Computers in Biology and Medicine 150 (2022) 106019

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

A secure healthcare 5.0 system based on blockchain technology entangled with federated learning technique

Abdur Rehman [a], Sagheer Abbas [a], M.A. Khan [b], Taher M. Ghazal [c,d], Khan Muhammad Adnan [e,*], Amir Mosavi [f,g,h]

[a] School of Computer Science, National College of Business Administration and Economics, Lahore, 54000, Pakistan
[b] Riphah School of Computing and Innovation, Faculty of Computing, Riphah International University, Lahore Campus, Lahore, 54000, Pakistan
[c] School of Information Technology, Skyline University College, University City Sharjah, 1797, Sharjah, United Arab Emirates
[d] Center for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), 43600, Bangi, Selangor, Malaysia
[e] Department of Software, Gachon University, Seongnam, 13120, Republic of Korea
[f] Institute of Information Engineering, Automation and Mathematics, Slovak University of Technology in Bratislava, 81107 Bratislava, Slovakia
[g] John von Neumann Faculty of Informatics, Obuda University, 1034, Budapest, Hungary
[h] Faculty of Civil Engineering, TU-Dresden, 01062, Dresden, Germany

ARTICLE INFO

Keywords:
Federated learning
Blockchain
IoMT
Healthcare 5.0
Medical sensors

ABSTRACT

In recent years, the global Internet of Medical Things (IoMT) industry has evolved at a tremendous speed. Security and privacy are key concerns on the IoMT, owing to the huge scale and deployment of IoMT networks. Machine learning (ML) and blockchain (BC) technologies have significantly enhanced the capabilities and facilities of healthcare 5.0, spawning a new area known as "Smart Healthcare." By identifying concerns early, a smart healthcare system can help avoid long-term damage. This will enhance the quality of life for patients while reducing their stress and healthcare costs. The IoMT enables a range of functionalities in the field of information technology, one of which is smart and interactive health care. However, combining medical data into a single storage location to train a powerful machine learning model raises concerns about privacy, ownership, and compliance with greater concentration. Federated learning (FL) overcomes the preceding difficulties by utilizing a centralized aggregate server to disseminate a global learning model. Simultaneously, the local participant keeps control of patient information, assuring data confidentiality and security. This article conducts a comprehensive analysis of the findings on blockchain technology entangled with federated learning in healthcare. 5.0. The purpose of this study is to construct a secure health monitoring system in healthcare 5.0 by utilizing a blockchain technology and Intrusion Detection System (IDS) to detect any malicious activity in a healthcare network and enables physicians to monitor patients through medical sensors and take necessary measures periodically by predicting diseases. The proposed system demonstrates that the approach is optimized effectively for healthcare monitoring. In contrast, the proposed healthcare 5.0 system entangled with FL Approach achieves 93.22% accuracy for disease prediction, and the proposed RTS-DELM-based secure healthcare 5.0 system achieves 96.18% accuracy for the estimation of intrusion detection.

1. Introduction

The IoMT is an emerging technology that is rapidly expanding [1]. The Internet of Things (IoT) enables the connection of numerous objects to collect data that may be used to enhance human health, productivity, and effectiveness [2–4]. Smart cities, smart grids, and smart houses are well-established concepts that are transforming our daily lives [5–7]. Among the most potential new technical approaches for addressing the global health equality gap is the use of an IoT-based monitoring system of patient health [8]. Another name for these IoT technologies is the IoMT. In this study, the terms IoT and IoMT are used interchangeably, even though the study will focus on the healthcare industry. Furthermore, the Internet of Things has the potential to improve healthcare and public safety significantly [9]. Individuals can obtain information on

* Corresponding author.
E-mail addresses: arbhatti@ncbae.edu.pk (A. Rehman), dr.sagheer@ncbae.edu.pk (S. Abbas), adnan.khan@riphah.edu.pk (M.A. Khan), taher.ghazal@skylineuniversity.ac.ae (T.M. Ghazal), adnan@gachon.ac.kr (K.M. Adnan), amirhosein.mosavi@stuba.sk (A. Mosavi).

https://doi.org/10.1016/j.compbiomed.2022.106019
Received 1 July 2022; Received in revised form 4 August 2022; Accepted 20 August 2022
Available online 21 September 2022

their lifestyles, physical and mental efficiency, and living surroundings, among other things, by connecting their beings to the Internet. This allows healthcare providers to monitor people's health remotely and in real-time. Furthermore, the data gathered could be used to promote evidence-based interventions for diseases, trauma, protection, early detection, and rehabilitation.

In today's world, transferring patients from their homes to hospitals for routine check-ups is extremely difficult. There are several challenges, including queuing, travel time, and the possibility of patients contracting various viruses while traveling through this polluted environment. As a result, the healthcare industry is focusing on in-home healthcare services, which allow patients to conduct medical examinations in the comfort of their own homes. A smart health monitoring system is designed to help patients who live in remote areas contact doctors in urban areas. This technology acts as a bridge between patients and clinicians. It keeps track of vital signs like heart rate, electrocardiogram (ECG), blood pressure, temperature, and whether or not a person has fallen. The system collects this data and sends it to the application for further analysis via a wireless connection.

Effective mining algorithms are urgently needed to analyse medical information to help in disease discovery, offer medical treatment, and enhance patient care. Machine learning is a sophisticated computational technique that has been used in a variety of domains such as image recognition, language processing, and health care [10]. Nonetheless, machine learning models acquire great accuracy only with a vast quantity of training set, which is crucial in healthcare, where precision may sometimes mean the difference between saving or losing a patient's life. In most cases, centralized training strategies include acquiring a big volume of information from a robust cloud server, which might lead to major consumer privacy violations, especially in the medical field.

As an open and accountable data protection mechanism, the development of blockchain technology paves the way for new ways to address key issues of privacy, security, and ethics in a smart healthcare system. However, blockchain has achieved excellent success as a backbone of cybersecurity architecture for a variety of smart healthcare technologies, such as the control of patient record access, information distribution, etc. [11]. It is justifiable to incorporate blockchain in intelligent domestic networks as it is autonomous of heterogeneous protocols that are often used in smart systems [12]. However, notwithstanding increased interest in smart healthcare technology, current work is distributed across diverse fields of study. Such a timely study is also being undertaken to close the void and have practical insights on blockchain technology, techniques, and their application in the field of the intelligent healthcare system.

