-based smart contracts to ensure secure collaboration among healthcare institutions without sharing raw data. Hierarchical FL leverages multilevel model aggregation to enhance model convergence, scalability, and resilience. ByITFL strengthens security by incorporating trust mechanisms and information-theoretic privacy scoring, while blockchain-based smart contracts ensure transparent, verifiable coordination among participating nodes. Furthermore, deep vulnerability detection using optimized averaged stochastic gradient descent weight-dropped long short-term memory models may further enhance the framework's security, enabling threat identification during decentralized data exchanges. Experimental results show that the proposed hierarchical FL model achieves 94.23% accuracy on the modified National Institute of Standards and Technology dataset, outperforming federated averaging (92.66%) under the same environments. In addition, communication analysis proved that the overall transmission is minimized by collecting updates at local servers before sending them to central servers. Therefore, it is nearly a future-ready technology that can be implemented without many geopolitical issues, even in the case of hypersensitive global situations.

Keywords: Global pandemics; Health stack; Federated learning; Medical data privacy; Machine learning

***Corresponding authors:**
Kisor Kumar Sahu (kisorsahu@iitbbs.ac.in);
Padmalochan Bera (plb@iitbbs.ac.in)

Citation: Tripathy R, Balabantaray A, Shah N, *et al*. A hierarchical federated learning-based health stack for future pandemic preparedness. *Artif Intell Health*. 2025;2(4):75-91. doi: 10.36922/AIH025080013

Received: February 21, 2025

1st revised: May 22, 2025

2nd revised: June 8, 2025

Accepted: June 9, 2025

Published online: June 30, 2025

1. Introduction

1.1. General overview: The big picture

The emergence of the COVID-19 pandemic in 2019 shook the foundations of human civilization. While it was not the deadliest pandemic in history, it ranks fifth in terms of death toll, with an official World Health Organization (WHO) estimate of 7 million deaths.[1] When ranked by death toll, the following pandemics are the deadliest, in descending order: (i) the Spanish flu (17 – 100 million deaths, 1918 – 1920),[2] (ii) the Plague of Justinian (15 – 100 million deaths, 541 – 549),[3] (iii) human immunodeficiency virus/acquired immunodeficiency syndrome (approximately 43 million deaths, 1981–present),[4] and (iv) the Black Death (7 – 35 million deaths, 1346 – 1353).[4]

What sets COVID-19 apart from previous pandemics is that it occurred in a highly globalized, post-internet world, where information could spread instantaneously.[5] Combined with significant advancements in modern sciences, particularly in medical and computational sciences, it was shocking to see how the pandemic exposed the limitations of human technological capabilities. As argued in previous studies,[5] although we had the components of a modern technological infrastructure to create a formidable defense, we failed to synergize them effectively to prevent the pandemic's escalation.

Therefore, the central question addressed in this article is: What would the next-generation computational infrastructure look like to effectively combat future pandemics? Such infrastructure should be deployable within current medical record-keeping systems, which involve heterogeneous data types and regulations (mostly privacy-related). To stimulate this discussion, we propose a federated learning (FL)[6,7]-based, global machine learning (ML) architecture with the potential to address this key issue. While we do not claim this is the ultimate solution (though we believe it may be), the main purpose of this article is to motivate the journey towards that goal. For brevity, we will highlight one or a few methods for each step, although multiple viable solutions may exist. A comprehensive review of FL applications in smart healthcare is presented elsewhere;[8-10] hence, we will only summarize key technical preliminaries of FL here.

In the absence of a global enforcing agency for administering critical pandemic/medical issues (as the WHO lacks such authority), a decentralized approach offers the most promising pathway for global acceptance and implementation. FL is a decentralized ML approach in which clients (e.g., organizations, healthcare units, mobile devices, and sensors) collaboratively train a shared model under the coordination of a central server without exchanging raw data.

The FL process typically begins with the central server initializing a global model and distributing its parameters to selected clients. Each client independently trains the model using its local dataset, ensuring data privacy by keeping raw data on the local device. After local training, clients send model updates, such as model gradients, to the central server. The server then aggregates these updates, commonly using methods such as federated averaging (FedAvg),[11] to update the global model. The updated model is then redistributed to the clients for the next round of training. This cycle continues iteratively until the model converges or reaches a predefined accuracy threshold. Finally, the trained global model is deployed for use.

As discussed, FL eliminates the need to transfer raw data to a central server; instead, only model parameters are exchanged. This approach significantly reduces the risk of data breaches, as sensitive information remains on the client's side. FL thus upholds data privacy and security, making it one of the highly accepted protocols across countries, communities, and agencies. For this reason, it has been chosen in this article as the foundational model for addressing future pandemics.

We envision that combating future pandemics will require the holistic deployment of the entire computing infrastructure, including, but not limited to, the Internet-of-Things (IoT) devices, sensors, edge computing, and cloud infrastructure. While FL offers privacy-preserving model training, its integration with emerging technologies, such as edge intelligence and IoT, in healthcare introduces new challenges that must be addressed. Recent studies have highlighted the growing need for edge intelligence and distributed learning in health care.[12,13] For example, research presents an overview of mobile health systems and IoT technologies, focusing on system architecture and data integration strategies.[14] However, centralized or cloud-based models often require raw or partially processed data to be transmitted, raising concerns about data security in sensitive applications, such as health care. In another study,[13] the authors discussed the role of artificial intelligence in IoT-enabled smart healthcare applications, with a primary focus on lightweight neural network architectures deployed directly on edge devices. Yet, the study does not address the security risks associated with transmitting trained model parameters from edge devices. This omission overlooks potential risks such as adversaries intercepting and reverse-engineering the model parameters to extract sensitive information, or manipulating them to induce bias or abnormal model behavior. In our proposed framework, we use a

privacy-preserving, hierarchically structured FL approach to address these challenges.

We present a secure and scalable FL framework using a hierarchical structure and homomorphic encryption (HE) for privacy-preserving, collaborative model training. Then, we discuss the main features of Byzantine-resilient information-theoretic FL (ByITFL),[14] which offers privacy-preserving aggregation and robust protection against communication-based attacks in FL through standard security techniques. Subsequently, we illustrate an implementation of a deep-learning protocol based on modern convolutional neural network (CNN) architectures that efficiently detect past pandemics like COVID-19 and diseases like lung cancer, with the aim of improving accuracy and computational efficiency by leveraging transfer learning, clustering algorithms, and region-of-interest (ROI) analysis. Blockchain/smart contracts play a key role in establishing digital trust and maintaining transparency, both of which are critical for success in a global endeavor, such as the fight against pandemics. Unfortunately, they are not free from external vulnerabilities. We discuss detecting vulnerabilities in smart contracts using an improved averaged stochastic gradient descent weight-dropped long short-term memory (AWD-LSTM) model, which achieves high accuracy and F1 score by incorporating opcode review analysis and addressing class imbalance in smart contract datasets. In Section 3, we experimentally evaluate our proposed framework and present a detailed cost analysis of the hierarchical FL[15] framework, showing its scalability through efficient communication and computation at client, local server, and central server levels. Communication costs are optimized using hierarchical data aggregation, while computation costs are minimized through localized operations, ensuring scalability and security. In Section 4, we address the issue of non-independent and identically distributed (non-IID) datasets and heterogeneity considerations, both of which are critical for the efficient implementation of our idea.

The FL model that we have described (including hierarchical FL, ByITFL, and HE) is more realistic, as it takes into consideration real-world applications. It is also suggested to use institutional incremental learning[10] and cyclic institutional incremental learning.[10] FL achieves a better rate of model improvement than data-private collaborative learning methods. Moreover, to compare the rates of model improvement, a global validation "dice-over-epoch" (where the model trains over epochs and the metric is computed on both training and validation data) for all collaborative methods showed that FL training converges relatively quickly to the same performance as collaborative data sharing training.

1.2. Technical preliminaries

In this section, we discuss some technical preliminaries used in the architectural design of the proposed hierarchical FL, such as static and dynamic datasets, hierarchical learning,[16] and HE.[17] A static dataset does not change over time, whereas a dynamic dataset is continuously or periodically updated with new data. HE is a privacy-preserving technique that enables computations, such as addition and multiplication, to be performed directly on encrypted data. When data or model parameters are transmitted to a server, there is a risk of interception or tampering by attackers, potentially compromising model integrity and predictions. In the proposed framework, we employ HE to encrypt model parameters before transmission. These encrypted parameters are then aggregated on the server using homomorphic addition and multiplication operations, ensuring both data privacy and secure computation. FL frameworks may suffer from scalability limitations due to communication bottlenecks that arise when a large number of clients frequently transmit model updates to a centralized server. This issue becomes more pronounced in distributed environments, such as health care.

To address this, we employ hierarchical learning.[18] Hierarchical learning is a structured learning process organized across multiple levels, where higher levels aggregate and refine knowledge from lower levels to improve performance and scalability. It has a multi-tiered structure comprising a central server along with multiple local servers at each level. This structure efficiently handles gradient aggregation and transmission in parallel, which reduces overall communication overhead and ensures privacy. Figure 1 represents a hierarchical learning architecture suitable for deployment in the global healthcare system.

