Edge Intelligence: Federated Learning-Based Privacy Protection Framework for Smart Healthcare Systems

Mahmuda Akter, *Graduate Student Member, IEEE*, Nour Moustafa, *Senior Member, IEEE*, Timothy Lynar, and Imran Razzak

***Abstract*—Federated learning methods offer secured monitor services and privacy-preserving paradigms to end-users and organisations in the Internet of Things networks such as smart healthcare systems. Federated learning has been coined to safeguard sensitive data, and its global aggregation is often based on a centralised server. This design is vulnerable to malicious attacks and could be breached by privacy attacks such as inference and free-riding, leading to inefficient training models. Besides, uploaded analysing parameters by patients can reveal private information and the threat of direct manipulation by the central server. To address these issues, we present a three-fold Federated Edge Aggregator, the so-called Edge Intelligence, a federated learning-based privacy protection framework for safeguarding Smart Healthcare Systems at the edge against such privacy attacks. We employ an iteration-based Conventional Neural Network (CNN) model and artificial noise functions to balance privacy protection and model performance. A theoretical convergence bound of Edge Intelligence on the trained federated learning model's loss function is also introduced here. We evaluate and compare the proposed framework with the recently established methods using model performance and privacy budget on popular and recent datasets: MNIST, CIFAR10, STL10, and COVID19 chest x-ray. Finally, the proposed framework achieves 90% accuracy and a high privacy rate demonstrating better performance than the baseline technique.**

***Index Terms*—Convolutional neural network (CNN), edge intelligence, federated learning, Internet of Things (IoT), privacy-preserving, smart healthcare system (SHS).**

I. Introduction

THE Internet of Things (IoT) growth enables smart healthcare systems to step forward for inaccessible patients during critical situations like pandemics and ambient assisted living (AAL). In this situation, a Smart Health Systems (SHS) [1] offers smart applications for real-time health monitoring, early-stage detection, and cognitive decision-making by gathering patients' immediate physiological records at any time and from any location via an IoT network. As a result, the demand for IoT systems is rapidly increasing, with an estimated value of $158 billion by 2022. Moreover, the recent COVID-19 outbreak has accelerated this industry's hyper-growth [2]. Sensitive information such as valuable health insights data collected through high technology devices in IoT systems needs to choose an efficient smart sharing platform that meets the high-level privacy of Australian Privacy law with HIPAA [3], [4] compliant.

Currently, federated learning ensures that a user's data stays on their device while applications run particular programs that learn how to handle the data and develop a better and more efficient model [5]. Federated learning is an efficient distributed Machine Learning for privacy preservation for training Machine Learning models. Centralised machine learning in intelligent smart healthcare systems may confront several privacy challenges, such as stealing users' data. Because users often lose control over their data during processing and sharing through centralised machine learning algorithms. Alternatively, federated learning offers a collaborative learning mechanism with machine learning algorithms by processing data at the user-end and sharing the learning model parameters to a central server instead of raw data.

Cooperation and broadcasting require a central monolithic intelligent server that executes a model's changes in a single unit. Federated learning is a healthcare system that can establish an IoT network with smart technology that generates heterogeneous data. Attackers in the central aggregator may plan to launch multiple privacy assaults, affecting the entire system at a single point of failure in its internal components [6]. As a result, bottleneck traffic experiences strong and well-recognised performance problems of data flow due to limited bandwidth to support the volume of data. Besides, all traffic is controlled by the central aggregator. In terms of privacy, if the global aggregator faces privacy breaches, then all connected users cannot access the application. Privacy attacks will affect the entire system and other challenges may arise that might cause single point of failure. Federated learning relies on an integrated entity for establishing devices, selecting members, and estimating the global aggregate in each epoch. These processes constantly bias the results [7], [8].

Manuscript received 8 April 2022; revised 19 June 2022; accepted 15 July 2022. Date of publication 20 July 2022; date of current version 6 December 2022. This work was supported by the University of New South Wales (UNSW), Canberra. *(Corresponding author: Nour Moustafa.)*

Mahmuda Akter, Nour Moustafa, and Timothy Lynar are with the School of Engineering and Information Technology, University of New South Wales, Canberra, ACT 2612, Australia (e-mail: mahmuda.akter@adfa.edu.au; nour.moustafa@unsw.edu.au; t.lynar@adfa.edu.au).

Imran Razzak is with the University of New South Wales, Sydney, NSW 2052, Australia (e-mail: z5131416@unsw.edu.au.).

Digital Object Identifier 10.1109/JBHI.2022.3192648

Edge Intelligence provides well-organised Artificial Intelligence placement at edge servers by leveraging large-scale computation and connectivity capabilities to process data close to the end devices generated [9]. By pre-processing trained models before transferring them directly to smart healthcare service providers, Edge Intelligence based on federated learning supports machine learning in a privacy-preserving manner. Edge intelligence can be improved processing to manage overall system privacy collapse for the next generation of smart healthcare systems. Furthermore, the Federated Edge Intelligence (FEI) architecture [10] enabled machine learning-based smart services and applications to make federated learning more energy-efficient.

Although edge intelligence-based federated learning has various advantages over machine learning-based intelligent applications in IoT systems, establishing such machine learning-driven intelligent applications with a high accuracy score necessitates greater data privacy protection. Datasets exposed to third-party edge servers may result in significant privacy infringement [11]. Furthermore, current privacy standards and regulations limit access to sensitive data in intelligent applications [12]. As a result, offering privacy-preserving access to large databases while assuring increased data analytics is crucial. Most current research focuses on maximising IoT communication resource allocation and distributing computational burden across edge computing [13]. As a result, incorporating edge intelligence into the edge computing layer could help to improve overall resource optimisation.

Anonymisation, encryption, and randomisation methods have been studied to produce privacy-preserving strategies [14]. They are, however, ineffective at preventing the leakage of sensitive data from Smart Health Systems in an IoT network. Differential privacy could take advantage of manufactured noise in this situation, causing each client's information to be perturbed locally and only sharing a randomised version with the aggregator. A Gaussian noise-adding methodology is more efficient for data privacy in shared models in federated learning-based privacy-preserving strategies. It will protect users' data from poisoning, backdoor attacks, and other threats. There have been some ways to handle privacy preservation for large data sources. A central aggregator in federated learning is in charge of gathering, aggregating, and broadcasting a model to many clients [15].

On the other hand, a breached central server could compromise privacy through direct link capability or client identifiability [16]. IoT devices are also not limited to a specific number in real-world data generators. Controlling a model's massive inundation to its central aggregator also produces data traffic and congestion. As a result, loading the distribution and pre-processing of the model is required to improve the middle association's privacy preservation.

Motivational Scenario—Federated learning has been applied to IoT devices with high-functioning designs to reduce communication rounds during model training. Moreover, it outperforms classic distributed learning approaches for IID (Independent and Identically Distributed) and non-IID data distribution methods [17]. Non-IID training data necessitates fewer data transmissions from IoT devices to maintain privacy [18]. Federated learning overcomes some drawbacks of centralised machine learning algorithms by directly processing data on the client-side and communicating parameters with a central server, ensuring data privacy. However, the emergence of new cyber threats against the federated learning paradigm results in various privacy breaches, a significant worry with intelligent embedded systems, particularly the Internet of Things. Because it takes advantage of the widespread presence of connected things in the healthcare system, this mindset has given rise to the concept of eHealth, allowing for new healthcare methods and solutions [19].

Furthermore, the attackers' purpose is to compromise users' confidentiality, availability, and integrity [20]. The insertion of carefully prepared hostile instances into traditional Machine Learning models made possible by acquiring training data from the public domain renders traditional Machine Learning models vulnerable to attackers. It allows the model to readily adjust to the attacker's desire [21]. In addition, the central aggregator might sometimes be honest but curious and wish to use parameterised models for analysis or give access to an unauthorised organisation. The curator might assess the client's nature and track which shared a specific updated model. Suppose all connected clients' data privacy will face a significant threat if privacy attacks violate the central aggregator. A massive amount of IoT data model aggregation and broadcasting from a central server might face significant traffic congestion that will impact overall system performances, including energy time and accuracy.

According to recent studies, federated learning as distributed deep learning (DDL) can provide better privacy protection than centralised deep learning because it only trains on shareable parameters [22]. The resource consumption of a federated learning-based algorithm is closely related to several key parameters, including the number of edge servers participating in each round of global model coordination, the number of local computational steps performed by each edge server, and the total number of global coordination rounds required to achieve a given model's accuracy [23]. Although studying the effects of these parameters has been done, most of these studies have concentrated on optimising a single parameter that could only affect data processing and computation at edge servers. Furthermore, because the hyper-parameterised model contains original data, a client's data need must be disrupted with additional noise to protect sensitive data. Traditional perturbation techniques, on the other hand, have some limitations.