The IoMT has generated security flaws in the healthcare system. Unified networks enable hackers to reach the linked devices for malicious purposes. The big issue is hack attempts on medical records and healthcare devices; not only are they the key components of a smart healthcare system but they can also be used for malicious purposes: delivering phishing and spam mail. Because of unencrypted wireless keys, smart healthcare devices are often the central target for DDoS attacks, particularly because they are immediately turned on to provide smarter solutions such as patient medical records and the automatic updating of false medical reports [13]. These challenges originate from a centralized IoMT system structure, and privacy issues are growing as the IoMT revolution progresses, namely record forgery and manipulation, device interference, and infiltration of illegal IoMT devices via attacks against server and gateway networks [14].

These issues can be addressed through the deployment of blockchain-based systems and unified "cloud-like" computer networks [15]. Satoshi Nakamoto created the blockchain technology in 2008, which featured a time-stamped collection of harmful proof documents that was maintained by a network of independent networks. Blockchain architecture is illustrated in Fig. 1. It is a series of blocks that are connected with simple cryptography. Inflexibility, decentralization, and transparency are the three main concepts in the functioning of Blockchain technologies. The three responsibilities have been highly productive and opened their doors to a broad array of technologies relevant to the virtual currency, such as the functioning of autonomous vehicles, such as smartphones, and embedded devices. Whereas Blockchain technology is safe and enables anonymity, there are still certain drawbacks to it in the current stage of deployment [16].

FL is a decentralized machine learning platform for the IoT that enables numerous devices to collaboratively learn machine learning models without exchanging their actual data. Fig. 2 depicts FL's architecture. This enhances the intelligent healthcare system by avoiding the leaking of patient information. Fig. 2 shows a federated learning-based healthcare system in which embedded sensors gather medical information from healthcare providers, multiple edge devices collaborate on federated learning algorithms, and machine learning techniques assess the patient's well-being and, if needed, seek immediate assistance in the cloud. Federated learning is a recently popular paradigm due to the incredible assurance it provides for studying fragmented sensitive information. Rather than combining data from disparate sources or relying on the traditional find-then-replicate strategy, it enables training of a common global model on a centralized server while retaining data in the relevant organizations. This method, known as federated learning, allows individual locations to work together to train a global model. Federated learning is the process of pooling training data from multiple sources to develop a global model without directly exchanging datasets. This ensures that patient privacy is maintained across sites.

Healthcare 5.0 [17–19] is a network architecture that requires fifth-generation communication as the core network architecture for linking healthcare equipment. IoT will create data that AI can utilize, contributing to the advancement of digital wellbeing by concentrating not just on patient well-being and quality of life, but also on the

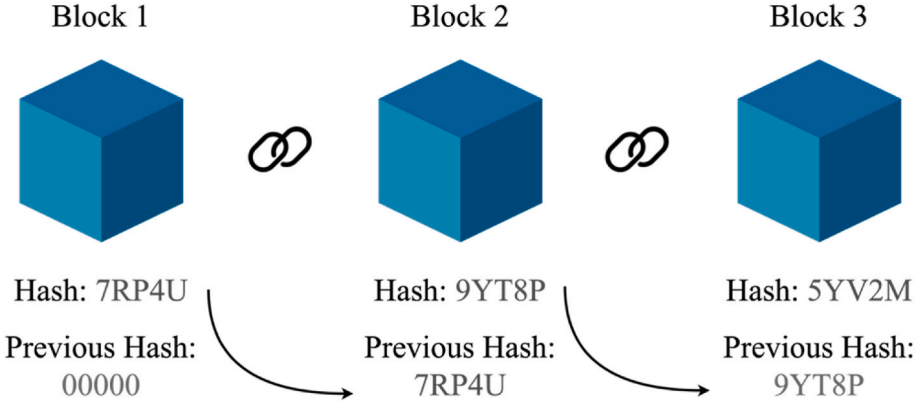


Fig. 1. Blockchain structure.