We experimentally evaluated our proposed framework using a proxy image classification dataset, implementing a hierarchical FL architecture and transmitting model parameters through HE. We compared its performance against standard FL lacking hierarchical structure and encrypted communication. Our experimental results demonstrate that the hierarchical framework achieved higher model accuracy and reduced training time compared to generic FL. The hierarchical structure effectively distributes the computational and communication load, enabling faster convergence. Additionally, the use of HE ensures that sensitive information remains confidential throughout the training process, thereby satisfying privacy requirements without significantly compromising efficiency. The proposed model is capable of incorporating both static and dynamic data. However, future pandemics may involve highly distributed or heterogeneous data types.

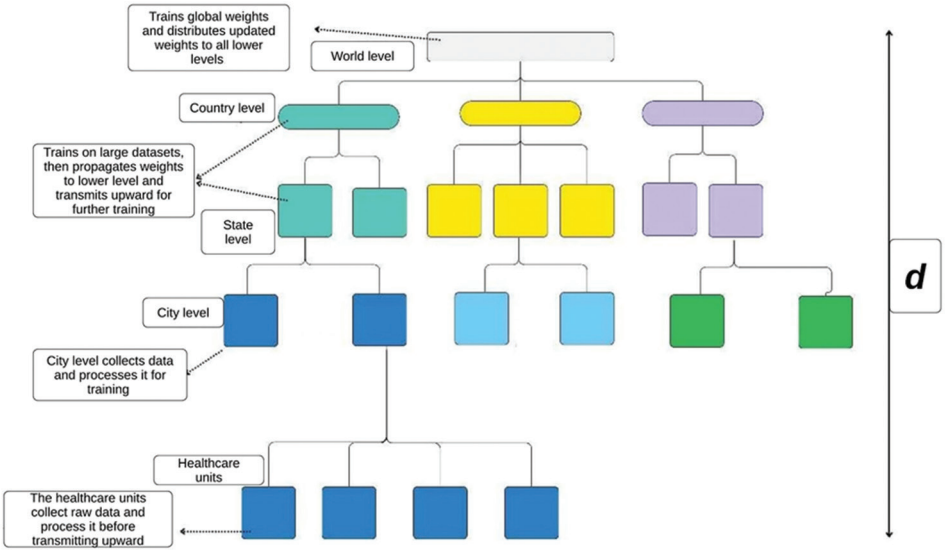

Figure 1. Schematic representation of the hierarchical learning framework implemented in the proposed FL scheme. Here, ***d*** represents the depth of the hierarchy, and root may represent the central organization that controls all subordinate levels (e.g., World Health Organization).

Our framework can be extended to train multiple specialized models at the central unit, depending on data characteristics and application requirements. In addition, given the assumption that future pandemics will generate massive volumes of data, we anticipate that the hierarchical structure of our framework can be scaled accordingly by incorporating additional aggregation layers or nodes to efficiently manage the increased load and maintain performance.

2. Proposed FL framework

In this section, we described our proposed secure and scalable FL framework, capable of incorporating dynamic data during model training. In collaborative medical research, to enhance predictive diagnostic accuracy, multiple healthcare data owners aim to train a unified model (or a set of unified models, depending on the nature of a future pandemic). The trained model should be available locally to all collaborators. In scenarios where a single data owner operates multiple healthcare units, privacy concerns are minimized, as such organizations can consolidate their data internally without external exposure. For example, in the United States (US), Kaiser Permanente operates a network of hospitals and clinics as a unified entity. Similarly, in India and internationally, Apollo Hospitals manages an extensive network of hospitals, clinics, and pharmacies. The proposed framework allows these data owners to centralize patient data for analysis and predictive modeling while ensuring strict internal control and compliance with privacy regulations. Figure 2 illustrates the architectural structure of our proposed framework. Here, data owners refer to entities (such as Kaiser Permanente or Apollo Hospitals), while data units include their affiliated hospitals, clinics, and pharmacies. These data owners interact with designated local servers rather than directly with the central server. Each data owner conducts private training on its localized data and communicates model parameters to the respective local server. The local servers aggregate these parameters and relay them to the central server.

Our proposed hierarchical FL framework effectively addresses security risks during communication phases and at the server level. The hierarchical structure enables scalability by accommodating a wide range of healthcare data, irrespective of its volume.

2.1. Secure model training procedure

The proposed framework consists of three primary entities: the central server, local servers, and local healthcare data holders acting as clients. The central server is responsible for global aggregation, i.e., the final updated parameters will be computed at the central server. The local or regional servers are responsible for aggregation at the cluster or regional level, while the client organizations train their local models on the datasets they manage locally.

Let S and L_i represent the central server and a regional server, respectively, and n denote the number of regional servers, where $1 \leq i \leq n$. Each regional cluster may contain a different number of clients. Let p be the number of clients under L_i, with each client denoted as C_{ij}, where i corresponds to the local server and j represents the client, where $1 \leq j \leq p$. Every client C_{ij} holds a private dataset D_{ij}.

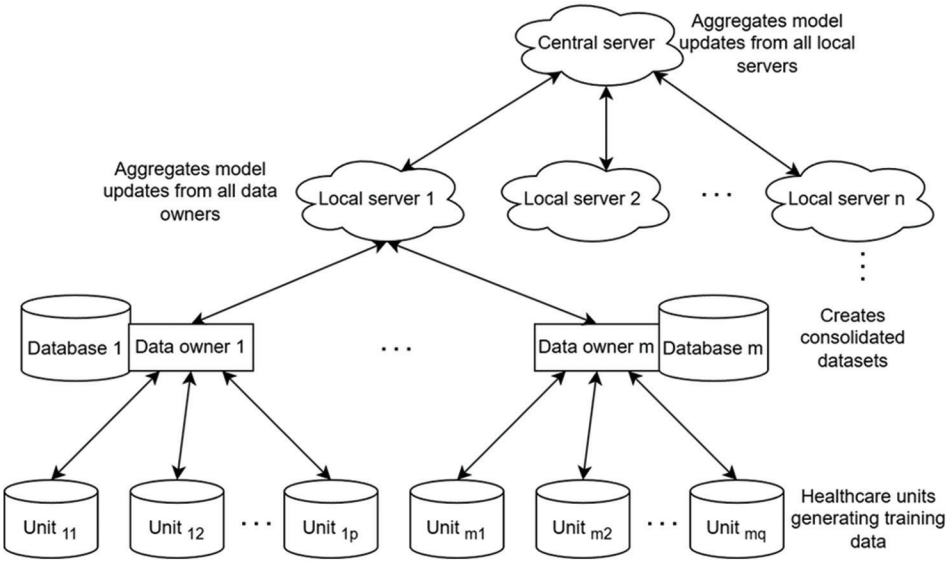

Figure 2. Architectural structure of the proposed hierarchical federated learning

The training process begins with the central server (S) broadcasting the initialized model parameters W_0 to all local servers, ensuring uniformity across local models for their cluster-level distributions. After receiving the initialized model parameters, each C_{ij} trains its local model on a batch of data b_r^{ij}, sampled from its dataset D_{ij}, and generates local model gradients G_r^{ij} for round r. This is represented as:

$$G_r^{ij} \leftarrow train(W_0, b_r^{ij}) \quad \text{(I)}$$

where r represents the current round index, and R is the total number of training rounds. The symbol "←" denotes the assignment operator, indicating that the result of the operation on the right-hand side is assigned to the variable on the left-hand side. The proposed framework uses HE[19] operations to ensure the security of the model gradients during client–server communications and employs a hierarchical structure to enhance scalability. The depth of the hierarchy can be increased to ensure the required scaling. During the transmission of model gradients, a key-generating authority generates a public–private/secret key pair (pk_r^{ij}, sk_r^{ij}) for each client C_{ij} at the beginning of each round r. Once the keys are received, each client encrypts its local gradients G_r^{ij} using HE. This is represented as:

$$[[G_r^{ij}]] \leftarrow encrypt(pk_r^{ij}, G_r^{ij}) \quad \text{(II)}$$

After encryption, every client C_{ij} transmits the encrypted gradients $[[G_r^{ij}]]$ (here $[[]]$ represents encrypted values) to its corresponding local server L_i. Upon receiving the encrypted model updates from all clients within the cluster, the local server L_i performs homomorphic aggregation. As HE supports only addition and multiplication, we use homomorphic addition to find the aggregate sum and then multiply the aggregate sum by the reciprocal of the number of clients in each cluster. This can be represented as:

$$[[G_r^i]] \leftarrow \frac{1}{|p|} \sum_{j \in p} [[G_r^{ij}]] \quad \text{(III)}$$

where, $|p|$ represents the cluster size. Following this, each local server L_i transmits aggregated encrypted model parameters $[[G_r^i]]$ from their respective clusters to the central server S. The central server further performs global aggregation on received model parameters. This can be represented as:

$$[[G_r]] \leftarrow \frac{1}{|n|} \sum_{i \in n} [[G_r^i]] \quad \text{(IV)}$$

where $|n|$ represents the number of participating local clients. The central server S then sends the aggregated model parameters back to the local servers, which are responsible for disseminating these parameters to the clients in their respective clusters. Before proceeding to the next round of training, each client C_{ij} decrypts the received aggregated encrypted model parameters using the private key sk_r^{ij} for that round. This decryption process can be represented as:

$$G_r \leftarrow decrypt(sk_r^{ij}, [[G_r]]) \quad \text{(V)}$$

Each client C_{ij} then proceeds to train its local model on another batch of data b_{r+1}^{ij}, which may either be static or dynamically collected. This training is performed on the

updated model parameters G_r, generating new model parameters for round $r + 1$, represented as:

$$G_{r+1}^{ij} \leftarrow train(G_r, b_{r+1}^{ij}) \quad \text{(VI)}$$

The hierarchical communication process is repeated using HE for each subsequent round. This process continues until either the final round R is reached or the model converges. The workflow of our proposed framework is diagrammatically represented in Figure 3, and the algorithm is as follows:

Algorithm: 1 Hierarchical federated learning with homomorphic encryption

Require: Initial global model W_0, rounds R, round count $r = 0$, local servers n, clients C_{ij}

Ensure: Final global model W^*

1: Central server S broadcasts W_0 to local servers L_i, where $1 \le i \le n$;
2: **for** $r = 1$ to R **do**
3: Each L_i distributes W_o to clients C_{ij};
4: Key authority shares encryption keys $(pk^{ij}{}_r, sk^{ij}{}_r)$;
5: **for** each client C_{ij} **do**
6: Each client decrypts $[[G_r]]$ as G_r, where $r \le 2 \le R$;
7: Train local model on G_r and $b_r^{ij} \subseteq D_{ij}$ to get G_r^{ij};
8: Encrypt G_r^{ij} and send $[[G_r^{ij}]]$ to L_i;
9: Each L_i aggregates collected $[[G_r^{ij}]]$ as and $[[G_r^i]]$ sends to S;
10: S performs global aggregation to obtain $[[G_r]]$ and updates W_{r+1};
11: S sends W_{r+1} to local servers for client distribution;
12: S broadcasts final model W^* through local servers.

2.2. Secure and reliable network architecture

We discussed the possible security challenges in the FL network by addressing both malicious and unintended disruptions caused by clients or communication vulnerabilities, for example, Byzantine attacks, gradient inversion, and packet sniffing. Solutions such as ByITFL,[14] trust bootstrapping via root datasets (FLTrust),[20] and client similarity analysis (e.g., FoolsGold)[21] ensure secure updates through trust and information-theoretic privacy scores. In addition, a cryptographic framework that blends ByITFL[14,22] and FedML-HE[23] to address the dual challenges of adversarial client behavior and privacy leakage in medical FL deployments is provided.

2.3. Threat model and security requirements

In FL, systems are susceptible to a range of attacks against security and privacy that compromise model performance and data privacy. The most critical of these issues occurs when malicious clients under an attacker's control submit poisoned updates, such as label-flipping or trimmed mean methods, to induce global model divergence. Experiments have shown that only 10% of such malicious clients may reduce model accuracy by up to 33.1%,[24] which highlights the need for robustness against such attacks.

Another urgent concern is privacy leakage. Even if raw data is retained locally, leakage of gradient distributions might compromise sensitive information. Attacks on medical imaging data, for example, have resulted in up to 74.24% reconstruction success rates for all data samples in the medical segmentation decathlon liver dataset from model updates[25] (for privacy parameter, $\varepsilon = 20$). Therefore, robust privacy-preserving mechanisms are essential.

A deployable FL system in healthcare should be robust, i.e., capable of operating in unreliable network environments, especially in wireless configurations with as much as packet loss rates (approximately 10%) for maximum users (approximately 90%).[26,14] These interruptions cause stale or missing updates, leading to unwanted divergence during model training.

To address the above challenges of privacy and security, a recent article by Xia *et al.*[14] introduces ByITFL, an FL-based privacy-preserving secure aggregation scheme. Here, Byzantine basically refers to clients that are malicious or behave arbitrarily, typically with the intention of poisoning training. ByITFL is designed to maintain the integrity and confidentiality of model updates during aggregation, without relying on computational assumptions. The scheme is information-theoretically private, i.e., even a semi-honest or curious server learns nothing beyond the aggregated gradients. Privacy is quantified using Shannon entropy (H):[22] for a finite field F, the local update entropy of a client X, given on the server's observations, must satisfy

$$H(X|View_{server}) \ge log|F| \quad \text{(VII)}$$

ensuring that individual model inputs are infeasible to reconstruct. Moreover, to be immune to Byzantine attacks, ByITFL incorporates robustness requirements from distributed consensus theory. Specifically, the system can tolerate up to t malicious clients only if the total number of clients N satisfies the inequality:[14,27]

$$N \ge 3t + 1 \quad \text{(VIII)}$$

This barrier ensures that there are sufficient correct clients to regulate the aggregation process so that the system can reject erroneous updates and achieve stable convergence.

2.4. ByITFL cryptographic framework

The ByITFL framework integrates three cryptography primitives to achieve Byzantine resilience and

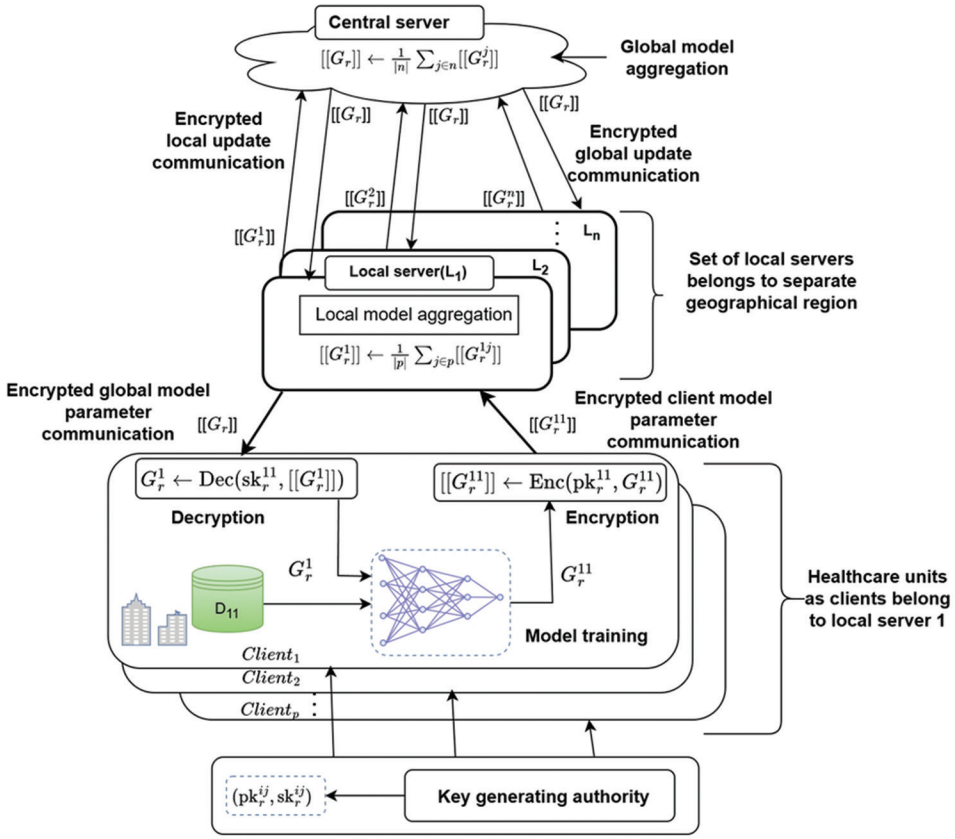

Figure 3. Workflow of the proposed hierarchical federated learning (See Section 2.1 for details)

information-theoretic privacy. Lagrange-coded computing (LCC)[27] offers robust aggregation in the presence of hostile clients by spreading gradient computation over participants. For clients tolerating up to malicious actors, LCC encodes gradients as:

$$\overline{g}_i = \sum_{k=1}^{K} g_k \prod_{1\le m\le K} \frac{\alpha_i - \beta_m}{\beta_k - \beta_m}, m \neq k \qquad \text{(IX)}$$

where α_i is a client-specific evaluation point, β_m is the Lagrange interpolation point for client m, β_k is the interpolation node ensuring redundancy, g_k is the raw gradient of the k^{th} client, and K is the number of honest clients. This ensures proper aggregation if $N \geq 3t + 1$. Verifiable secret sharing employs Pedersen commitments:

$$C\left(\overline{g}_i\right) = g^{\overline{g}_i} h^{ra_i} \ mod\ P\ (g, h \in \mathbb{G}, ra_i \leftarrow \mathbb{Z}) \qquad \text{(X)}$$

Where $C\left(\overline{g}_i\right)$ stands for the adherence to the encoded gradient $\overline{g}_i$, P stands for a large prime number, g and h are the generators of a cyclic group $\mathbb{G}$, and ra_i is a random number from the set of integers Z.[14,22] This enables zero-knowledge proof of gradient consistency without exposing raw updates. To prevent gradient inversion attacks, a second-degree polynomial ($\overline{ReLU}$) approximation substitutes the non-differentiable trust scoring with:

$$\overline{ReLU}(x) = \frac{1}{4}\left(x^2 + x\right)\left(Degree\ 2\ approximation\right) \qquad \text{(XI)}$$

where x is the cosine similarity input. This ensures ϵ-differential privacy (ϵ = 0.25) during trust score computation.[14,22]