Moreover, Edge computing enhances the performance of connectivity, speed, and reduces latency of data transfer [24] from user to server. To manage the privacy preservation in edge servers using machine learning techniques is necessary; the realms of diagnosis, prognosis, and spread control, as well as assistive systems, monitoring, and logistics, have been split into various IoT-based healthcare applications [25]. The characteristics of smartphones to control and monitor implantable medical devices, as well as some practical security methods that improve the security and privacy of patients, is addressed in [26].

Key Contributions - This study proposes a novel conceptual three-fold hierarchical privacy-preserving framework known as

a federated edge aggregator over a typical federated learning method for IoT-based Smart Health Systems. By instituting edge aggregator between client and server to process learned model before global updating as Edge Intelligence. We assume that IoT-based smart healthcare systems obtained heterogeneous data from various data sources such as rural hospitals, remote pathology, and home-based self-isolated patients' health monitoring data. To ensure data variation for privacy-preserving evaluation, we experiment on the non-IID data distribution of the dataset. By leveraging the proposed Federated Edge Aggregator method, individual health IoT data is learned in its smart device using the Convolutional Neural Network (CNN) model. After perturbing its parameters with differential privacy, they are sent to the edge layer.

After several iterations, until the model converged, this blended model was sent for Global aggregation to the central healthcare service provider. To classify smart health monitoring groups, a couple of edge aggregator is distributed where each edge aggregation will process each sector, such as pathology or isolated patients' vitals data. The central healthcare provider aggregates all the received models for final update and broadcast. To our best knowledge, Edge Intelligence for privacy preservation in federated learning and differential privacy has not been investigated before. In our context patients, monitoring and data learning remains in patients' end smart devices and share data to the service provider for early detection before critical situation without traceability of the client's identity and avoiding other privacy attacks. This paper also shows the robustness of the proposed method by evaluating existing popular datasets in both Grayscale and RGB compared with the baseline traditional federated learning method. The contributions stated below are the focus of this paper:

—We propose a hierarchical three-fold Federated Edge Aggregator (FEA) framework with differential privacy applied with edge intelligence.

—We design a framework that ensures data privacy by leveraging the Gaussian noise adding mechanism before passing model parameters under certain noise perturbation levels.

—We show the federated learning methods privacy attacks classifications and possible defence against those threats with extensions.

—We establish an optimal privacy-preserving method to balance privacy budget, model accuracy, and training, test loss calculation in different data distribution strategies, and visualise the trade-off between privacy protection level and convergence performance with different datasets and compare with the recent method.

—We showed that our proposed method accounts for higher performance than the baseline method and different hyperparameter settings.

The paper is organised as follows. Section II focuses on background and related work based on recent research to the proposed study. The essential perceptions and schemes used in the proposed method are described in Section III. Section IV explains the proposed Federated Edge Aggregator method and mathematical analysis. Section V demonstrates the experimental results and performance evaluation of the proposed methods.

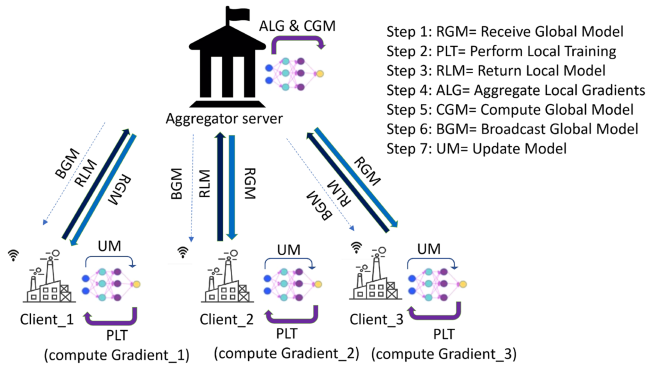

Fig. 1. A Complete scenario of the Federated Learning procedure.

Finally, Section VI accomplishes the work with the conclusion and delivers future research directions.

II. Background and Related Works

A. Federated Learning

Since 2015, the concept of federated learning has attracted much attention [27]. It holds much promise in healthcare data analytics [28]. Because data owners in a federated learning system are not required to submit their data to the central aggregator, learning is done through a sharing mechanism rather than raw data transmission, ensuring confidentiality and data privacy at a cost significantly greater than the loss of accuracy [15]. Each node and central aggregator train a combined Neural Network(NN) model in federated learning. To ensure that the neural network model converges, the nodes communicate their gradients to the central server, aggregating all gradients and providing updates to each node [29]. $T_i(i = 1, 2, \ldots, n)$ have agreed on a model design if each branch has n IoT nodes, as shown in Fig. 1.

The mathematical concept of federated learning is described as follows. The function $f(x, N)$ is a Neural Network model, where x is the inputs from the sensors and N the model's parameters, which contains all the biases and connections among all the neurons. If the nodes in the branch T_i hold the training set $(D^i$={$(x_i, y_i) \mid j$= 1,2,...,P}, where x_i is the attributes, y_i the class labels and P the size of D^i), the loss function $L_f(D^i, N)$ can be defined as:

$$L_f(D^i, N) = \frac{1}{P} \sum_{(x_j, y_j)} (y_j - f(x_j, N))^2 \tag{1}$$

The aim of training the NN model is to obtain the gradient to update N to reduce the amount of the loss function $L_f(D^i, N)$. The stochastic gradient descent (SGD) is used to calculate the participant Ti's gradient (w_i) as:

$$w_i = \nabla L_f(D^* i, N) \tag{2}$$

where ∇L_f is a derivative of the loss function (L_f) and $D^* i$ a random subset of D^i. Then, participants upload their gradients

TABLE I
SUMMARY OF KEY NOTATIONS

Math Symbols	Descriptions
T	Number of participants
i	Number of IoT node
f	Function of neural network model
N	Model parameter
D^i	Training set
χ	Input/attribute
y	Output/Class label
P	Size of D^i
L_f	Loss function
w	Gradient
f	Derivative of the loss function
η	Learning rate
F	Randomised function
Pr	Probability
d1	Dataset 1
d2	Dataset 2
C	All subset of F
ε	Privacy budget
σ	Noise scale
N	Noise distribution function
p	Sampling probability
M	Samples
N	Numbers

to the central server for aggregation as:

$$w = \sum_{i=1}^{n} w_i \tag{3}$$

Finally, the central server sends the aggregated updates to every member, and members update model parameter N = N-$\eta \cdot$ w/n after receiving w, where is the learning rate. If the closure conditions are not encountered, continue the next round of federated learning. Because this approach does not rely on the exchange of client data – only the updated model parameters – it provides immediate client privacy. Furthermore, Federated Aggregator algorithms such as FedAvg and FedProx work effectively in the presence of non-iid client data distributions [30]. Table I describes the mathematical symbols briefly.

B. IoT-Enabled Smart Healthcare Systems

The IoT is a massive, embedded network of connected physical objects and people. IoT devices usually generate data and exchange data through a network. IoT is emerging with tremendous revolutionisation in different applications, for example, smart cities, smart medical systems, Industry 4.0, agriculture 4.0, etc. For standardisation of IoT on these applications, Artificial Intelligence brings IoT in unique deployment, smart management, smart prediction, and decision making and introduces advanced technology that requires privacy involvement. Edge nodes are essential as brokers to mitigate privacy issues and efficiently manage massive IoT data integration [31]. Edge intelligence improves and speeds performance while safeguarding user privacy [32].

A vast population means more chronic diseases and critical situations requiring frequent visits to healthcare providers, higher medical treatment expenses, and the need to monitor vital status. As a result, IoT technology appears to be a potential solution for creating smart health systems [33]. Many disease detection schemes can be deployed using smart sensors and connecting with smart devices in the user's home environment. These schemes collect users' health-related data and process it in the data analysis cloud. As a remote detection system, smart health systems can greatly aid isolated patients who have problems moving or are incapacitated. Smart healthcare has seen several success stories because of the amount of user-health data accessed by IoT devices and recent breakthroughs in Machine Learning [34].

C. Differential Privacy

Because of its outstanding privacy guarantees, Differential Privacy is commonly used. The privacy leakage of single data belonging to a person in a dataset is addressed when some information from that dataset is made publicly available. So that to establish a strong privacy assurance, differential privacy brings a robust statistical method. As discussed below, differential privacy is defined as a mathematical computation and algorithmic.

Definition 1: A randomised function F gives ϵ-differential privacy if for all data sets d1 and d2 differing on at most one element, and all C $\subseteq$ Range(F),

$$Pr[F(d1) \in C] \leq exp(\epsilon) \times Pr[F(d2) \in C] \tag{4}$$

Where Pr[F(d1)] is a probability of randomised function F of dataset d1 and Pr[F(d2)] is a probability of randomised function F of dataset d2. F is a mechanism that meets this description, which focuses on worries about personal information leaking even if the user removes data from the data collection. No outputs are more or less likely. C is all subset of F and is a constant specified by the user, and exp is the base of the natural logarithms. ϵ is a measure of The IoT is a massive, embedded network of connected physical objects and people. IoT devices usually generate data and exchange data through a network. Nowadays, IoT is emerging with a tremendous privacy budget, where less value indicates less privacy loss. If ϵ is 0, that means no privacy loss.