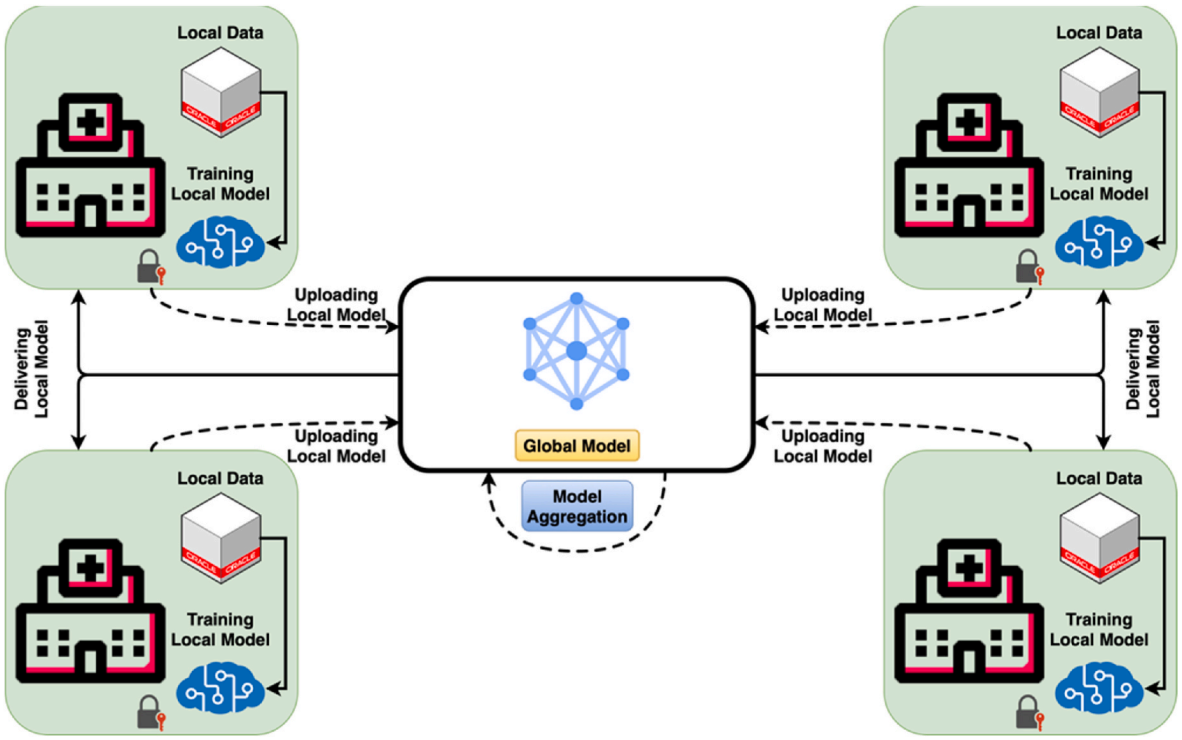


Fig. 2. A federated learning technique.

well-being and quality of life of people worldwide. The primary issues in healthcare 4.0 are flawless information transfer with little or no information leakage. Automation and AI are two emerging healthcare 5.0 technologies that have the potential to transform any type of employment. Intelligence in healthcare 5.0 encompasses ideas such as accurate and systematic disease diagnosis, virtual patient monitoring and detection, remote surgery, and intelligent treatment, that comprises online training for patients suffering from anxiety. Artificial Intelligence (AI) is a broad concept that encompasses all intelligent advancements. It refers to the ability of machine-learning methods to predict final results without the involvement of humans. As a result of the tremendous expansion and advancement of technology in healthcare, the term "smart health system" has been coined. In light of these capabilities, numerous solutions have been proposed for use in numerous industries, including smart homes, Industrial Internet of Things (IoT), and smart healthcare. However, as privacy threats become increasingly sophisticated, there are still a number of obstacles to overcome when implementing blockchain-based FL in healthcare:

1. The parameters of the model kept in the blockchain can still be taken by adversaries to deduce the original confidential clinical data.
2. Some clinical data from medical devices may be falsified to deceive the FL process.
3. There is no motivation for medical devices to provide data and processing power to FL.

To address the aforementioned problems, this study blends FL and sophisticated encryption to provide a safe and privacy-preserving healthcare 5.0 system. The following are the most significant contributions made by this study;

1. We present a blockchain-based FL framework for fifth generation healthcare that not only constructs a correct collaborative model based on various edge devices, but also governs the entire training procedure.
2. The proposed method provides an additional degree of security to blockchain-based FL, we present an RTS-DELM method that balances privacy and model accuracy by adjusting noise depending to the training process.
3. The proposed system considers multiple medical organizations in the proposed model because the locally trained model of other medical organizations may enhance the capability of the healthcare 5.0 system by sharing a global model.
4. To implemented the federated learning approach in healthcare 5.0 to improve the learning process of clinical data by locally training models
5. To provide an intelligent hybrid approach to enhance secure communication and effective healthcare monitoring.
6. We design an Intrusion Detection system (IDS) in healthcare 5.0 system that enhances the security and privacy by detecting intrusions and attack patterns.

The proposed method secures the FL-based healthcare 5.0 Network by accurately assessing its reliability with regard to the most important security goals of secrecy, validity, and accessibility. To support the claim that the overhead caused by the proposed technique is minimal compared to the value of its security and privacy, the study assesses the suggested method's capacity to securely protect sensitive data while consuming quite minimal resources. The purpose of this study is to examine a system model based on a Real-Time Deep Extreme Learning System (RTS-DELM) for intelligent disease prediction and intrusion detection in healthcare 5.0 that achieve the maximum possible level of accuracy.

The remaining portions of this research study can be broken down into the following categories. Section 2 offers an overview of relevant research. Section 3 enlightens the proposed methodology. Section 4 presents the simulation and findings of the suggested method. Finally, section 5 discusses the study's findings.

2. Literature review

The term "blockchain" has been popular among proponents of smart healthcare in recent years, and several research publications have explored how blockchain technology may be applied in the field. S. Aggarwal et al. [20] explored several facets of healthcare, such as the integration of transactions, home healthcare, and investment distribution. The smart home industry has several potential applications for the blockchain. M. Andoni et al. [21] offered a comprehensive analysis of the many blockchain applications of a P2P resource sharing network. The report provides in-depth knowledge on the deployment and abilities of several smart home networks, including smart grid security issues, Big Data analysis, AI and payment services. They concluded that the study did not sufficiently take into consideration challenges associated with smart homes, such as financial planning for smart cities and smart home security.

G. Li et al. [22] suggested a blockchain structure that would be based on users to ensure the safety of information communication in the IoT. Z. Zhou et al. [23] conducted research into various blockchain techniques, predetermined investigation, and decentralized computing to relocation control over particular automobiles and enhance their effectiveness. Du et al. [24] proposed a study with the goals of investigating the implementation of blockchain technology in smart healthcare, developing a centralized conceptual approach for smart healthcare, defining the effect of blockchain on smart healthcare, and eventually, developing a stakeholder-based advancement application framework for smart healthcare. Ihnaini et al. [25] suggested a smart diabetic disease prediction method centered on deep machine learning and information fusion concepts. By combining information, the proposed technique can reduce unnecessary strain on the system's computing resources while also improving the proposed system's efficiency in correctly predicting and recommending this life-threatening condition. Finally, an ensemble machine learning approach is used to create a diabetes prediction model.

Nowadays, data can be easily exchanged across multiple networks, allowing experts and organizations to make the best use of existing capabilities while meeting society's medical demands. Users can gain access to strong and efficient healthcare services thanks to the Internet of Things. The use of smart sensors has aided in the proper monitoring of community healthcare demands. Wearable devices can be used to monitor a wide range of bodily functions. Some can be integrated to monitor various bodily systems to ensure that medical services are delivered to such people in a beneficial manner. The data collected in this manner can be examined, pooled, and mined to perform effective disease prediction [26]. Khan et al. [27] suggested novel healthcare facilities for senior citizens focused on the patients' actual needs and problems. To better satisfy the basic demands of elderly healthcare, the researchers applied machine learning approaches.