2.5. ByITFL aggregation protocol

The protocol is implemented in three phases. In trust initialization, the server has a root dataset D (e.g., 50 – 100 reference samples) to obtain a reference gradient:

$$g_0 = \frac{1}{|D_{root}|}\sum_{v_j \in D_{root}} \nabla_\theta \mathcal{L}oss\left(f_\theta\left(v_j\right), y_j\right) \qquad \text{(XII)}$$

Where v_j and y_j are input-label pairs, θ denotes model parameters, and *Loss* is the loss function. ∇_θ is a gradient operator. It computes the partial derivatives of the loss function with respect to all the parameters in θ, indicating how the model parameters influence the loss. Secure trust

scoring evaluates client updates with a polynomial $\overline{ReLU}$ function:

$$TS_i = \overline{ReLU}\left(\frac{\langle g_0, \bar{g}_i \rangle}{\| g_0 \| . \| \bar{g}_i \|}\right) \quad \text{(XIII)}$$

Where TS_i is the trust score for client i, g_0 is the reference gradient, and $\bar{g}_i$ is the encoded gradient.[22] Here, the dot product $\langle g_0, \bar{g}_i \rangle$ gives the scalar projection, measuring how directionally aligned the two gradient vectors are, while the Euclidian norm in the denominator, which measures the magnitude of the vectors g_0 and $\bar{g}_i$, normalizes this measure so that it lies in (−1, 1). The updates with $TS_i < 0.1$ can be discarded. For LCC-based aggregation, global gradients are computed as:

$$g_{global} = \frac{\sum_{i=1}^{N} TS_i . \bar{g}_i}{\sum_{i=1}^{N} TS_i} + \eta \ (\eta \sim \mathbb{N}\left(0, 0.1 \left\| g_{global} \right\|^2\right) \quad \text{(XIV)}$$

where η is the re-randomization noise drawn from a zero-mean Gaussian distribution $\mathbb{N}$ with a standard deviation equal to 10% of the global gradient's ℓ_2 norm. Such noise interferes with deterministic patterns across training iterations, making gradient memorization or reverse engineering less likely without compromising update fidelity.[14,27]

2.6. Integration of FedML–HE with ByITFL for enhanced privacy

The combination of FedML–HE's selective HE[23] and ByITFL's Byzantine-resilient architecture[28] offers an optimally balanced solution to FL's dual challenges of privacy and security, providing a secure and reliable network architecture. By using FedML–HE's parameter-level encryption on sensitive gradients that are detected through ByITFL's trust scoring,[28] the hybrid solution is able to effectively protect important model updates against inversion attacks[23] while maintaining Byzantine resilience. Empirical evaluations have demonstrated that selective encryption can cut down on communication overhead by up to 10 times for large models such as ResNet-50[23] without a loss in malicious client detection accuracy. Earlier work on Byzantine-resilient secure aggregation architectures points to the possibility of combining cryptographic privacy with adversarial robustness. In addition, FedML-HE's performance-optimized encryption pipeline is optimally compatible with ByITFL's computationally limited architecture. The hybrid solution helps prevent privacy leakage risks in trust-based aggregation while maintaining the system's ability to filter poisoned updates, a circumstance verified by cross-institutional medical FL trials.[23]

2.7. Deep-learning protocol

In the previous section, the FL architecture was outlined. However, specific deep-learning models that might be integrated to address future pandemics were not discussed. As future pandemics are uncertain, to illustrate the point, COVID-19 and lung cancer were taken as benchmarks for potential adaptations of deep-learning protocols.

In this context, the study by Gogineni *et al.*[29] was chosen as a reference. The study investigated the potential of deep-learning models for automated COVID-19 detection using chest X-ray images, presenting a promising alternative to the current gold standard – reverse-transcription polymerase chain reaction (RT-PCR) test. This choice has two distinct merits. First, COVID-19 represents the most recent pandemic for benchmarking. Second, choosing images as input data type. This choice is crucial as image data is representative of various real-world medical datasets and therefore can be used as a reliable proxy (since the exact nature of future pandemics is unknown). Moreover, images are one of the most complex and prevalent data types in the medical arena; therefore, demonstrating with proxy images offers one of the most effective benchmarking strategies to mimic real-world complexities. Although videos represent more complex data types, they are far less frequent in the medical context and, in a crude sense, can be considered as a sequential stacking of images with time-series data (audios, if any) added in an additional channel.

In the study by Gogineni *et al.*,[29] several CNN architectures were implemented, including ResNet34,[30] SeResNext50,[31,32] DenseNet121,[33] and EfficientNet.[34] These models were chosen for their distinct advantages in image classification tasks. ResNet34 utilizes skip connections, allowing for efficient training of deep networks, while SeResNext50 incorporates squeeze-and-excitation blocks, which recalibrate channel-wise feature responses for improved representational capacity. Meanwhile, DenseNet121, with its dense connections between layers, facilitates feature reuse and enhances information flow. Finally, EfficientNet models are designed using neural architecture search, optimizing the balance between accuracy and computational efficiency.[34] Transfer learning, using the ImageNet dataset,[35] was employed to improve model performance on the relatively limited medical image dataset. A learning rate scheduler[36] and a one-cycle training policy were also implemented for better convergence and generalization.

The models' performance demonstrated encouraging results.[29] ResNet34 and DenseNet121 achieved the highest overall accuracy of 94.09% in classifying images as COVID-19, normal, or pneumonia. This accuracy is considerably higher than the typical 70 – 80% sensitivity

reported for RT-PCR tests. A more detailed analysis revealed that EfficientNet exhibited the highest specificity (95.4%) for COVID-19, while ResNet34 showed the highest sensitivity (94%). Interestingly, EfficientNet showed the highest performance in classifying normal cases, SeResNext50 excelled in classifying pneumonia cases, and ResNet34 was most effective for COVID-19 cases. These results are comparable to, and in some cases outperform, those reported in earlier studies using CNNs for COVID-19 detection from chest X-rays and computed tomography (CT) scans.[37-39] The present deep-learning approach offers advantages in terms of speed, resource efficiency, and potentially greater accuracy compared to RT-PCR. The approach is particularly appealing for resource-limited settings where rapid screening is critical. Moreover, the decoupled workflow, separating image acquisition and diagnostic evaluation, offers greater operational flexibility and scalability, aligning with the principles outlined in the previous study[29] for building a robust pandemic response.

A previous study[40] explored a novel approach for lung cancer detection, combining an unsupervised clustering algorithm for ROI proposals with CNNs.[41,42] The modularity optimization-based graph clustering method[41,42] applied to preprocessed CT scans from the 2016 lung nodule analysis dataset[43,44] reduces CNN complexity by identifying potential nodule locations. This preprocessing step includes lung segmentation using marker-controlled "watershed segmentation" on Sobel-filtered images, focusing the analysis on relevant areas and reducing the computational burden. The segmented lung regions are then converted into a network, with pixels representing nodes and edges connecting neighboring pixels. This network is then clustered using a modularity function optimized for spatially embedded networks,[41,42] which has been successfully used previously in analyzing granular assemblies.[43] This method effectively identifies nodules based on grayscale-intensity similarity, generating ROI proposals for subsequent CNN analysis.

The proposed method utilizes these ROI proposals to streamline the CNN classification process. Rather than relying on computationally expensive selective search[43] or fully labeled datasets required for methods like MaskRCNN,[45] this approach employs the clustering algorithm to generate a manageable number of fixed-size ROIs. These ROIs are then fed into a CNN, trained using transfer learning initialized with ImageNet weights,[35] and optimized using learning rate scheduler techniques.[36] This training strategy, employing discriminative learning rates for different layers of the network, facilitates efficient learning of both general and dataset-specific features, which are of great value for the FL learning protocols discussed in this article. Several pre-trained CNN architectures, including ResNet50,[30] SeResNext50,[31,32] and DenseNet161,[33] were evaluated, with DenseNet161 demonstrating the optimal performance. The CNN classifies each ROI as cancerous or non-cancerous, providing not only diagnostic information but also, combined with the clustering results, an approximate segmentation mask. This approach bypasses the need for computationally intensive semantic segmentation techniques, such as GrabCut[46] or context-aware masks,[47] offering a more efficient method for identifying and localizing cancerous nodules within the lung.

2.8. Public data authentication and blockchain

The previous section discussed the need to ensure a smooth communication protocol between the central server, local servers, and data-owning entities, which necessitates entering into a legal contract between the parties. It is important to note that, ideally, the central server should be owned either directly by the United Nations or any of its subsidiaries, such as the WHO. Due to the lack of specific sovereign oversight over the entire protocol, creating digital trust is imperative. Fortunately, a robust technical solution is available to build trust in the digital space between unknown agencies through a decentralized protocol called blockchain. Blockchain-based smart contracts offer an attractive method that creates a transparent technological framework governing the relations between the parties participating in the FL architecture. However, such smart contracts are not free from threats, and in the following, a few novel methods to ensure data safety in smart contracts are discussed.