Definition 2: A randomised function F gives (ϵ, δ)-differential privacy if for all data sets d1 and d2 differing on at most one element, and all C $\subseteq$ Range(F),

$$Pr[F(d1) \in C] \leq exp(\epsilon) \times Pr[F(d2) \in C] + \delta \tag{5}$$

(ϵ, δ)-differential privacy says that for every pair of neighbouring databases d1, d2, it is improbable that the observed value F(d1) will be much more or much less likely to be generated when the database is d1 than when the database is d2. It ensures that for all adjacent d1, d2, the absolute value of the privacy loss will be bounded by ϵ with a probability of at least $1 - \delta$.

III. PRELIMINARIES

A. Edge Intelligence

This section briefly discusses the underlying principles of Edge Intelligence and the artificial noise-adding processes utilised to construct the suggested work. Edge Intelligence is

the concept of deploying computationally expensive intelligent models (primarily machine learning, deep learning, and data analytics) on edge clouds rather than the central cloud [36]. Edge computing may deliver enormous quantities of data directly to the cloud. It can receive data from the edge layer before it goes to the aggregator and determines its importance [37]. However, edge computing is a type of computing between smart end-devices at the network's edge and traditional cloud or data centres [38]. Where Fog computing allows relevant data storage, irrelevant data deleting and analysing data for remote access, and localised learning models. Instead of depending on a centralised server, an edge layer has been proposed to pre-process massive IoT data before uploading and storing resources closer to the devices where data is gathered.

Edge Intelligence [39] significantly reduces communication costs and simultaneously plays data consumer and data generator by executing multitasking such as caching, managing, offloading, and service delivery. For efficient responsibilities in the network, edge nodes must be well-designed to meet major service requirements, such as dependability, security, and privacy protection. Due to low response times, communication costs, and higher privacy traits, edge intelligence is a great emerging advantage for clients to process sensitive data before uploading it to a centralised aggregator server.

B. Threat Model

Unlike traditional machine learning systems, federated learning systems must contend with three potential adversaries: clients, aggregators, and outsiders or eavesdroppers. In general, numerous protection measures defend federated learning from a wide range of threats, reducing the chance of privacy issues.

C. Privacy-Preserving

Privacy preservation is an essential demand for users, clients, and service providers. Privacy might need any data that can be sourced from the Internet of Things, social networks, Intelligent business applications, databases, documents, the web, etc. [41]. In cyber security, privacy might include personal information, individual privacy, interactive privacy, and communication privacy [42]. To protect digital assets' privacy, cyber security mechanisms are defined as Privacy preservation. Different types of privacy preservation approaches are grouped into five categories [14] encryption-based, perturbation-based, authentication-based, differential privacy (DP), and blockchain-based. A wide range of research is going on to provide privacy preservation schemes. There have been numerous studies related to privacy preservation in smart healthcare systems. We can classify as shown in Fig. 2.

D. Noise Adding Before Aggregation

There are four popular noise mechanisms in differential privacy: Exponential, Laplacian, Gaussian, and Hybrid mechanisms [43]. The Gaussian mechanism mostly complies with (ϵ, δ)-differential privacy rather than pure -differential privacy.

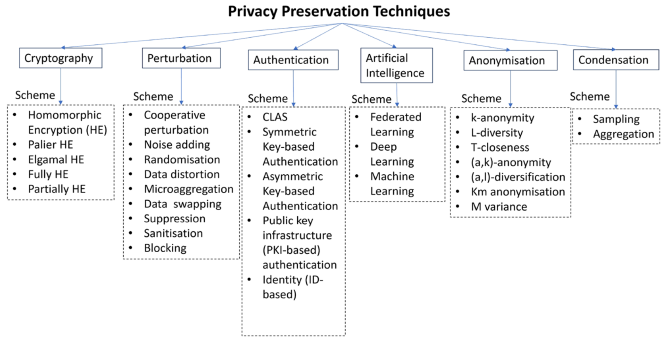

Fig. 2. Classification of privacy preservation techniques.

Hence, we consider our proposed method's Gaussian mechanism for adding noise. In the Gaussian mechanism, Additive noise is combined with numeric queries and both IID and non-IID entries in the Gaussian technique. When the wireless federated learning method is employed for its iterations, it may be used for Local Differential Privacy to acquire the entire privacy leakage [44]. Gaussian mechanism returns several functions f(d1), and the definition F(d1) implies with (ϵ, δ) -differential privacy.

$$F(d1) = f(d1) + \nu(\sigma)^2 \tag{6}$$

Most of the research has focused on determining the privacy loss for a given noise distribution and the composition of privacy losses. For the Gaussian noise, here used noise scale that the moment accountant maintains track of a bound on the random variable's moments of privacy loss, ensuring efficient noise distribution.

E. Moment's accountant theorem Nowadays, the study of privacy loss for a specific noise distribution and the composition of privacy losses has gotten much attention. We prefer noise scale σ for the Gaussian noise with sampling probability p, the ratio of M samples and N numbers.

$$p = \frac{M}{N} \tag{7}$$

After T number of steps, privacy loss ϵ is accounted for in differential privacy with constant c,

$$\epsilon < cp^2T \tag{8}$$

For (ϵ, δ)-differentially private stochastic gradient descent (SGD) algorithm, we choose,

$$\sigma > \frac{c\left(p\sqrt{\left(T\log(\frac{1}{\delta})\right)}\right.}{\epsilon} \tag{9}$$

We consider the moment accountant theorem in our method.

IV. FL-Based Privacy Protection Method

A. System Overview

By integrating edge aggregator at the intermediate level with traditional FL architecture, the proposed 3-fold Federated Edge Aggregator framework ensures individual client privacy and data privacy, as shown in Fig. 3. This framework illustrates a typical

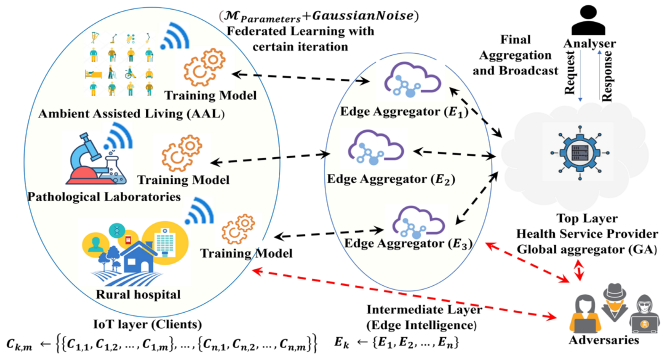

Fig. 3. A Conventional scenario to train and deploy federated learning in Smart Healthcare Systems.

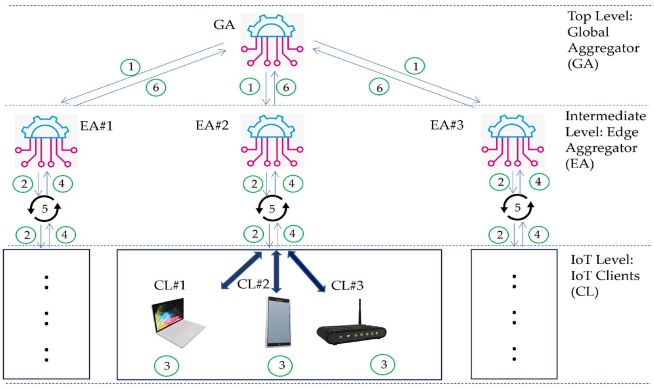

Fig. 4. Proposed framework of three-fold edge aggregator-based federated learning in smart healthcare systems.

scenario for training and deploying federated learning using Federated Edge Aggregator on an IoT-based Smart Healthcare System, collaborating with application sectors, an inquiry analyser, and adversaries. Individual edge aggregator is responsible for aggregating and broadcasting the respective Client's Model for some iterations to produce a better-converged model at the edge. After aggregating models from individual clients, the edge aggregator will pass the model to the Global Aggregator (GA).

The proposed method in Fig. 4 visualises the steps that are indicated by circled numbering, and labels are titled: Top Level, Intermediate Level, and IoT Level. The complete functionality of the levels is described below:

- Top Level: The top-level contains one GA as a tree hierarchy of the proposed 3-folds Federated Edge Aggregator. The primary responsibility of GA is to execute step-1 and step-7.
- Intermediate Level: The proposed 3-folds FEA introduces a set of EAs at the intermediate level to preserve IoT clients' privacy and their data through steps 2–5.
- IoT Level: The bottom level of the proposed 3-fold Federated Edge Aggregator contains a set of clients, including laptops, TAB, SmartPhone, and Sensing Devices. The primary function in this level is to execute step 3 and communicate with the Intermediate level.