Xu et al. [28] presented an overview of federated learning techniques, focusing on those used in biomedicine. Review and explain the broad solutions to the statistical challenges, system challenges, and privacy concerns inherent in federated learning, while emphasizing the implications and potential for healthcare. Li et al. [29] offered an overview of the application of machine learning and bioinformatics technologies in the smart medical business utilizing bibliometric visualization and Web of Science (WOS). A review focuses on the countries that conduct the most research, the primary research subjects, funding sources, and hotspots for research in this field. In addition, the study outlines major difficulties and future research objectives for the use of machine learning and deep learning methods in the healthcare industry.

Siddiqui et al. [30] applied the data fusion technique in a deep learning model to predict breast cancer stages. They applied decision-based fusion to increase the accuracy of the suggested methodology. Medjahed et al. [31] proposed an intelligent healthcare monitoring system based on a data fusion approach. The proposed system is based on a multi-sensor platform that can enable full control over smart homes.

Using AI applications to help diagnose diseases and improve disease prognosis and minimize patient treatment times has become standard practice since the development of artificial intelligence. Dai et al. [32] transformed the challenge of hospitalization forecasting into a supervised classification issue, leading to a significant set of possible medical expense savings. Son et al. [33] used a Support Vector Machine (SVM) algorithm to determine drug compliance in individuals with heart problems. Tariq et al. [34] created a heterogeneous fusion Artificial intelligence-based method to forecast the intensity of COVID-19 using historical medical information. Sedik et al. [35] proposed a methodology for feature extraction to enlarge the information and employed convolutional neural networks and convolutional long short-term memory algorithms to identify coronavirus. Qayyum et al. [36] suggested a clustered FL-based technique for processing medical visual data at the edge, enabling remote infirmaries to capitalize from multimodal information while maintaining their confidentiality. Brisimi et al. [37] forecasted timely treatments for patients with cardiovascular illnesses by utilizing FL to solve dispersed sparse Support Vector Machine difficulties. Nonetheless, the above-mentioned centralized training approaches necessitate the collection of confidential medical information in a unified databank, which is problematic due to data privacy concerns. Rather, federated learning appears as a decentralized architecture that enables cooperative learning while retaining all sensitive information locally, providing a private solution for connecting disparate healthcare data on end devices. Chang et al. [38] propose a blockchain-based federated learning approach for smart healthcare where the MIoT devices apply the federated learning to fully use the dispersed healthcare information and the edge nodes sustain the blockchain to avoid a data loss. In recent years, several research using federated learning in intelligent healthcare have been published as shown in Table 1.

3. Proposed methodology

3.1. Blockchain module implementation

A blockchain was initially designed about and developed by Satoshi Nakamoto in the year 2008 [39]. The block comprises a vast amount of transaction information, including the block id, a hash of the previous block, transaction details, nonce, and time stamps. In a blockchain scheme, where miners determine the correct hash to add a block, the winners would first search the existing block before searching for a new block. The proof of work methodology is used to verify whether or not a given block of transactions is legitimate. The steps listed below describe the core components of blockchain technology. In a smart healthcare system, any node that is connected to the Internet must communicate with a repository of storage data, along with miners in a blockchain framework. The blockchain holds all unprocessed transactions pending an opening to a new block to validate it. Many of the transactions are first reviewed and then shortly analyzed by the Merkle tree. The new smart healthcare system connectivity ecosystem will be created by blockchain technology, as it is versatile and compliant with healthcare IoMT applications.

3.2. RTS-DELM module implementation

RTS-DELM is a data analytics platform that automates data analysis tasks and provides insightful information. A RTS-DELM approach analyses real-time data using the Deep Extreme Learning Machine (DELM). The DELM can be used to determine energy consumption levels, inventory services, and specify transportation operations, to name a few applications [40]. The RTS-DELM method can be used to modify datasets in healthcare networks, implying that any errors can be excluded. It employs a novel approach to disease prediction and diagnosis. The goal of this study is to evaluate a system model based on the RTS-DELM for

Table 1
Comparison of literature with proposed model.

Authors/Objectives	Type of Data	Predictive Model	Decision Making	Fused Decision Making	Healthcare 5.0 Paradigm	Use of Blockchain	Use of IDS	Use of FL
Du et al. [24]	Medical Records	Yes	Yes	No	No	Yes	No	No
Medjahed et al. [31]	Medical Records	Yes	Yes	Yes	No	No	No	No
Ihnaini et al. [25]	Medical Records	Yes	Yes	Yes	No	No	No	No
Khan et al. [27]	Medical Records	Yes	Yes	No	No	No	No	No
Li et al. [29]	Medical Records	No	No	No	No	Yes	No	No
Siddiqui et al. [30]	Medical Records	Yes	Yes	Yes	No	No	No	No
Chang et al. [38]	Medical Records	Yes	Yes	No	No	Yes	No	Yes
Proposed Model	**Sensor data**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

the most accurate adaptive forecasting of healthcare monitoring.

3.3. Federated learning module implementation

Federated learning is a recently popular paradigm due to the incredible assurance it provides for learning with fragmented sensitive data. Rather than combining data from multiple sources or relying on the traditional find-then-replicate strategy, it allows for the training of a common global model on a central server while retaining data in the appropriate organizations. Federated learning, as it is known, allows individual sites to contribute to the training of a global model. Federated learning is the process of combining training data from multiple sources to build a global model without directly sharing datasets. This ensures that patient privacy is preserved across locations.