Detecting vulnerabilities in smart contracts is crucial due to their immutable nature and the potential for significant financial losses, as evidenced by past incidents, including the decentralized autonomous organization[48] hack and the Parity wallet freeze.[49] Gogineni *et al.*[50] constructed a multiclass classifier to detect vulnerabilities in smart contracts by fine-tuning an AWD-LSTM model.[51] The model was fine-tuned using a dataset of 40,877 unique opcode combinations from smart contracts. The smart contracts were classified into four categories: suicidal, prodigal, greedy, and normal. To address the class imbalance, only distinct opcode combinations were retained for normal smart contracts, as they comprised the majority of the dataset and often contained repeated sequences.

The AWD-LSTM model architecture combined a pretrained encoder with a custom classification head, inspired by the ULMFiT method[52] used in natural language processing. This method achieved an overall accuracy of 91.3% and a weighted average F1 score of

90.0% in classifying smart contract vulnerabilities. The model's performance was evaluated using various metrics, including precision, recall, F1 score, confusion matrix, and receiver operating characteristic curves, demonstrating its effectiveness in detecting vulnerabilities compared to random guessing.

3. Results

In this section, the cost-effectiveness and performance of the proposed hierarchical FL framework were theoretically and experimentally analyzed. The analyses considered both communication costs and computation costs for all entities involved in the training process: Clients, local servers, and the central server. To demonstrate the utility of the proposed framework, the observed accuracy was compared with the benchmark FL algorithm, FedAvg,[53] with fine-tuning.

3.1. Experimental evaluation

The experiments were conducted using the structure of Flower Framework,[54] extended to support both a standard FedAvg setup and the proposed hierarchical FL framework. Python 3.10.12 (Phyton software foundation, US) was used in a Jupyter Notebook (v7.4.3, Jupyter, US) environment, along with TensorFlow, Pandas, NumPy, and scikit-learn libraries. The TenSEAL library[55] was utilized to integrate HE computation, supporting encrypted tensor operations using the Cheon-Kim-Kim-Song[56] scheme from Microsoft's simple encrypted arithmetic library (SEAL). This simulation setup emulates the interactions between clients and the server in a real-world scenario while maintaining a controlled environment for reproducibility.

The modified National Institute of Standards and Technology (MNIST) image dataset[57] was employed for experiments, as image data is representative of numerous real-world medical datasets and can therefore be used as a reliable data proxy. MNIST consists of 70,000 grayscale images, with 60,000 images used for training and 10,000 for testing. Each image is 28 × 28 pixels and encoded with intensity values ranging from 0 to 255.

For model training, a CNN architecture was designed with an input layer, two hidden layers activated by *ReLU* functions, and an output layer generating a probability distribution over 10 classes using a softmax function. The model was trained for 30 communication rounds. In the baseline FedAvg setup, the simulation involved 12 and 21 clients and a single central server. For the hierarchical FL implementation, two configurations were tested: first, with four local servers, each connected to three client nodes (totaling 12 nodes, matching the first non-hierarchical case), and one central server coordinating the aggregation; second, with three local servers and seven clients per server (totaling 21 nodes, corresponding to the second non-hierarchical case), and a central server. The learning rate was set to 0.01, and each client performed three local training epochs per round. Throughout the training process, both training time and accuracy at each round were monitored to compare model performance. Figure 4a and 4b illustrate the accuracy and training time per round comparison, respectively.

Model accuracy was observed to increase gradually with each communication round for both algorithms. However, the hierarchical FL framework consistently demonstrated higher accuracy due to its intermediate cluster-level aggregation, which is less biased to outlier client data. The proposed framework achieved an accuracy of 94.23%, whereas the FedAvg approach reached 92.66% under the same experimental settings. Training time per round showed minor fluctuations for both algorithms, depending on the data distribution within each training

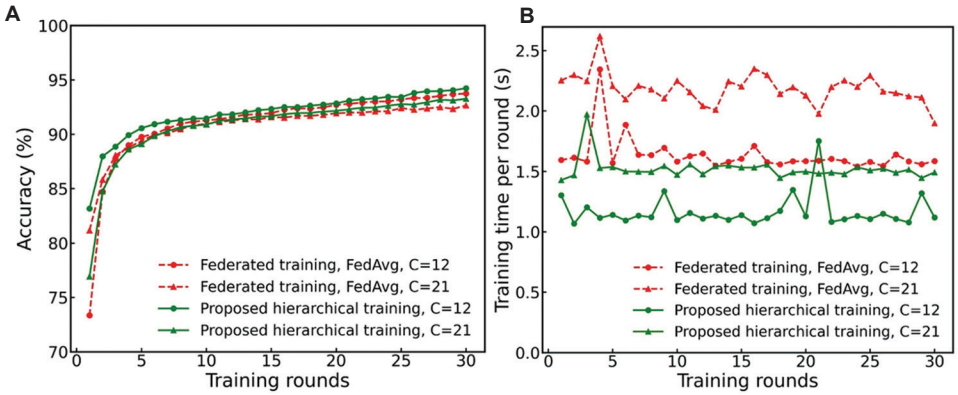

Figure 4. Comparisons between FedAvg and the proposed hierarchical FL framework. (A) Accuracy comparison. Note that the hierarchical FL demonstrates enhanced security compared to FedAvg. (B) Training time per round comparison. Note that the hierarchical FL consumes lesser compute time, demonstrating a higher efficiency than FedAvg.

batch. Nevertheless, the proposed framework exhibited lower training time across all scenarios. This improvement is attributed to the hierarchical structure: whereas FedAvg requires all clients to transmit model updates to the central server concurrently, the hierarchical approach organizes clients into clusters. Each client communicates only with their respective local servers, and the aggregated updates are subsequently forwarded to the central server, thereby reducing communication overhead. Given its time efficiency, the hierarchical model is better suited for global deployment and offers greater scalability compared to FedAvg, particularly in addressing future pandemics. As the hierarchical model outperforms standard FL protocols with complex data types, such as images, it is anticipated to perform effectively with other data types as well, including categorical, numerical, and time-series data, such as cancer classifications, biophysical parameters, and electrocardiography, respectively.

3.2. Communication cost analysis

In the proposed framework, communication occurs at two levels: from client to local server and from local server to central server. All communication is bidirectional, meaning both entities exchange messages during each round. The communication cost of the aforementioned level was analyzed.

3.2.1. Client-to-local server communication

Each client C_{ij} communicated with its assigned local server L_i by transmitting encrypted model gradients $[[G^{ij}_r]]$ and its associated public key pk_r^{ij} during each round r. The communication cost per client is represented using big-O notation:

$$Cost_{\text{client-to-local}} = O(|G^{ij}_r| + |pk_r^{ij}|) \quad \text{(XV)}$$

where $|G^{ij}_r|$ is the size of the encrypted gradient and $|pk_r^{ij}|$ is the size of the encryption key for p clients under a single local server. The parameter p was included in the calculation, as it was assumed to represent the maximum number of clients under a local server. The total communication cost is represented as:

$$Cost_{\text{total-client-to-local}} = O(p\cdot(|G^{ij}_r| + |pk_r^{ij}|)) \quad \text{(XVI)}$$

3.2.2. Local server-to-central server communication

Each local server L_i aggregated encrypted gradients from its clients and transmitted the aggregated model updates $[[G^i_r]]$ to the central server. The communication cost per local server is represented as:

$$Cost_{\text{local-to-central}} = O(|G^i_r| + |pk_r^i|) \quad \text{(XVII)}$$

where $|G^i_r|$ is the size of the aggregated gradient and $|pk_r^i|$ is the size of the encryption key for the local server. For n local servers, the total communication cost is represented as:

$$Cost_{\text{total-local-to-central}} = O(n\cdot(|G^i_r| + |pk_r^i|)) \quad \text{(XVIII)}$$

3.2.3. Total communication cost

Combining the costs for all client-to-local server and local-to-central server communications, the overall communication cost for the framework is represented as:

$$Cost_{\text{total}} = (p.\ |G^{ij}_r| + n.\ |G^i_r| + p.\ |pk_r^{ij}| + n.\ |pk_r^i|) \quad \text{(XIX)}$$

The parameter size at local server L_i, denoted as $|G^i_r|$, includes the presence of p clients, i.e., $|G^i_r| = p\cdot|G^{ij}_r|$, where $|G^{ij}_r|$ represents the parameter size from an individual client j. Therefore, to reflect the total cost across n local servers, $n\cdot|G^i_r|$ is used instead of $n\cdot p\cdot|G^{ij}_r|$.

3.3. Computation cost analysis

The computation cost of the proposed framework was analyzed by evaluating the operations performed at each level: clients, local servers, and the central server. This section provides a detailed computation cost analysis.