The steps-by step descriptions of the proposed 3-folds FEA framework are as below:

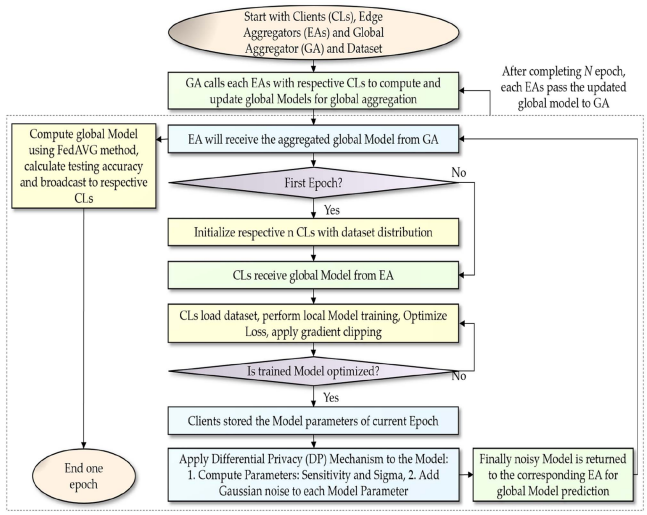

Fig. 5. Flowchart of the proposed 3-folds FEA method.

- Step 1: Global aggregator initialises edge aggregators, hyperparameters, model parameters, noise parameters, dataset, and distribution method.
- Step 2: Edge aggregators initialise corresponding IoT clients, passing model parameters and assigned data.
- Step 3: IoT clients load the training set and label to train the model, calculate training loss, employ optimisation technique, load model parameters, add Gaussian noise to each model parameter and update the model
- Step 4: Edge aggregator accumulates model parameters of the corresponding set of IoT clients. Edge aggregator broadcasts updated model to the connected IoT clients edge aggregator calculates testing accuracy test data and corresponding label.
- Step 5: Repeat Step 2-Step 5 until the model converged/stabilised.
- Step 6: Global aggregator accumulates edge aggregators model parameters. The global aggregator broadcasts an updated model to the connected edge aggregators that calculate overall testing accuracy from test data and corresponding labels.

Edge intelligence is a hybrid of edge computing and artificial intelligence that takes advantage of end devices' storage, networking, and processing capabilities to enable edge caching, model training, and inference closer to the source of data. So, instituting edge intelligence at the intermediate level brings a viable solution to address privacy attacks. Edge intelligence allows data owners to process models without transmitting their sensitive parameters directly to third-party servers.

The flowchart in Fig. 5 represents the overall workflow of the proposed technique, which starts with the global aggregator, edge aggregator, and client initialising. And the process shows for the N epoch until the model converged. And the conception of differential privacy, in which each client perturbs its learned parameters locally by purposefully adding noise before uploading them to the server for aggregation, is used to avoid information leaking successfully. Fig. 5 shows

Algorithm 1. Privacy Protection in 3-fold FEA.

Input: Global Aggregator G, Set of Edge Aggregators $E_k \leftarrow \{E_1, E_2, \ldots, E_n\}$, Set of Clients $C_{k,m} \leftarrow \{\{C_{1,1}, C_{1,2}, \ldots, C_{1,m}\}, \ldots, \{C_{n,1}, C_{n,2}, \ldots, C_{n,m}\}\}$ Dataset $\mathcal{D}$

Output: Model Partial Privacy $\mathcal{M}_{E_k}$, Model Complete Privacy $\mathcal{M}_G$, Accuracy Partial Privacy $\mathcal{A}_{E_k}$, Accuracy Complete Privacy $\mathcal{A}_G$

Initialise Hyperparameters: Learning Rate η, Batch Size β, Communication Rounds $\mathcal{N}$, Privacy Bounds: Epsilon (ε), Delta (δ);
$UserDataDictionary \leftarrow DistributionData(\mathcal{D}, C_{k,m})$ [*Algorithm* 2]
for $i = 1, 2, \ldots\ldots, k$ **do** // k is the number of Edge Aggregators (EA)
  for $j = 1, 2, \ldots\ldots, \mathcal{N}$ **do** // $\mathcal{N}$ no. of iterations for each EA
    $C_{i,m} \leftarrow \{C_{i,1}, C_{i,2}, \ldots, C_{i,m}\}$ *random Clients set of* i^{th} *Edge Aggregator*
    for $l = 1, 2, \ldots\ldots, m$ **do** // m is the number of Clients per EA
      $\{TrainData_{C_{i,l}}, TrainLabel_{C_{i,l}}\} \leftarrow UserDataDictionary$
      $\mathcal{M}_{C_{i,l}} \leftarrow Model(TrainData_{C_{i,l}}, TrainLabel_{C_{i,l}}, Hyperparameters, Optimization)$ //CNN training
      $\mathcal{M}_{Parameters} \leftarrow Load\ Model\ (\mathcal{M}_{C_{i,l}})\ Parameters\ from\ Dictionary$
      $\mathcal{M}_{Parameters} \leftarrow \mathcal{M}_{Parameters} + GaussianNoise\ (\varepsilon, \delta)$
      $\mathcal{M}_{C_{i,l}} \leftarrow Update\ Dictionary\ with\ \mathcal{M}_{Parameters}\ of\ Model\ \mathcal{M}_{C_{i,l}}$
    end for
    $\mathcal{M}_{E_{i,j}} \leftarrow Aggregation\ (\mathcal{M}_{C_{i,1}}, \ldots, \mathcal{M}_{C_{i,m}})$ //EA will aggregate using FedAvg Algorithm
    for $l = 1, 2, \ldots\ldots, m$ **do**
      $Broadcast\ (C_{i,m}, \mathcal{M}_{E_{i,j}}) \rightarrow Clients\ Receive\ Model\ Parameters$
    end for
    $\{TestData, TestLabel\} \leftarrow \mathcal{D}$
    $\mathcal{A}_{E_{i,j}} \leftarrow TestingAccuracy\ (\{TestData, TestLabel\}, \mathcal{M}_{E_{i,j}})$
  end for
  $\mathcal{M}_{E_i} \leftarrow \mathcal{M}_{E_{i,\mathcal{N}}}$ *Final Model of* i^{th} *Edge Aggregator after* $\mathcal{N}^{th}$ *iteration*
  $\mathcal{A}_{E_i} \leftarrow \mathcal{A}_{E_{i,\mathcal{N}}}$ *Final Accuracy of* i^{th} *Edge Aggregator at* $\mathcal{N}^{th}$ *iteration*
end for
$\mathcal{M}_G \leftarrow Aggregation\ (\mathcal{M}_{E_1}, \mathcal{M}_{E_1}, \ldots, \mathcal{M}_{E_k})$ //GA will aggregate using FedAvg Algorithm
for $l = 1, 2, \ldots\ldots, m$ **do**
  $Broadcast\ (E_k, \mathcal{M}_G) \rightarrow Edge\ Aggregators\ Receive\ Model\ Parameters$
end for
$\{TestData, TestLabel\} \leftarrow \mathcal{D}$
$\mathcal{A}_G \leftarrow TestingAccuracy\ (\{TestData, TestLabel\}, \mathcal{M}_G)$
return $\mathcal{M}_{E_k}, \mathcal{A}_{E_k}, \mathcal{M}_G, \mathcal{A}_G$

Algorithm 2. Data Distribution of 3-fold FEA.

Input: Dataset $\mathcal{D}$, Set of Edge Aggregators $E_k \leftarrow \{E_1, E_2, \ldots, E_n\}$, Set of Clients $C_{k,m} \leftarrow \{\{C_{1,1}, C_{1,2}, \ldots, C_{1,m}\}, \ldots, \{C_{n,1}, C_{n,2}, \ldots, C_{n,m}\}\}$, Distribution Method $\aleph$, Shard Parameter τ,

Output: $UserDictionary[k][m]$

Initialise: Training and Testing Data from $\mathcal{D}$, Transform Data to Tensor, Apply Normalisation on Data
$\{TrainData, TrainLabel\} \leftarrow Training$ // Split data and label
$\{TestData, TestLabel\} \leftarrow Testing$ // Split data and label
if $\aleph$ = 'non-iid' **then**
  $Index\ have\ completed \leftarrow sort(TrainLabel)$ //user's data will differ from another user
else if $\aleph$ = 'iid' **then**
  $Index \leftarrow random(TrainLabel)$ // Users may get similar data labels
end if
for $i = 1, 2, \ldots\ldots, k$ **do** // k is the number of Edge Aggregators (EA)
  $AvailableShards \leftarrow \tau \times m$ // Divide data into number of chunks
  $DataPerShard \leftarrow \frac{length(TrainData)}{AvailableShards}$ //amount of data per chunks
  for $j = 1, 2, \ldots\ldots, m$ **do** // m is the number of IoT Clients
    $UserShardIndex \leftarrow random(AvailableShards, \tau)$ // Assign chunks to individual user on random basis
    for $k = 1, 2, \ldots\ldots, \tau$ **do** // For each chunk
      $UserIndex[k \times DataPerShard] \leftarrow [(DataPerShard \times UserShardIndex) : (DataPerShard \times UserShardIndex + 1)]$ //All the Indexes for an individual user
    end for
    $AvailableShards \leftarrow AvailableShards - UserShardIndex$ //Previously Assigned chunks will be removed for next selection
    $FinalUserIndex \leftarrow Index[UserIndex]$ //Pickup Label indexes
    $Data \leftarrow TrainData[FinalUserIndex]$ // Selected data by indexes
    $Label \leftarrow TrainLabel[FinalUserIndex]$ // Selected label by indexes
    $UserDictionary[i][j] \leftarrow append(Data, Label)$ // data and label of user
  end for
  $UserDictionary[i][j] \leftarrow append(TestData, TestLabel)$
end for
return $UserDictionary[k][m]$ //Complete dictionary for m users of k EA

the whole working procedure of the method by evaluating an existing dataset.