Several firms or institutions collaborate to solve a machine-learning problem using federated learning, which is managed by a central server or service provider. As a result, a deep learning model is hosted and optimized by a central server. The model is trained by dispersing itself across remote centralized datacentres, which may include hospitals or other medical organizations, preserving data localization at these locations. Throughout the training process, no data from any participant is exchanged or transferred. Rather than sending data to a single server, as in traditional deep learning, the server maintains a globally shared architecture that is accessible to all institutions. Following that, each organization develops its model based on patient data. Following that, each institute communicates with the server using the error gradient of the model. The central server collects all participant feedback and modifies the global model according to predefined criteria. The predefined criteria allow the model to evaluate the excellence of the response and thus include only information that adds value. As a result, feedback from institutions reporting poor or unusual results may be overlooked. This method creates a single federated learning cycle that is repeated until the global model is acquired.

3.4. Data fusion module implementation

Data fusion techniques combine information from multiple sensors to produce more precise observations than a single, independent sensor could. The process of obtaining data from disparate but likely linked sources and combining it to have the greatest impact is known as information extraction. A security framework includes several network security sensors, and it is challenging to gain a comprehensive, wide-angle perspective of the security system's dynamic security situation. Additionally, equipment spread across a vast region may be challenging to handle effectively. In order to enhance the model's efficacy and give a comprehensive analysis of the system's protection situation, it is also essential to efficiently and smartly combine the sensor data. In comparison, because the information comes from multiple sources, multiple data sources can provide a higher level of consistency in terms of trustworthiness.

Sensors contribute to the Internet of Things by serving as critical tools for evaluating smart healthcare systems, ecosystems, and consumers. The following devices fall under this category: Healthcare, Cameras, and Interactivity are just a few of the terms that come to mind when thinking about Computers processing the data collected by sensors. For example, in conjunction with the healthcare system, the heart monitor sensor regulates the cardiac rate. The application layer is a component of the IoMT network system that includes closed-circuit devices, wearables, and so on [41].

3.5. Intrusion detection system (IDS) module implementation

A comprehensive intrusion detection system is necessary to fully assess the healthcare system. Any type of data can be analyzed using the machine learning method termed RTS-DELM [42].

The autonomous dataflow architecture used by this machine learning software allows it to track the information flow and identify trends of infiltration and assault. It's crucial to develop robust and adaptable algorithms to manage the continuously evolving smart blockchain-based applications.

The suggested strategy provides a safe smart healthcare network design that would solve both present issues with centralized security of healthcare networks and potential future threats. In the work described here, a RTS-DELM technique will be employed to create a smarter, safer healthcare system employing sensors driven by the IoMT that are more effective as shown in Fig. 3. The main contributions of this study are a thorough analysis of scientific advancements feasible to blockchain-based healthcare 5.0 systems powered by RTS-DELM, a fresh perspective on various employments (like data exchange between healthcare systems), and assistance from the most recent stages of technological development [43].

3.6. Dataset description

For disease prediction analysis, we employed the publicly accessible Parkinson's disease dataset [44], and for intrusion detection, we employed the publicly available NSL-KDD dataset [45]. Fig. 4 lists all protection protocols falling under their identity group. The NSL-KDD dataset is a refined edition of KDD 99 that includes several enhancements related to the original KDD 99 data collection. The NSL-KDD data collection contains 41 functions for each record. Fig. 4 offers a proper description of the functions. This study includes 195 prolonged vowel phonations from 31 individuals, 23 of whom were diagnosed with Parkinson's disease. The primary objective of data processing is to