3.3.1. Client-side computation

Each client C_{ij} performed local model training on its secured data. The training cost was proportional to the local dataset D_{ij} and the complexity of the model M. Then, it encrypted the model gradients G^{ij}_r using the encryption key pk_r^{ij}. Similarly, it also decrypted the aggregated gradients received from the central server at the end of each round. The computation cost per client is represented as:

$$Cost_{\text{client}} = O(|D_{ij}|\cdot M + Enc(|G_r^{ij}|) + Dec(|G_r^{ij}|)) \quad \text{(XX)}$$

where $Enc(|G^{ij}_r|)$ is the encryption cost and $Dec(|G^{ij}_r|)$ is the decryption cost for the gradients. Suppose p clients were present under a local server; the total client-side computation cost is represented as:

$$Cost_{\text{total-client}} = O(p\cdot(|D_{ij}|\cdot M + Enc(|G_r^{ij}|) + Dec(|G_r^{ij}|))) \quad \text{(XXI)}$$

3.3.2. Local server-side computation

Each local server L_i performed the aggregation of encrypted gradients received from all clients in the cluster using homomorphic addition and multiplication, ensuring that the aggregation was performed without decrypting the gradients. If the number of clients assigned to L_i was p, then p additions and a single multiplication were required for aggregation. For simplicity, the time taken for multiplication was assumed as unity. The computation cost per local server is represented as:

$$Cost_{\text{local-server}} = O(p\cdot T_{add}) \quad \text{(XXII)}$$

where T_{add} represents the time taken for homomorphic addition. For n local servers in the framework, the total local server-side computation cost is represented as:

$$Cost_{\text{total-local-server}} = O(n \cdot p \cdot T_{add}) \quad \text{(XXIII)}$$

3.3.3. Central server-side computation

Similar to local servers, the central server performed aggregation of gradients received from n local servers. The computation cost at the central server is represented as:

$$Cost_{\text{total-local-server}} = O(n \cdot T_{add})$$

3.3.4. Total computation cost

Combining the computation costs at the client, local server, and central server levels, the total computation cost for the proposed framework is represented as:

$$Cost_{\text{total}} = O(p \cdot (|D_{ij}| \cdot M + Enc(|G_r^{ij}|) + Dec(|G_r^{ij}|)) + n \cdot p \cdot T_{add} + n \cdot T_{add}) \quad \text{(XXIV)}$$

It can be further simplified to:

$$Cost_{\text{total}} = O(p \cdot (|D_{ij}| \cdot M + Enc(|G_r^{ij}|) + Dec(|G_r^{ij}|) + n \cdot T_{add})) \quad \text{(XXV)}$$

The hierarchical structure optimized both communication and computation by leveraging local servers to consolidate updates before transmitting them to the central server. This ensured scalability, even in scenarios with large datasets and numerous participants.

4. Discussion

In this section, the issue of data heterogeneity in FL is addressed, specifically in the context of datasets distributed across different medical centers or countries. FL encounters significant challenges in real-world medical settings owing to the intrinsic heterogeneity among contributing institutions. Data heterogeneity arises when data distribution varies substantially across clients, leading to non-IID data.[58] Heterogeneity may manifest as statistical variations, differences in system capabilities, disparities in model architecture, and additional challenges.[59,60] While the proposed model can directly handle IID datasets, its robustness is demonstrated by showing how it can manage non-IID datasets. Several techniques are proposed to mitigate data imbalance effects and enhance model performance in hierarchical systems designed for managing medical data management.

According to recent studies, and within the context of hierarchical medical data management, generative adversarial networks (GANs), particularly the newly developed robust diffusion models,[61] can effectively achieve uniformity in data availability across medical facilities. Zadeh *et al.*[62] utilized GANs for cross-modality brain image synthesis, including transformations such as CT to positron emission tomography (PET), CT to magnetic resonance imaging (MRI), MRI to PET, and vice versa.

To address statistical heterogeneity or non-uniform data distribution, which significantly impacts model accuracy, increased communication rounds are often necessary. However, this can introduce bias in the global model, particularly disadvantaging clients with underrepresented data from various institutions. Therefore, aligning the distributions of data across medical centers is critical to mitigating model bias caused by variations in the population of patients or data collection techniques.[58,63] Class balancing[64] should be supported with equal representation of all disease classes or conditions across federated nodes to prevent biased learning outcomes. Additionally, standardization of quality[65] is necessary to normalize data collected via varying equipment and protocols to enhance uniformity and reliability. Moreover, volume balancing[66] helps prevent dominant contributions from larger hospitals, ensuring equitable learning from all centers.

To fulfill these requirements, GANs, especially robust diffusion models, offer a promising method for establishing data uniformity across hospitals with varied dataset sizes. By generating synthetic images to supplement existing datasets, GANs enable more balanced training with minimal bias. For example, if three medical centers have 500, 400, and 250 data points, respectively, GANs can generate synthetic images to equalize each dataset to approximately 500 data points. Compared to traditional weighted averaging of model parameters, this approach provides a more balanced solution for hierarchical medical system performance. The working principle is based on iterative noise addition and removal, where the generator network analyzes the denoising function to reconstruct the original image.

Despite the benefits of data augmentation via GANs, FL in medical imaging still encounters challenges due to the inherent diversity of imaging data. Scans from different sites vary in scanner type, protocol, and patient demographic, making synthetic data approaches more complex.[67] Recent FL frameworks, such as distributed synthetic learning, aim to train GANs to produce a single homogeneous dataset of synthetic images for use by all clients,[67] yet practical concerns remain. Specifically, the application of differential privacy can hamper performance. For example, Kossen *et al.*[68] reported that enforcing a privacy parameter $\varepsilon \approx 7.4$ on GAN-produced angiograms lowered a U-net vessel segmentation's dice score from 0.84 to 0.75.

In addition, GAN-augmented FL models are susceptible to membership inference attacks (MIAs). MIAs allow attackers to deduce whether a particular data point belongs to the training set. For example, Zhang *et al.*[69] demonstrated class-level and user-level MIAs with GANs, achieving over

90% attack accuracy in FL environments. To mitigate these attacks, a membership inference defense mechanism named DefMIA was introduced, which adds adversarial perturbations to global model parameters, reducing attack accuracy to approximately 50% without affecting model performance. Overall, GAN-based FL remains susceptible to accuracy degradation and privacy concerns when applied to non-IID medical data, necessitating critical evaluation and ongoing refinement.

Model heterogeneity refers to differences across model architectures, capacities, and training approaches across medical centers. Before FL implementation, architectural compatibility must be ensured to facilitate effective knowledge transfer across diverse model structures without compromising performance. Scale adaptability should also be enabled in the platform to accommodate the varying computational capabilities of institutions by enabling flexible model sizes. In addition, knowledge alignment mechanisms are needed to coordinate learning across heterogeneous models. Personalization support should be enabled to address institution-specific requirements or patient groups without compromising the global model's integrity or performance. Therefore, to address model variation, a hierarchical self-distillation approach, such as HierarchyFL,[70] is recommended. This method facilitates architecture compatibility by enabling resource-limited centers to efficiently learn from advanced models deployed at major sites. However, conventional aggregation techniques, such as FedAvg, cannot be directly applied in this setting. In addition, model architecture inconsistencies reduce knowledge transfer, as some clients may be unable to contribute weights, leading to performance degradation and inconsistency.

System heterogeneity refers to differences in computational resources, storage capacity, and communication bandwidth. Strategies to manage these differences are essential. Resource-conscious participation should be prioritized, with computational overhead adapted to each institution's hardware and network capacity. Efficiency in communication is also required for optimal data transfer, particularly within low or unstable bandwidth settings. Furthermore, storage efficiency should accommodate institutions with limited infrastructure to ensure broad participation. Power efficiency is also important, especially for edge devices, to reduce power consumption with consistent performance within clinical settings. A framework incorporating client selection through reinforcement learning and resource-aware metrics, such as that proposed in personalized FL,[71] is recommended. This framework performs federation tasks dynamically according to each center's available resources, supporting equitable participation. Nonetheless, system heterogeneity continues to extend training time due to hardware capabilities and network disparities, increase client dropout rates, and introduce latency, all of which hinder overall FL performance. These challenges warrant further research.

5. Conclusion

In this article, a hierarchical FL framework is proposed, designed to enhance scalability, security, and computational efficiency through multilevel aggregation and HE, enabling effective responses to future pandemics. FL is a decentralized ML paradigm in which each client trains a model on local data and shares only model updates with a central server, thus preserving data privacy. The hierarchical structure, consisting of client nodes, local servers, and a central server, supports distributed learning and efficient communication. Local servers perform intermediate aggregation, reducing communication costs and enhancing scalability. The proposed framework supports dynamic data and can be extended to train specialized models at the central unit based on data characteristics. It ensures secure model updates through parameter encryption and seamless integration with ByITFL, which incorporates Byzantine fault tolerance and information-theoretic privacy protection, strengthening the integrity and confidentiality of updates during aggregation.

To address the demands of medical image-based diagnostics, the framework is adaptable to advanced CNN architectures, such as ResNet34, DenseNet121, and EfficientNet, which have demonstrated high accuracy in tasks such as COVID-19 detection. Similarly, in lung nodule detection, the incorporation of modular clustering and transfer learning enhances imaging operations by efficiently identifying ROI, reducing resource usage while improving diagnostic accuracy. On the other hand, blockchain-based smart contracts facilitate trusted coordination among participants, while GANs contribute to standardized data processing. Experiments conducted using the Flower framework and TenSEAL on the MNIST dataset demonstrated improved model accuracy, reduced training time per round, and lower communication overhead due to the scope of optimized intermediate aggregation.