B. Proposed Three-Fold FEA Algorithm

The pseudocode of the proposed three-fold Federated Edge Aggregator is presented in Algorithm 1. This algorithm presents a complete procedure from IoT clients' assignment to the final aggregated model. Firstly, a set of EA ($E_k \leftarrow \{E_1, E_2, \ldots, E_n\}$) and corresponding Clients ($(C_k, m \leftarrow \{C_{1,1}, C_{1,2}, \ldots, C_{1,m}\}, \ldots, \{C_{n,1}, C_{n,2}, \ldots, C_{n,m}\})$) is initialised with default hyperparameters and dataset. Then based on the data distribution method (IID or non-IID), data is distributed among a certain number of clients corresponding to each edge aggregator as described in Algorithm 2.

For each iteration $j (j = 1, 2, \ldots, N)$, individual IoT Clients will load their assigned data, train the model, compute training loss, apply optimisation to reduce training loss, and add Gaussian noise to the model parameters. Once all the clients execute the procedures, the corresponding edge aggregator will aggregate clients model, broadcast updated model among respective clients and compute testing accuracy for a certain number of iterations, N. After that, the global aggregator will aggregate EAs Model, broadcast the updated model to the corresponding EAs and finally, compute the global model accuracy. This joint resource allocation and incentive design framework for edge intelligence reduce reliance on a central controller.

Based on the data distribution method, Algorithm 2 distributes the amount of training data with corresponding labels to each client. The non-IID method firstly sorts the training dataset as per labels and then distributes it among Clients; however, the IID method randomly as-signs the training data. The non-IID-based method ensures a different dataset for each client, whereas the IID method may assign a similar dataset. It is to be noted that the proposed method ensures robustness by employing the non-IID method along with the IID method in terms of data distribution.

C. System Analysis

This section investigates the computational complexity and the level of privacy of IoT data of the proposed 3- fold Federated Edge Aggregator model.

Analysis of Computational Complexity: To analyse the computational complexity of our system employing Big notation. At first, we investigate the computational complexity of the baseline method that does not have an intermediate layer. We use a basic federated learning approach as a baseline method and add noise before aggregating it to the central server. The complexity of federated learning is (I *N *G), where I represent the number of iterations, N is the total number of clients, and G is the centre number.

Secondly, we analysed the time complexity of this Baseline method as (C), where C refers to the number of classes. In Our proposed method, the computational complexity involves the IoT, Intermediate, and Top layers. The complexity of all layers is (I+E+N/n+G). Here, I mean the number of Iteration, E means edge aggregator, N/n implies N number edge aggregator distributes n clients, and G is a global aggregator. Additionally, the worst-case scenario is estimated by (A). Similarly, the time complexity of our proposed method is (n), where n is the number of EA used for enhancing privacy preservation.

Analysis of the suitability of resource-constrained devices: Machine learning methods are challenging to implement because they demand sufficient CPU power and big storage memory. To investigate the suitability of resource-constrained

devices, our method supports friendly, lightweight end devices for end data processing. Additionally, the edge intelligence device at the edge server and central service provider is used for overall model updating and analysis.

Analysis of the attack resistance via edge intelligence: the essential part of privacy preservation is investigated here. Initially, we assume that our edge aggregator is trusted and located near edge nodes. A couple of edge aggregator is distributed for purpose-based (rural hospital or pathological laboratory) clients' IoT network. All the edge aggregator is connected with the global aggregator (central telehealth service provider). Here we consider global aggregator might be honest but has curious behaviour. Clients directly communicate with a global aggregator in a traditional federated learning mechanism.

In this case, an adversary might trace or localise the client's identity by attacking a global aggregator. Similarly, a global aggregator might be able to analyse clients' patterns because it communicates directly to get model parameters for update every time. At the same time, if the global aggregator faces privacy breaches, all the related client's privacy might fall apart. Hence, edge intelligence performs a broker for model pre-processing at the edge layer before sending it to a global aggregator. The edge aggregator sends a blended model after running a certain iteration of several clients' updating models. The global aggregator will be unaware of the client's information. Moreover, the whole system collapse probability becomes reduced because all the clients are not connected at a single point.

V. Experimental Results and Performance Evaluation

A. Environment Setup

We perform our experiments using Microsoft Windows 10 Home with Intel Core^{TM} i7 processor (2.6–5.0 GHz), 16 GB RAM NVIDIA GeForce GTX 1650 Ti with Max-Q design graphics processor HP Laptop, and 1 TB SSD. Python 3.6, Pysift, Pytorch library, and PyCharm 2021.1.2 are used as development tools.

B. Datasets

Federated learning models with Differential Privacy are implemented and tested on a dataset from the Modified National Institute of Standards and Technology (MNIST) database [13]. The testing accuracy and computational durations levels are investigated as a baseline technique by adjusting different hyperparameters, as indicated in Table II.

The MNIST database is enormous, with handwritten digits extensively used in machine learning training and testing. It includes 60,000 training and 10,000 testing grayscale images with a $28 \times 28 \times 1$ resolution and ten output classes labeled 0 to 9. In the RGB dataset, we use the popular CIFAR10 dataset containing a 50000 training set and 10000 testing set with ten labels where the input image dimension is 32×32. The STL10 dataset contains 5000 training sets and 8000 testing sets with 10 labels, where the input image dimension is 96 x 96. Due to the low amount of training set of STL10 dataset, overall learning may impact.

TABLE II
Hyperparameters With Values Used for Our Federated Edge Aggregator Method Analysis

Hyperparameter	Notation	Value
Communication Round/Iteration	I	100
Local Epoch/Iteration	Li	1
Batch size	B	10
Learning Rate	η	0.01
Delta	δ	1e-5=0.00001
Epsilon	ϵ	4.0
Sampling Rate	P	3%=0.03
Training loss Gradient clipping	C	32
Output class	O	10
Noise scale	Σ	4

To show the performance of medical images to justify the smart healthcare system, we choose the newly released COVID19 Chest X-RAY image dataset with 4 labels 20685 training set (COVID19: 3496, Lung Opacity: 5892, Normal: 10072, Pneumonia:1225) and 240 testing set, Whereas Input image dimension 299 $\times$299. Noted that, due to our machine's memory limitation, we had to reduce image dimensions during the experiment and resized image dimension was 32×32, which might impact the overall accuracy calculation in our method evaluation.

C. Data Distribution

The following two data distribution processes are investigated for existing datasets to assess the method's robustness. These are explained below:

—**IID:** The term "independent and identically distributed" (IID) refers to a sample of independent events in which all items are taken from the same probability distribution. In this distribution, the data is handled and shuffled in random order. So that two clients can get similar class label data. In IID distribution, divide the training data into equal-sized shards and allocate one shard to each client [17].

—**Non-IID:** Heterogeneity and hierarchical coupling connections among the data and clients are involved in non-independent and identically distributed (non-IID) data. The data must first be sorted before being provided to all clients so that comparable class labels are not assigned to them [45]. Non-IID data distribution is widely employed when user devices generate data.

Most of the experimental work on non-IID datasets has focused on label distribution bias, which consists in partitioning a flat with an existing dataset based on labels to create a non-IID dataset. In non-IID distribution, the unbalanced local data samples, labels, and features as the specific probability distribution of training examples stored at local nodes can substantially impact the training process's performance. In non-IID distribution, data divides are nonoverlapping and balanced, resulting in the same number of data points for each client [17].

D. Model

The study was carried out using a two-layer sequential Convolutional Neural Network (CNN) model with a fully connected layer on top. Each layer has a convolution with a kernel size of 5X5, batch normalisation, and the activation function RELU.

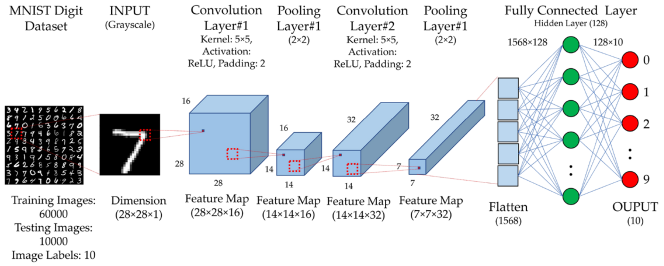

Fig. 6. A two-layer sequential Convolutional Neural Network (CNN) model with a fully connected layer.