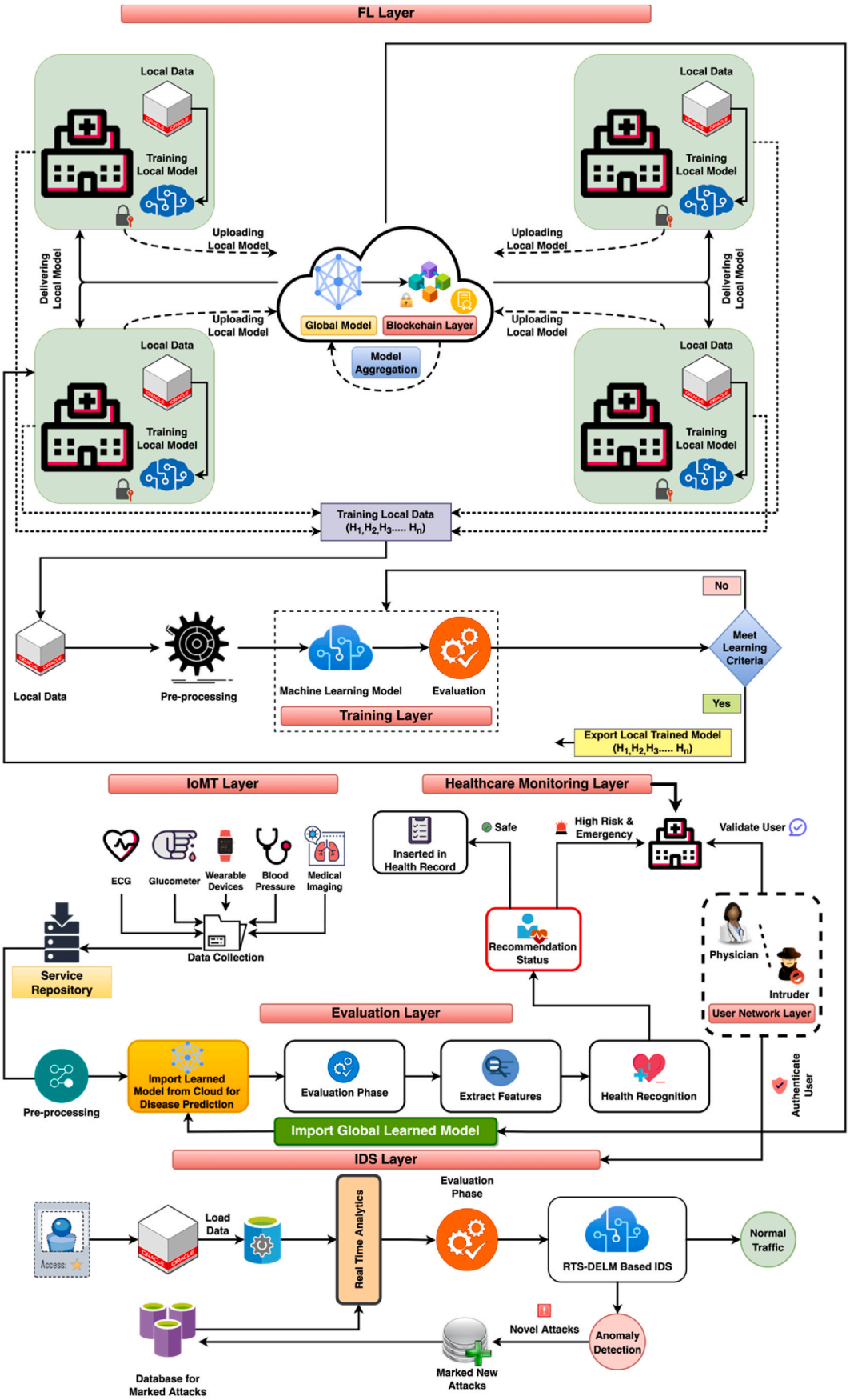


Fig. 3. A proposed federated learning-based healthcare monitoring system for healthcare 5.0.

No	Features	Form of value	No	Features	Form of value
1	Duration	Integer	22	is_guest_login	Integer
2	protocol_type	Nominal	23	count	Integer
3	Service	Nominal	24	srv_count	Integer
4	Flag	Nominal	25	serror_rate	Float
5	src_bytes	Integer	26	srv_serror_rate	Float
6	dst_bytes	Integer	27	rerror _rate	Float
7	land	Integer	28	srv_rerror_rate	Float
8	wrong_fragment	Integer	29	same_srv_rate	Float
9	urgent	Integer	30	diff _srv_rate	Float
10	hot	Integer	31	srv_diff_host_rate	Float
11	num_failed_logins	Integer	32	dst_host_count	Float
12	root_shell	Integer	33	dst_host_srv_count	Float
13	num_compromised	Integer	34	dst_host_same_srv_rate	Float
14	roots_hell	Integer	35	dst_host_diff_srv_rate	Float
15	su_attempted	Integer	36	dst_host_same_src_port_rate	Float
16	num_root	Integer	37	dst_host_srv_diff_port_rate	Float
17	num_file_creations	Integer	38	ddst_host _serror_rate	Float
18	num_shells	Integer	39	dst-_host_srv_serror_rate	Float
19	num_access_files	Integer	40	dst_host_rerror_rate	Float
20	num_outbound_cmds	Integer	41	dst_host_srv_rerror_rate	Float
21	Is_host_login	Integer			

Fig. 4. NSL-KDD dataset structure.

distinguish between healthy individuals and those with Parkinson's disease, based on the "status" attribute, which is set to non-PD for healthy individuals and PD for those with Parkinson's disease; this is a two-decision classification issue. Fig. 5 depicts the attributes of the data set.

3.7. Working of the proposed model

RTS-DELM computational technology may be used to make federated learning-based systems more intelligent. When using the RTS-DELM distributed blockchain technology, it is possible to increase the level of data secrecy. Additionally, by transferring new information and accelerating comprehension, RTS-DELM may be utilized to enhance understanding [30]. It provides the platform and network architecture for the development of a decentralized blockchain application [43]. The deployment architecture of the advanced system RTS-DELM is examined in this paper. Utilizing sensors, mobile devices, and IoMT systems as sources of information are the right way to employ this technology to gather intelligence. These strategies produce knowledge that is employed in smart applications. Despite this, real-time data analysis uses the RTS-DELM approach to assess and make predictions [42].

The creation of data for study reduces data mistakes such as repetition, missing information factors, malfunctions, and interference. When a little fraction of a data collection is all that is needed, the RTS-DELM method performs effectively. The architecture is capable of supporting a broad variety of applications in several fields, including fraud detection and/or prevention. As shown in Fig. 3, the proposed RTS-DELM system employs a large number of hidden layers, hidden neurons, and various activating mechanisms to optimize the healthcare monitoring system. Data capture, preparation, and assessment are the three separate phases that make up the suggested approach for breaking down data analysis. Prediction and performance make up the two sub-layers that make up the evaluation layer. Accurate data is gathered from sensors and actuators for analysis. The collecting layer uses the data that is provided as raw data. To eliminate inconsistencies in the preprocessing layer, a thorough approach for data cleaning and preparation is used.

The foremost objectives of this approach are:

- To implement the RTS-DELM algorithm to build an optimal approach for healthcare monitoring at the client side.
- In this study, an intelligent algorithm is proposed to identify and monitor healthcare in patients and identify the severity of the patients.
- To implement an Intrusion Detection System (IDS) to determine the flow of data to detect intrusions and attack patterns

No	Features	No	Features	No	Features
1	MDVP:Fo(Hz)	9	MDVP:Shimmer	17	RPDE
2	MDVP:Fhi(Hz)	10	MDVP:Shimmer(dB)	18	DFA
3	MDVP:Flo(Hz)	11	Shimmer: APQ3	19	spreadi
4	MDVP:Jitter(%)	12	Shimmer: APQ5	20	spread2
5	MDVP:Jitter(Abs)	13	MDVP:APO	21	D2
6	MDVP:RAP	14	Shimmer:DDA	22	PPE
7	MDVP:PPO	15	NHR	23	Class Label
8	Jitter: DDP	16	HNR		

Fig. 5. Parkinson's disease dataset structure.

- To implement and improve IoMT in existing applications to sustain the medical standards. Furthermore, to increase the model's efficiency, a federated learning approach is implemented as shown in Table 2 & Table 3. Further evaluation is carried out using disease datasets and the freshly trained layer is applied to the pre-trained framework to increase the efficacy of the system.
- To preserve records of the local data to preserve the patient's privacy.
- To test the utility of a trained model by assessing it on a real-time dataset. with federated learning, the implementation of the prediction system is evaluated. In addition, model output is evaluated with other models of machine learning.
- To prevent humans from getting chronic diseases by the spreading of the disease.
- To estimate the performance of the proposed methodology by assessing different datasets.
- Furthermore, test and evaluate the datasets on different machine learning algorithms to justify the proposed approach.