Overall, the proposed framework offers a scalable, privacy-preserving, and computation-efficient solution for digital healthcare. These advancements collectively position the proposed framework as a robust approach for addressing future healthcare challenges and responding effectively to potential pandemics.

Acknowledgments

None.

Funding

None.

Conflict of interest

The authors declare that they have no competing interests.

Author contributions

Conceptualization: Kisor Kumar Sahu
Formal analysis: All authors
Investigation: Rojalini Tripathy
Methodology: All authors
Visualization: All authors
Writing – original draft: All authors
Writing – review & editing: All authors

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data

Data are available at the following resource: Deng, L. The MNIST database of handwritten digit images for machine learning research. IEEE signal processing magazine;2012. 29(6), 141-142. doi: 10.1109/MSP.2012.2211477 and code can be shared upon reasonable request over email: kisorsahu@iitbbs.ac.in.

References

1. World Health Organization. *COVID-19 Deaths. WHO COVID-19 Dashboard*. World Health Organization Data; 2023. Available from: https://data.who.int/dashboards/covid19/deathss [Last accessed on 2024 Dec 14].
2. Roser M. *The Spanish Flu: The Global Impact of the Largest Influenza Pandemic in History*. Our World in Data; 2020. Available from: https://ourworldindata.org/spanish/flu-largest-influenza-pandemic-in-history [Last accessed on 2024 Dec 14].
3. Mordechai L, Eisenberg M, Newfield TP, Izdebski A, Kay JE, Poinar H. The justinianic plague: An inconsequential pandemic. *Proc Natl Acad Sci U S A*. 2019;116:25546-25554.

   doi: 10.1073/pnas.1903797116
4. Wade L. *From Black Death to Fatal Flu, Past Pandemics Show why People on the Margins Suffer Most*. Science Magazine; 2020. Available from: https://www.science.org/content/article/black-death-fatal-flu-past-pandemics-show-why-people-margins-suffer-most [Last accessed on 2024 Dec 14].

   doi: 10.1126/science.abc7832
5. Pathak AD, Saran D, Mishra S, Hitesh M, Bathula S, Sahu KK. Smart war on COVID-19 and global pandemics: Integrated AI and blockchain ecosystem. In: *Computational Modeling and Data Analysis in COVID-19 Research*. United States: CRC Press; 2021. p. 67-94.

   doi: 10.1201/9781003137481-5
6. McMahan B, Moore E, Ramage D, Hampson S, Arcas BA. Communication-efficient learning of deep networks from decentralized data. In: *Artificial Intelligence and Statistics*. Hamburg: Statista; 2017. p. 1273-1282.

   doi: 10.48550/arXiv.1602.05629
7. Yurdem B, Kuzlu M, Gullu MK, Catak FO, Tabassum M. Federated learning: Overview, strategies, applications, tools and future directions. *Heliyon*. 2024;10:e38137.

   doi: 10.1016/j.heliyon.2024.e38137
8. Nguyen DC, Pham QV, Pathirana PN, *et al*. Federated learning for smart healthcare: A survey. *ACM Comput Surv*. 2022;55(3):1-37.

   doi: 10.1145/3501296
9. Rieke N, Hancox J, Li W, *et al*. The future of digital health with federated learning. *NPJ Digit Med*. 2020;3(1):119.

   doi: 10.1038/s41746-020-00323-1
10. Sheller MJ, Edwards B, Reina GA, *et al*. Federated learning in medicine: Facilitating multi-institutional collaborations without sharing patient data. *Sci Rep*. 2020;10(1):12598.

    doi: 10.1038/s41598-020-69250-1
11. Sun T, Li D, Wang B. Decentralized federated averaging. *IEEE Trans Pattern Anal Mach Intell*. 2022;45(4):4289-4301.

    doi: 10.48550/arXiv.2104.11375
12. Korzun DG. Internet of things meets mobile health systems in smart spaces: An overview. In: *Internet of Things and Big Data Technologies for Next Generation Healthcare*. Berlin: Springer Nature; 2017. p. 111-129.

    doi: 10.1007/978-3-319-49736-5_6
13. Velichko A, Korzun D, Meigal A. Artificial neural networks for iot-enabled smart applications: Recent trends. *Sensors*. 2023;23(10):4853.

    doi: 10.3390/s23104853
14. Xia Y, Hofmeister C, Egger M, Bitar R. *Byzantine-Resilient Secure Aggregation for Federated Learning without Privacy Compromises*. United States: IEEE; 2024.

    doi: 10.48550/arXiv.2405.08698
15. Stephanie V, Khalil I, Atiquzzaman M, Yi X. Trustworthy privacy-preserving hierarchical ensemble and federated

learning in healthcare 4.0 with blockchain. *IEEE Trans Ind Inform*. 2022;19(7):7936-7945.

doi: 10.1109/TII.2022.3214998

16. Li G, Hu Y, Zhang M, *et al*. FedHiSyn: A hierarchical synchronous federated learning framework for resource and data heterogeneity. In: *Proceedings of the 51st International Conference on Parallel Processing*. United States: Association for Computing Machinery; 2022. p. 1-11.

    doi: 10.1145/3545008.3545065

17. Wang B, Li H, Guo Y, Wang J. PPFLHE: A privacy-preserving federated learning scheme with homomorphic encryption for healthcare data. *Appl Soft Comput*. 2023;146:110677.

    doi: 10.1016/j.asoc.2023.110677

18. Ooi MPL, Sohail S, Huang VG, *et al*. Measurement and applications: Exploring the challenges and opportunities of hierarchical federated learning in sensor applications. *IEEE Instrum Meas Mag*. 2023;26(9):21-31.

    doi: 10.1109/MIM.2023.10328671

19. Tripathy R, Meshram J, Bera P. HalfFedLearn: A secure federated learning with local data partitioning and homomorphic encryption. *Fut Gener Comput Syst*. 2025;171:107858.

    doi: 10.1016/j.future.2025.107858

20. Cao X, Fang M, Liu J, Gong NZ. *Fltrust: Byzantine-Robust Federated Learning Via Trust Bootstrapping*. United States: Cornell University; 2020.

    doi: 10.48550/arXiv.2012.13995

21. Fung C, Yoon CJ, Beschastnikh I. The limitations of federated learning in sybil settings. In: *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*. 2020. p. 301-316.

22. Adilova L, Rosenzweig J, Kamp M. *Information-Theoretic Perspective of Federated Learning*. [Preprint]; 2019.

    doi: 10.48550/arXiv.1911.07652

23. Jin W, Yao Y, Han S, *et al*. *FedML-HE: An Efficient Homomorphic-Encryption-Based Privacy-Preserving Federated Learning System*. United States: Cornell University; 2023.

    doi: 10.48550/arXiv.2303.10837

24. Mozaffari H, Choudhary S, Houmansadr A. Fake or compromised? making sense of malicious clients in federated learning. In: *European Symposium on Research in Computer Security*. 2024. p. 187-207.

    doi: 10.48550/arXiv.2403.06319

25. Ziller A, Mueller TT, Stieger S, *et al*. Reconciling privacy and accuracy in AI for medical imaging. *Nat Mach Intell*. 2024;6(7):764-774.

    doi: 10.1038/s42256-024-00858-y

26. Zhou P, Fang P, Hui P. *Loss Tolerant Federated Learning*. [Preprint]; 2021.

    doi: 10.48550/arXiv.2105.03591

27. Yu Q, Li S, Raviv N, Kalan SMM, Soltanolkotabi M, Avestimehr SA. Lagrange coded computing: Optimal design for resiliency, security, and privacy. In: *22nd International Conference on Artificial Intelligence and Statistics*. [Preprint]; 2019. p. 1215-1225.

    doi: 10.48550/arXiv.1806.00939

28. So J, Güler B, Avestimehr AS. Byzantine-resilient secure federated learning. *IEEE J Select Areas Commun*. 2020;39(7):2168-2181.

    doi: 10.48550/arXiv.2007.11115

29. Gogineni AK, Hitesh M, Jha PK, Sen SS, Das S, Sahu KK. Deep learning on chest X-ray and computed tomography scans for detection of COVID-19 as a part of a network-centric digital health stack for future pandemics. *Artif Intell Health*. 2024;2:29-41.

    doi: 10.36922/aih.2888

30. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. United States: IEEE; 2016. p. 770-778.

    doi: 10.1109/cvpr.2016.90

31. Hu J, Shen L, Albanie S, Sun G, Wu E. *Squeeze-and-Excitation Networks*. United States: Cornell University; 2017.

    doi: 10.48550/arxiv.1709.01507

32. Xie S, Girshick R, Dollár P, Tu Z, He K. Aggregated residual transformations for deep neural networks. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, USA: IEEE; 2017. p. 5987-5995.

    doi: 10.1109/CVPR.2017.634

33. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: *IEEE Conference on Computer Vision and Pattern Recognition*. United States: IEEE; 2017. p. 4700-4708.

    doi: 10.1109/CVPR.2017.243

34. Tan M, Le QV. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*; 2019. Available from: https://arxiv.org/pdf/1905.11946 [Last accessed on 2024 Dec 18].

    doi: 10.48550/arXiv.1905.11946

35. Simon M, Rodner E, Denzler J. *ImageNet Pre-trained Models with Batch Normalization*; 2016. Available from: https://arxiv.org/pdf/1612.01452 [Last accessed on 2024 Dec 18].

    doi: 10.48550/arXiv.1612.01452

36. Smith LN. *Cyclical Learning Rates for Training Neural Networks*. Piscataway: IEEE Xplore; 2017. p. 464-472.

    doi: 10.1109/WACV.2017.58

37. Wang L, Lin ZQ, Wong A. COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. *Sci Rep*. 2020;10(1):19549.

doi: 10.1038/s41598-020-76550-z

38. Wang S, Kang B, Ma J, *et al*. A deep learning algorithm using CT images to screen for corona virus disease (COVID-19). *Eur Radiol*. 2021;31:6096-6104.

doi: 10.1007/s00330-021-07715-1

39. Joaquin AS. *Using Deep Learning to Detect NCOV-19 from X-Ray Images*. Medium; 2020. Available from: https://towardsdatascience.com/using/deep/learning/to/detect/ncov/19/from/x/ray/images/1a89701d1acd [Last accessed on 2024 Dec 18].