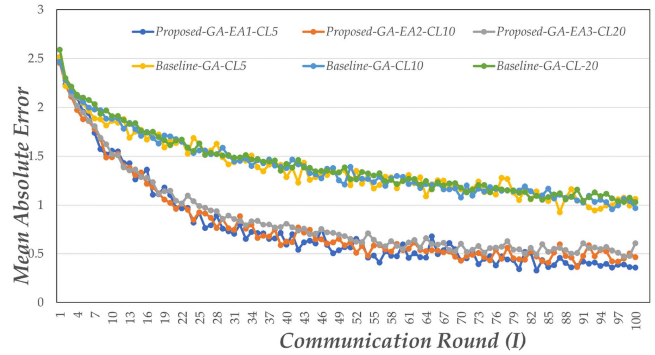

Fig. 7. Training error comparison for non-IID data distribution of the MNIST dataset: Proposed vs. Baseline.

Fig. 6 shows the procedure of the CNN model, which is how patients' data will be learned from their own devices.

As a feed-forward neural network convolutional neural network(CNN) use filters and pooling layers, whereas some other Deep learning model feed results back into the network. Basically, the input size and the resulting output size are fixed in CNN. It automatically identifies the critical features without human supervision with limited data whereas the Deep learning model requires huge data to perform better. And deep learning is expensive and has training complexity.

The non-IID method was implemented to split and assign training data to each client. It is to be noted that the non-IID-based technique firstly sorts training data as per label and then gives it to clients; hence, it is ensured that each client will get a maximum of three-digit variations data. The IID-based technique, on the other hand, does not require sorting.

E. Experimental Results and Discussion

We performed several experiments with different datasets to study the impact of differential privacy on model accuracy under various privacy loss settings.

- **Training Error Analysis:** Fig. 7 and Fig. 8 demonstrates the proposed and baseline methods' training error in terms of Mean Absolute Error (MAE). We consider the MNIST dataset with two distribution methods: IID and non-IID. The proposed method reported lower training error than the baseline method for certain communication rounds in both data distribution methods.

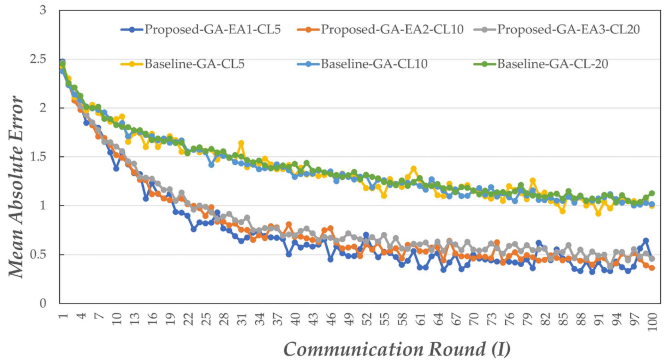

Fig. 8. Training error comparison for IID data distribution of the MNIST dataset: Proposed vs. Baseline.

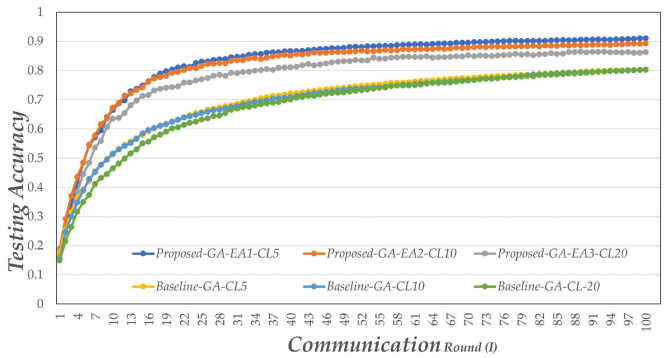

Fig. 9. Testing accuracy comparison for non-IID data distribution of the MNIST dataset: Proposed vs. Baseline.

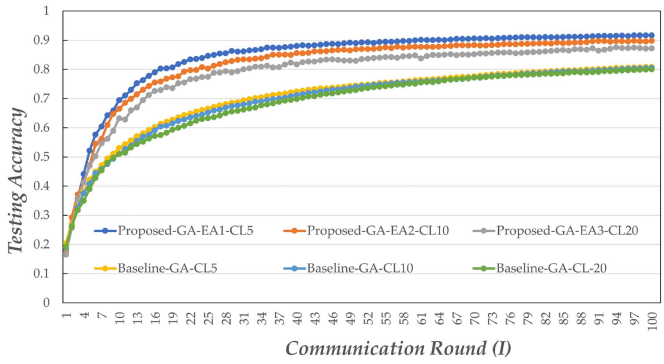

Fig. 10. Testing accuracy comparison for IID data distribution of the MNIST dataset: Proposed vs. Baseline.

- **Testing Accuracy Analysis:** Fig. 9 and Fig. 10 illustrate the proposed and baseline methods' testing accuracy in terms of the MNIST dataset with two distribution methods: non-IID and IID. Fig. 9 shows the reported 80% and 90% testing accuracy at the 20th and 75th communication rounds, respectively, for the proposed method; however, 62% and 78% are reported for the same communication rounds of the baseline method in terms of non-IID distribution.

  Fig. 10 shows the reported 80% and 90% testing accuracy at the 17th and 61st communication rounds, respectively,

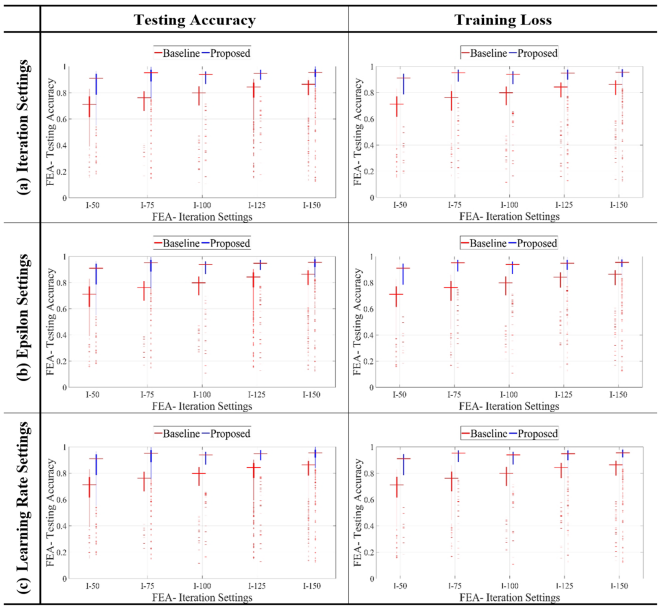

Fig. 11. Box Plot of Hyperparameters different settings in terms of testing accuracy and training loss dem onstration for Baseline and Proposed method. (a) Iteration Settings. (b) Epsilon Setting. (c) Learning Rate Setting.

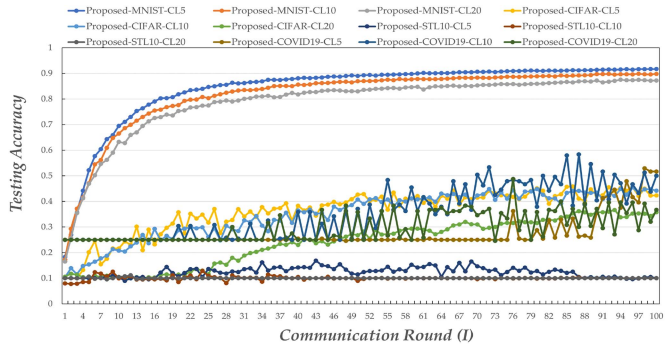

Fig. 12. Performance analysis of our proposed method with different datasets (MNIST, CIFAR10, STL10, and COVID19 chest x-ray).

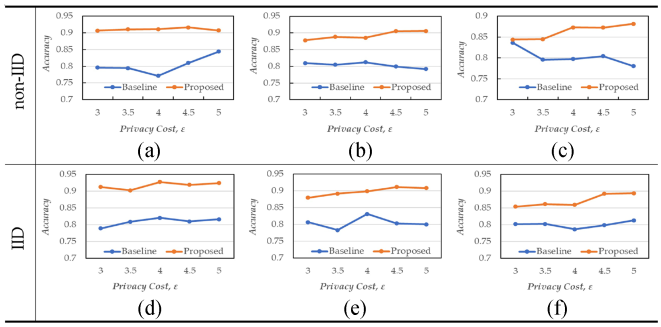

Fig. 13. Levels of testing accuracy vs of proposed and baseline methods for different data distributions: non-IID based with (a) 5 IoT clients, (b) 10 IoT clients, and (c) 20 IoT clients; and IID based with (d).