The study will provide a thorough analysis of technical advancements relevant to RTS-DELM-enabled federated learning-based systems to provide a fresh perspective on various implementations (such as healthcare data exchange). The main aim of this study is to build an intelligent algorithm proposed for healthcare monitoring among patients.

The working of the proposed model is described as follows;

- Hospitals identify and assign training tasks to a local model, which is subsequently uploaded to a centralized server, where it is disseminated to all IoMT devices as a global model.
- Additionally, the training phase comprises three layers: the sensing layer, the preprocessing layer, and the application layer.
- Medical data obtained by IoMT devices may contain missing or erroneous data.
- To reduce noisy data, the preprocessing layer uses moving average and normalization to address missing values.
- Clinical data is transmitted to the Application layer following preparation. Additionally, the Application Layer is separated into two sections: Prediction Layer and Performance Layer.

Table 2
Proposed RTS-DELM Entangled with Federated Learning Pseudo code (Server-side).

[Server Side]	
Sr No.	Steps
1	Start
2	Initialize $w^k_{G,fml}$ & $v^k_{G,fml}$ **Where** $w^k_{G,fml}$ & $v^k_{G,fml}$ represents the weight between input and y hidden layers at server side and the weight between y hidden layers and $y+1$ hidden layers neurons at server side respectively.
3	**for** each cycle k from I to K **do**
4	$S_k \leftarrow$ (Random set of clients from η) a) **for** each client $l \in S_k$ parallelly **do** $[w^n_{k+1}, v^n_{k+1}] \leftarrow$ **Client Training** (n, w_k, v_k) **end for**
5	$w^k_{G,fml} = \frac{1}{\Sigma_{n\eta}} \Sigma^N_{n=1} \frac{S_n}{S} w^k_{n+1}$ (Avg Aggregation)
6.	$v^k_{G,fml} = \frac{1}{\Sigma_{n\eta}} \Sigma^N_{n=1} \frac{S_n}{S} v^k_{n+1}$
7.	end for
8.	Stop

Put the optimum weights values of $w^k_{G,fml}$ & $v^k_{G,fml}$ in Equations (1) and (2), and predict the values for lung cancer found or not.

Table 3
Proposed Smart FML- RTS-DELM Pseudo code (Client-side).

[Client Training (L, w, v)]	
Sr No.	Steps
1	Start
2	Split local data to mini batches of size S
3	Initialization of both layer weights (ω_{ij} & υ_{jk}), Error (E) = 0 and the number of epochs £ = 0
4	For each training pattern p b) do the feedforward phase to i) calculate φ_j using eq (1) ii) calculate φ_k using eq (2) c) Calculate error signals for the output and the hidden layer d) Then equalize the weights υ_{jk} and ω_{ij} (backpropagation of errors) using eq (8) & eq (9).
5	£ = £ + 1
6.	Test stopping criteria: if no stopping criterion is satisfied, go to step 4.
7.	Return optimum local trained model weights ω_{ij} and υ_{jk} to Server
8.	Stop

- Following the prediction layer, the results of the forecast layer are transmitted to the performance layer, which uses the accuracy and miss rate of the prediction layer to determine whether or not the learning requirements are met.
- In the case of 'No' the model will be retrained but in the case of 'Yes', the locally trained model is exported to the cloud as a global model.
- Medical organizations receive a copy of a global model and train it on local data. Following that, each institution upgrades its model on the cloud without exchanging datasets directly. This assures the continuity of patient privacy between medical organizations.
- The cloud aggregates all updated parameters provided by participating institutions to create a new global model, which is subsequently disseminated to all participating organizations.
- As opposed to outcomes produced from a single information source, the data fusion technique can give results that are more dependable and stable.
- Then the data requester launches a data sharing request to the centralized server. Upon receiving the request, the centralized server verifies the access. After authorizing access, the nodes train a global data model jointly by federated learning. Once the model is trained, the data requester gets the corresponding sharing results.
- The input layer parameters will be detected during the validation phase and forwarded to the evaluation phase for healthcare monitoring.

Each client Proposed RTS-DELM has used input layer, six hidden layers and an output layer as exhibited in Table 3. The backpropagation algorithm has several phases, including initialization of weight, feedforward, backpropagation of error, and updating of weight and bias, as indicated in the table above. Each neuron in the hidden layer is equipped with a Sigmoid activation function. The proposed system based on RTS-DELM may be stated [40];

$$\varphi_{j^l} = \frac{1}{1 + e^{-\left(b_y + \sum_{i=1}^{m} (\omega_{ij} * r_i)\right)}} \; where j = 1, 2, 3 \ldots n \tag{1}$$

where r_i input data, b_1 is bias, m represents total number of input neurons and j represents total number of hidden layer neurons.

Output layer activation function is given below [40];

$$\varphi_k = \frac{1}{1 + e^{-\left(b_2 + \sum_{j=1}^{n} (v_{jky} * \varphi_j)\right)}} \; where k = 1, 2, 3 \ldots r \tag{2}$$

where y represents hidden layers [41];

$$E = \frac{1}{2} \sum_{k} (\tau_k - \varphi_{k^{y=6}})^2 \tag{3}$$

Above equation E represents backpropagation error where, τ_k & φ_k symbolize the anticipated output and projected output.