40. Gogineni AK, Kishore R, Raj P, Naik S, Sahu KK. Unsupervised Clustering algorithm as region of interest proposals for cancer detection using CNN. In: *Computational Vision and Bio-Inspired Computing*. Coimbatore: ICCVBIC; 2019. p. 1386-1396.

doi: 10.1007/978-3-030-37218-7_146

41. Kishore R, Gogineni AK, Nussinov Z, Sahu KK. A nature inspired modularity function for unsupervised learning involving spatially embedded networks. *Sci Rep*. 2019;9(1):2631.

doi: 10.1038/s41598-019-39180-8

42. Kishore R, Krishnan R, Satpathy M, Nussinov Z, Sahu KK. Abstraction of meso-scale network architecture in granular ensembles using 'big data analytics' tools. *J Phys Commun*. 2018;2(3):031004.

doi: 10.1088/2399-6528/aab386

43. Uijlings JR, Van De Sande KE, Gevers T, Smeulders AW. Selective search for object recognition. *Int J Comput Vision*. 2013;104:154-171.

doi: 10.1007/s11263-013-0620-5

44. Van Ginneken B, Jacobs C. *LUNA16*. Zenodo; 2019. Available from: https://luna16.grand-challenge.org/ [Last accessed on 2024 May 20].

doi: 10.5281/zenodo.2595813

45. He K, Gkioxari G, Dollar P, Girshick R. Mask R-CNN. In: *IEEE International Conference on Computer Vision (ICCV)*. United States: IEEE; 2017. p. 2980-2988.

doi: 10.48550/arXiv.1703.06870

46. Rother C, Kolmogorov V, Blake A. GrabCut interactive foreground extraction using iterated graph cuts. *ACM Trans Graph (TOG)*. 2004;23(3):309-314.

doi: 10.1145/3596711.3596774

47. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. Learning deep features for discriminative localization. In: *IEEE Conference on Computer Vision and Pattern Recognition*. United States: IEEE; 2016. p. 2921-2929.

doi: 10.1109/CVPR.2016.319

48. Scharfman J. Decentralized autonomous organization (DAO) fraud, hacks, and controversies. In: *The Cryptocurrency and Digital Asset Fraud Casebook DeFi, NFTs, DAOs, Meme Coins, and Other Digital Asset Hacks*. Vol. 2. Berlin: Springer; 2024. p. 65-106.

doi: 10.1007/978-3-031-60836-0_3

49. Destefanis G, Marchesi M, Ortu M, Tonelli R, Bracciali A, Hierons R. Smart contracts vulnerabilities: A call for blockchain software engineering? In: *International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*. United States: IEEE; 2018. p. 19-25.

doi: 10.1109/IWBOSE.2018.8327567

50. Gogineni AK, Swayamjyoti S, Sahoo D, Sahu KK, Kishore R. Multi-Class classification of vulnerabilities in Smart Contracts using AWD-LSTM, with pre-trained encoder inspired from natural language processing. *IOP SciNotes*. 2020;1(3):035002.

doi: 10.1088/2633-1357/abcd29

51. Merity S, Keskar NS, Socher R. *Regularizing and Optimizing LSTM Language Models*. United States: Cornell University; 2017.

doi: 10.48550/arXiv.1708.02182

52. Howard J, Ruder S. *Universal Language Model Fine-Tuning for Text Classification*. United States: Harvard University; 2018.

doi: 10.48550/arXiv.1801.06146

53. Collins L, Hassani H, Mokhtari A, Shakkottai S. Fedavg with fine tuning: Local updates lead to representation learning. In: *Advances in Neural Information Processing Systems*. Vol. 35. United States: MIT Press; 2022. p. 10572-10586.

doi: 10.48550/arXiv.2205.13692

54. Beutel DJ, Topal T, Mathur A, *et al*. *Flower: A Friendly Federated Learning Research Framework*. United States: Cornell University; 2020.

doi: 10.48550/arXiv.2007.14390

55. Benaissa A, Retiat B, Cebere B, Belfedhal AE. *Tenseal: A Library for Encrypted Tensor Operations Using Homomorphic Encryption*. United States: Cornell University; 2021.

doi: 10.48550/arXiv.2104.03152

56. Sathishkumar P, Pugalarasan K, Ponnparamaguru C, Vasanthkumar M. Improving healthcare data security using cheon-kim-kim-song (ckks) homomorphic encryption. In: *2024 International Conference on Knowledge Engineering and Communication Systems*. Vol. 1. United States: IEEE; 2024. p. 1-6.

doi: 10.1109/ICKECS61492.2024.10616691

57. Deng L. The mnist database of handwritten digit images for machine learning research. *IEEE Sign Process Mag*. 2012;29(6):141-142.

doi: 10.1109/MSP.2012.2211477

58. Jimenez GM, Solans D, Heikkila M, *et al*. *Non-IID Data in Federated Learning: A Survey with Taxonomy, Metrics, Methods, Frameworks and Future Directions*. United States: Cornell University; 2024.

doi: 10.48550/arXiv.2411.12377

59. Ye M, Fang X, Du B, Yuen PC, Tao D. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Comput Surv*. 2023;56(3):1-44.

doi: 10.48550/arXiv.2307.10616

60. Gao D, Yao X, Yang Q. *A Survey on Heterogeneous Federated Learning*. United States: Cornell University; 2022.

doi: 10.48550/arXiv.2210.04505

61. Müller-Franzes G, Niehues JM, Khader F, *et al*. A multimodal comparison of latent denoising diffusion probabilistic models and generative adversarial networks for medical image synthesis. *Sci Rep*. 2023;13:12098.

doi: 10.1038/s41598-023-39278-0

62. Zadeh FS, Molani S, Orouskhani M, Rezaei M, Shafiei M, Abbasi H. *Generative Adversarial Networks for Brain Images Synthesis: A Review*. United States: Cornell University; 2023.

doi: 10.48550/arXiv.2305.15421

63. Legler T, Hegiste V, Anwar A, Ruskowski M. Addressing heterogeneity in federated learning: challenges and solutions for a shared production environment. *Procedia Comput Sci*. 2025;253:2831-2840.

doi: 10.48550/arXiv.2408.09556

64. Dai Y, Chen Z, Li J, Heinecke S, Sun L, Xu R. Tackling data heterogeneity in federated learning with class prototypes. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 6. Washington, DC: AAAI Press; 2023. p. 7314-7322.

doi: 10.48550/arXiv.2212.02758

65. Babar M, Qureshi B, Koubaa A. Investigating the impact of data heterogeneity on the performance of federated learning algorithm using medical imaging. *PLoS One*. 2024;19(5):e0302539.

doi: 10.1371/journal.pone.0302539

66. Zhao Y, Li M, Lai L, Suda N, Civin D, Chandra V. *Federated Learning with Non-Iid Data*. United States: IEEE; 2018.

doi: 10.48550/arXiv.1806.00582

67. Chang Q, Yan Z, Zhou M, *et al*. Mining multi-center heterogeneous medical data with distributed synthetic learning. *Nat Commun*. 2023;14(1):5510.

doi: 10.1038/s41467-023-40687-y

68. Kossen T, Hirzel MA, Madai VI, *et al*. Toward sharing brain images: Differentially private TOF-MRA images with segmentation labels using generative adversarial networks. *Front Artif Intell*. 2022;5:813842.

doi: 10.3389/frai.2022.813842

69. Jiale ZH, Chengcheng ZH, Xiaobing SU, *et al*. Membership inference attack and defense method in federated learning based on GAN[J]. *J Commun*. 2023;44(5):193-205.

doi: 10.11959/j.issn.1000-436x.2023094

70. Xia J, Zhang Y, Yue Z, Hu M, Wei X, Chen M. *HierarchyFL: Heterogeneous Federated Learning via Hierarchical Self-Distillation*. United States: Cornell University; 2022.

doi: 10.48550/arXiv.2212.02006

71. Yang H, Li J, Hao M, Zhang W, He H, Sangaiah AK. An efficient personalized federated learning approach in heterogeneous environments: A reinforcement learning perspective. *Sci Rep*. 2024;14:28877.

doi: 10.1038/s41598-024-80048-3