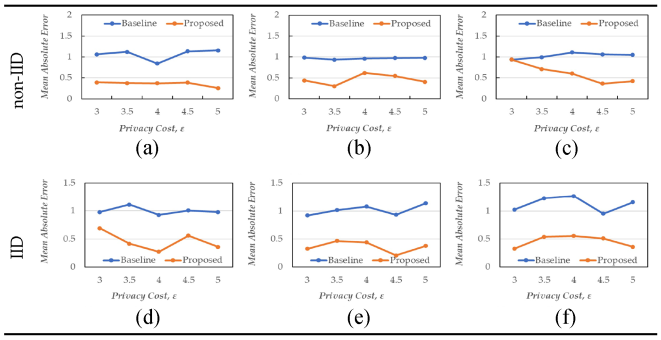

Fig. 14. Training losses vs of proposed and baseline methods for different data distributions: non-IID based with (a) 5 IoT clients, (b) 10 IoT clients, and (c) 20 IoT clients; and IID based with (d) 5 IoT clients, (e) 10 IoT clients and 20 IoT clients.

for the proposed method; however, 61% and 76% are reported for the same communication rounds of the baseline method in terms of IID distribution. As can be seen, in both cases, the proposed method reported higher testing accuracy than the baseline method for the entire communication rounds.

- **Performance Analysis based on Hyperparameter Different Settings:** The box plots in Fig. 11 demonstrates the proposed and baseline methods' testing accuracy and training error in terms of the hyperparameter's different settings. We consider the MNIST dataset in this regard for the performance evaluation. The proposed method reported a higher median value (bold red hyphen marked on the graphs) than the baseline method for all the settings of iteration (Fig. 11(a)) and epsilon (Fig. 11(b)) in terms of testing accuracy prediction. The proposed method reported lower median values in the box plots than the baseline methods for the settings mentioned above regarding training error prediction. It is also observed that the proposed method reports superior performance than the baseline method till a certain Learning Rate, as shown in Fig. 11(c).
- **Performance Analysis with different datasets:** Our proposed method has robustness in different datasets (MNIST, CIFAR10, STL10, and COVID19 chest x-ray) as well. We evaluate performance with recent medical datasets to justify our method's smart healthcare system (COVID19 chest x-ray). Fig. 12 shows how different datasets are adaptable and outperform our proposed method in terms of various client numbers.
- **Privacy Cost Analysis:** The privacy cost significantly impacts Model construction due to the amount of noise added during Model learning depending on this parameter. Theoretically, strong privacy indicates more noise in the model, which can be ensured by applying a small value; however, it costs Model accuracy. Hence, choosing an value is a trade-off between Model accuracy and privacy preservation. Fig. 13 and Fig. 14 demonstrate the testing accuracy and training loss for different settings of the value of the proposed method and baseline method. Considering both the IID and non-IID data distribution method of the MNIST dataset, five different settings of the privacy budget have been considered in this experiment 3.0, 3.5, 4.0, 4.5, and 5.0.

TABLE III
OVERALL ACCURACY AND COMPUTATION TIME FOR A DIFFERENT SETTING OF ϵ

Privacy Cost	Overall accuracy(%)		Computation Time(s)	
	non-IID	IID	non-IID	IID
$\epsilon = 3.0$	79.16	78.26	1970.313	3414.336
$\epsilon = 3.5$	80.30	70.21	2133.337	3447.497
$\epsilon = 4.0$	78.56	79.89	2337.833	3471.935
$\epsilon = 4.5$	79.60	75.31	2052.076	3453.754
$\epsilon = 5.0$	81.28	72.89	2024.520	3470.981

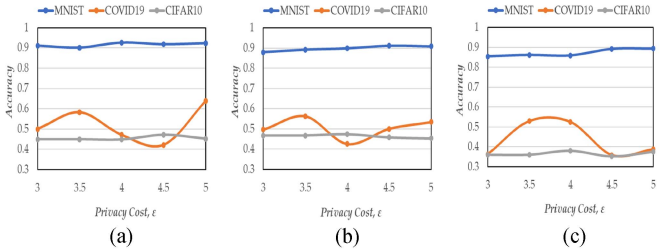

Fig. 15. Testing accuracy vs. privacy cost analysis of the proposed method for different datasets (MNIST,COVID19,CIFAR10).

The proposed system computes the overall testing accuracy by a global aggregator, aggregating the Models from individual edge aggregator and then applying the testing dataset to the final model. Table III lists the overall testing accuracy (in percentage) and computation time (in seconds) of the proposed method under different settings of. As can be seen, the privacy budget has less impact (78% 81% accuracy) for non-IID data distribution than for IID data distribution (70% 80% accuracy). Although the IID-based model takes 1.5 times more computation time than non-IID-based; however, the computation time varies a lot for non-IID (the 1970s-2337 s) compared with a slight variation for IID-based for different settings of.

- **Privacy Cost Analysis for different datasets:** Fig. 15 demonstrates the overall testing accuracy compared with the value of the proposed method in different datasets (MNIST, COVID19 chest x-ray, CIFAR10). Five different settings of the privacy budget have been considered in this experiment as well 3.0, 3.5, 4.0, 4.5, and 5.0. In this figure, we can see that our proposed method has met a significant privacy paradigm.

F. Advantages and Disadvantages

Using a popular large dataset, the suggested EI-based privacy preservation in FL produces promising results. Furthermore, fixed user numbers for each EA are assigned during the performance evaluation, such as 10 clients for the first EA, 5 for the second, and 20 for the third, demonstrating that the suggested method allows client flexibility.

Adding noise to a parameter of a learned model reduces data quality while increasing data privacy for a customer. A user's number is not fixed in real-time. EI is only utilised for pre-processing in the proposed method, not for programming the data collecting and decision-making functions. Although it is proved that privacy assaults may be defended in an FL framework, performance against various privacy attacks is not taken into account throughout the evaluation. These constraints will be overcome in the future by developing this work.

VI. CONCLUSION

A convergent iteration-based three-fold Federated Edge Aggregator architecture with differential privacy has been proposed and developed in this paper for smart healthcare systems. It shows edge intelligence for federated learning at the edge layer to help a collection of health organisations protect their privacy. After completing iterations, the edge aggregator delivers the blended model parameters to a central aggregator. As a result, the global aggregator cannot track crucial data about a single user. Extensive testing has revealed that this approach provides notable results regarding overall learning accuracy and time complexity. It also protects against global aggregator's unauthorised manipulation by providing extra privacy protection. The investigations involve training and testing popular datasets with IID and non-IID distributions and assessing the overall training loss and privacy cost. A baseline technique (NbAFL method) with no edge aggregator and a noised parameter transmitted directly to the central aggregator is used to evaluate performance at first. The suggested three-fold Federated Edge Aggregator approach is then tested and compared with the recent method. It is demonstrated to have a much better impact on learning accuracy, consume less computing time, have efficient data load control, and provide better privacy by assessing privacy costs. Comparative results show that this strategy offers a more privacy-preserving paradigm for the exact privacy cost as the baseline method. In the future, A fine-grained microservices-based Federated Edge Aggregator will be researched to follow this study.

REFERENCES

[1] L. Baghersalimi, T. Teijeiro, D. Atienza, and A. Aminifar, "Personalised real-time federated learning for epileptic seizure detection," *IEEE J. Biomed. Health Informat.*, vol. 26, no. 2, pp. 898–909, Feb. 2021.
[2] techjury.net, "49 stunning Internet of Things statistics [the rise of IoT]," Accessed: Jun. 2022. [Online]. Available: https://techjury.net/blog/internet-of-things-statistics/
[3] "HIPAA Australia: The privacy act," 1988. [Online]. Available: https://compliancy-group.com/hipaa-australia-the-privacy-act-1988
[4] S. Abdullah, J. Arshad, M. M. Khan, M. Alazab, and K. Salah, "PRISED tangle: A privacy-aware framework for smart healthcare data sharing using IOTA tangle," *Complex Intell. Syst.*, pp. 1–19, 2022, doi: 10.1007/s40747-021-00610-8.
[5] insidebigdata.com, "Why the future of healthcare is federated AI," Accessed: Jun. 2022. [Online]. Available: https://insidebigdata.com/2021/03/16/why-the-future-of-healthcare-is-federated-ai/
[6] N. Bugshan, I. Khalil, N. Moustafa, and M. S. Rahman, "Privacy-preserving microservices in industrial Internet of Things driven smart applications," *IEEE Internet Things J.*, early access, Jul. 21, 2021, doi: 10.1109/JIOT.2021.3098980.
[7] R. Wang, J. Lai, Z. Zhang, X. Li, P. Vijayakumar, and M. Karuppiah, "Privacy-preserving federated learning for internet of medical things under edge computing," *IEEE J. Biomed. Health Informat.*, early access, Mar. 8, 2022, doi: 10.1109/JBHI.2022.3157725.
[8] Z. Yan, J. Wicaksana, Z. Wang, X. Yang, and K. -T. Cheng, "Variation-aware federated learning with multi-source decentralized medical image data," *IEEE J. Biomed. Health Informat.*, vol. 25, no. 7, pp. 2615–2628, Jul. 2021, doi: 10.1109/JBHI.2020.3040015.
[9] W. Y. B. Lim et al., "Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 3, pp. 536–550, Mar. 2022, doi: 10.1109/TPDS.2021.3096076.

[10] K. Cao, Y. Cui, Z. Liu, W. Tan, and J. Weng, "Edge intelligent joint optimization for lifetime and latency in large-scale cyber-physical systems," *IEEE Internet Things J.*, early access, Aug. 04, 2021, doi: 10.1109/JIOT.2021.3102421.

[11] M. S. Rahman, I. Khalil, M. Atiquzzaman, and X. Yi, "Towards privacy preserving AI based composition framework in edge networks using fully homomorphic encryption," *Eng. Appl. Artif. Intell.*, vol. 94, 2020, Art. no. 103737.

[12] W. Fang, X. Z. Wen, Y. Zheng, and M. Zhou, "A survey of Big Data security and privacy preserving," *IETE Techn. Rev.*, vol. 34, no. 5, pp. 544–560, 2017.

[13] Y. Xiao, Y. Li, G. Shi, and H. V. Poor, "Optimising resource-efficiency for federated edge intelligence in IoT networks," in *Proc. Int. Conf. Wireless Commun. Signal Process.*, 2020, pp. 86–92, doi: 10.1109/WCSP49889.2020.9299798.

[14] M. Keshk, B. Turnbull, N. Moustafa, D. Vatsalan, and K. K. R. Choo, "A privacy-preserving-framework-based blockchain and deep learning for protecting smart power networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, 5110–5118, Aug. 2020.

[15] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated learning," *Synth. Lectures Artif. Intell. Mach. Learn.*, vol. 13, no. 3, pp. 1–207, 2019.

[16] M. Yamin, Y. Alsaawy, A. B. Alkhodre, A. Sen, and A. Ahmed, "An innovative method for preserving privacy in Internet of Things," *Sensors*, vol. 19, no. 15, 2019, Art. no. 3355.

[17] F. Sattler, S. Wiedemann, K. R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.

[18] K. Wei et al., "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 3454–3469, 2020, doi: 10.1109/TIFS.2020.2988575.

[19] J. J. P. C. Rodrigues, S. Jabbar, M. Abdallah, C. Verikoukis, and M. Guizani, "Future communication trends toward Internet of Things services and applications," *IEEE Wireless Commun.*, vol. 26, no. 6, pp. 6–8, Dec. 2019, doi: 10.1109/MWC.2019.8938176.

[20] X. Liu et al., "Privacy and security issues in deep learning: A survey," *IEEE Access*, vol. 9, pp. 4566–4593, 2021, doi: 10.1109/ACCESS.2020.3045078.

[21] I. Rosenberg, A. Shabtai, Y. Elovici, and L. Rokach, "Adversarial machine learning attacks and defense methods in the cyber security domain," *ACM Comput. Surv. (CSUR)*, vol. 54, no. 5, pp. 1–36, 2021.

[22] Y. Li, H. Li, G. Xu, T. Xiang, X. Huang, and R. Lu, "Toward secure and privacy-preserving distributed deep learning in fog-cloud computing," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11460–11472, Dec. 2020, doi: 10.1109/JIOT.2020.3012480.

[23] Q. Wang, Y. Xiao, H. Zhu, Z. Sun, Y. Li, and X. Ge, "Towards energy-efficient federated edge intelligence for IoT networks," in *Proc. IEEE 41st Int. Conf. Distrib. Comput. Syst. Workshops*, 2021, pp. 55–62, doi: 10.1109/ICDCSW53096.2021.00016.

[24] S. Singh, R. Sulthana, T. Shewale, V. Chamola, A. Benslimane, and B. Sikdar, "Machine-learning-assisted security and privacy provisioning for edge computing: A survey," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 236–260, Jan. 2022.

[25] H. K. Bharadwaj et al., "A review on the role of machine learning in enabling IoT based healthcare applications," *IEEE Access*, vol. 9, pp. 38859–38890, 2021.

[26] V. Hassija, V. Chamola, B. C. Bajpai, and S. Zeadally, "Security issues in implantable medical devices: Fact or fiction?," *Sustain. Cities Soc.*, vol. 66, 2021, Art. no. 102552.

[27] G. A. Kaissis, M. R. Makowski, D. Ráckert, and R. F. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nature Mach. Intell.*, vol. 2, no. 6, pp. 305–311, Jun. 2020.

[28] J. Xu and F. Wang, "Federated learning for healthcare informatics," *J. Healthcare Informat. Res.*, vol. 5, no. 1, 1–19, 2021.

[29] A. Fu, X. Zhang, N. Xiong, Y. Gao, H. Wang, and J. Zhang, "VFL: A verifiable federated learning with privacy-preserving for Big Data in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 3316–3326, May 2022, doi: 10.1109/TII.2020.3036166.

[30] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, "Internet of Things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios," *IEEE Access*, vol. 8, pp. 23022–23040, 2020, doi: 10.1109/ACCESS.2020.2970118.

[31] M. Gheisari, Q. Pham, M. Alazab, X. Zhang, C. Fernández-Campusano, and G. Srivastava, "ECA: An edge computing architecture for privacy-preserving in IoT-based smart city," *IEEE Access*, vol. 7, pp. 155779–155786, 2019, doi: 10.1109/ACCESS.2019.2937177.

[32] X. Xu, C. He, Z. Xu, L. Qi, S. Wan, and M. Z. A. Bhuiyan, "Joint optimization of offloading utility and privacy for edge computing enabled IoT," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2622–2629, Apr. 2020, doi: 10.1109/JIOT.2019.2944007.

[33] J. Li et al., "A federated learning based privacy-preserving smart healthcare system," *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 2021–2031, Mar. 2022, doi: 10.1109/TII.2021.3098010.

[34] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "FedHome: Cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Trans. Mobile Comput.*, vol. 21, no. 8, pp. 2818–2832, Aug. 2022, doi: 10.1109/TMC.2020.3045266.

[35] M. Kim, O. Günlü, and R. F. Schaefer, "Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 2650–2654, doi: 10.1109/ICASSP39728.2021.9413764.

[36] N. Hudson, M. J. Hossain, M. Hosseinzadeh, H. Khamfroush, M. Rahnamay-Naeini, and N. Ghani, "A framework for edge intelligent smart distribution grids via federated learning," in *Proc. Int. Conf. Comput. Commun. Netw.*, 2021, pp. 1–9, doi: 10.1109/ICCCN52240.2021.9522360.

[37] Y. Tu, Y. Ruan, S. Wagle, C. G. Brinton, and C. Joe-Wong, "Network-aware optimization of distributed learning for fog computing," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 2509–2518, doi: 10.1109/INFOCOM41043.2020.9155372.

[38] S. Svorobej et al., "Simulating fog and edge computing scenarios: An overview and research challenges," *Future Internet*, vol. 11, no. 3, 2019, Art. no. 55.

[39] X. Zhang, Y. Wang, S. Lu, L. Liu, L. Xu, and W. Shi, "OpenEI: An open framework for edge intelligence," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst.*, 2019, pp. 1840–1851, doi: 10.1109/ICDCS.2019.00182.

[40] N. Bouacida and P. Mohapatra, "Vulnerabilities in federated learning," *IEEE Access*, vol. 9, pp. 63229–63249, 2021, doi: 10.1109/ACCESS.2021.3075203.

[41] H. Y. Tran and J. Hu, "Privacy-preserving Big Data analytics a comprehensive survey," *J. Parallel Distrib. Comput.*, vol. 134, pp. 207–218, 2019.

[42] S. Desai, R. Alhadad, N. Chilamkurti, and A. Mahmood, "A survey of privacy preserving schemes in IoE enabled smart grid advanced metering infrastructure," *Cluster Computi.*, vol. 22, no. 1, pp. 43–69, 2019.

[43] M. Keshk, B. Turnbull, E. Sitnikova, D. Vatsalan, and N. Moustafa, "Privacy-preserving schemes for safeguarding heterogeneous data sources in cyber-physical systems," *IEEE Access*, vol. 9, pp. 55077–55097, 2021, doi: 10.1109/ACCESS.2021.3069737.

[44] M. Seif, R. Tandon, and M. Li, "Wireless federated learning with local differential privacy," in *Proc. IEEE Int. Symp. Inf. Theory*, 2020, pp. 2604–2609, doi: 10.1109/ISIT44484.2020.9174426.

[45] L. Huang, Y. Yin, Z. Fu, S. Zhang, H. Deng, and D. Liu, "LoAdaBoost: Loss-based AdaBoost federated machine learning with reduced computational complexity on IID and non-IID intensive care data," *PLoS One*, vol. 15, no. 4, 2020, Art. no. e0230706.