In equation (4), the weight of the output changes at a constant pace [41], the layer is composed as;

$$\Delta W \propto -\frac{\partial E}{\partial W}$$
$$\Delta v_{j,k^y} = -\varepsilon \frac{\partial E}{\partial \nu_{j,k^y}} \tag{4}$$

After employing the Chain rule technique above eq can be composed as [42];

$$\Delta v_{j,k^y} = -\varepsilon \frac{\partial E}{\partial \varphi_{k^y}} \times \frac{\partial \varphi_{k^y}}{\partial \nu_{j,k^y}} \tag{5}$$

After substituting the values in equation (5), the value of weight changed can be obtained as shown in equation (6) [40,42].

$$\Delta v_{j,k^y} = \varepsilon(\tau_k - \varphi_{k^y}) \times \varphi_{k^y}(1-\varphi_{k^y}) \times (\varphi_j)$$
$$\Delta v_{j,k^y} = \varepsilon \xi_k \varphi_j \tag{6}$$

where,

$$\xi_k = (\tau_k - \varphi_{k^y}) \times \varphi_{k^y}(1-\varphi_{k^y})$$

Utilize the chain rule to maintain the weights between the input and hidden layers [41].

$$\Delta \omega_{i,j} \propto -\left[\sum_k \frac{\partial E}{\partial \varphi_{k^y}} \times \frac{\partial \varphi_{k^y}}{\partial \varphi_j}\right] \times \frac{\partial \varphi_j}{\partial \omega_{i,j}}$$

$$\Delta \omega_{i,j} = -\varepsilon \left[\sum_k \frac{\partial E}{\partial \varphi_{k^y}} \times \frac{\partial \varphi_{k^y}}{\partial \varphi_j}\right] \times \frac{\partial \varphi_j}{\partial \omega_{i,j}}$$

In the above eq, ε symbolizes the constant [41],

$$\Delta \omega_{i,j} = \varepsilon \left[\sum_k (\tau_k - \varphi_{k^y}) \times \varphi_{k^y}(1-\varphi_{k^y}) \times (\nu_{j,k})\right] \times \varphi_{k^y}(1-\varphi_{k^y}) \times \alpha_i$$

$$\Delta \omega_{i,j} = \varepsilon \left[\sum_k (\tau_k - \varphi_{k^y}) \times \varphi_{k^y}(1-\varphi_{k^y}) \times (\nu_{j,k})\right] \times \varphi_j(1-\varphi_j) \times \alpha_i$$

$$\Delta \omega_{i,j} = \varepsilon \left[\sum_k \xi_k (\nu_{j,k^y})\right] \times \varphi_j(1-\varphi_j) \times \alpha_i$$

After simplifying the preceding equation, it can be expressed as [42].

$$\Delta \omega_{i,j} = \varepsilon \xi_j \alpha_i \tag{7}$$

where,

$$\xi_j = \left[\sum_k \xi_k (\nu_{j,k^y})\right] \times \varphi_j(1-\varphi_j)$$

$$\nu^+_{j,k^y} = \nu_{j,k^y} + \lambda \, \Delta v_{j,k^y} \tag{8}$$

where λ represents the learning rate.

The above equation is used for updating the weights between output and hidden layers and the below equation is used for updating the weights between input and hidden layers [40–42].

$$w^+_{i,k} = w_{i,j} + \lambda \, \Delta w_{i,j} \tag{9}$$

4. Simulation results

In this research, the healthcare 5.0 system entangled with federated learning approach was applied to the Parkinson's disease dataset [44] for Parkinson's disease prediction and the NSL-KDD dataset [45] for detecting any intrusion activity in a system. The datasets are randomly divided into 70% of training and 30% of data is used for validation and testing. The data is investigated in diagnosing Parkinson's disease and detecting intrusion activity in a system. To estimate the efficacy of the proposed method, numerous numerical metrics like Miss rate, accuracy, specificity, sensitivity, True Positive Rate (TRP), True Negative Rate (TNR), Positive Prediction Value (PPV) & Negative Prediction Value (NPV) were employed listed as follows [40–42];

$$\text{Miss rate} = \frac{\sum_{b=0}^{2}\left(Q_b/S_{z\neq b}\right)}{\sum_{b=0}^{2}(T_b)}, \; Where \; z = 0, 1 \tag{10}$$

$$\text{Accuracy} = \frac{\sum_{b=0}^{2}\left(Q_b/S_b\right)}{\sum_{b=0}^{2}(Q_b)} \tag{11}$$

$$\text{Sepecificity} = \frac{Q_0/S_0}{\left(Q_0/S_0 + Q_0/S_1\right)} \tag{12}$$

$$\text{Sensitivity} = \frac{Q_1/V_1}{\left(Q_1/S_0 + Q_1/S_1\right)} \tag{13}$$

$$\text{TPR} = \frac{TP}{(TP+FN)} \tag{14}$$

$$\text{TNR} = \frac{TN}{(TN+FP)} \tag{15}$$

$$\text{PPV} = \frac{TP}{(TP+FP)} \tag{16}$$

$$\text{NPV} = \frac{TN}{(FN+TN)} \tag{17}$$

In Eqs. (10) and (11), Q signifies the predictive output and S symbolizes the actual output. Q_0 and S_0 represents that there is no Parkinson's disease and there is no intrusion activity in predictive output and actual output correspondingly. Q_1 and S_1 signifies there is a Parkinson's disease and intrusion activity in the predictive result and actual result individually. Q_b/S_b denotes predictive and actual results are parallel. Correspondingly, $Q\_b/S\_(z \neq b)$ epitomizes error, whereby outcomes, both predictive and actual, are altered.

4.1. Intrusion detection system (IDS)

Table 4 exhibited the suggested RTS-DELM-based secure healthcare 5.0 system for the prediction of intrusion in the system during the training level. Through the training phase, a total of 400 records are applied at each client side (H_1, H_2, H_3, H_4), which are technically separated into normal and attack records, respectively. It is seen that the forecasting system successfully achieve maximum accuracy for forecast the intrusion in a system at each client side. Table 4 shows the different statistical measures during training level at each client side. As shown in Table 4, proposed system is effective in terms of accuracy to determine the intrusion at each client side. In addition, some other statistical measures are calculated during training level. As we can see that in Table 4, H_1 client gives the 93.75% accuracy, 98.25% sensitivity, 82.61% specificity, 95% negative predictive value, 17.39% false positive rate, 6.67% false discovery rate and 1.75% false negative rate. While H_2 client gives the 94.72% accuracy, 98.95% sensitivity, 84.07% specificity, 96.94% negative predictive value, 15.93% false positive rate, 6% false discovery rate and 1.05% false negative rate. While H_3 client gives the 97.75% accuracy, 98.99% sensitivity, 94.17% specificity, 97% negative predictive value, 5.83% false positive rate, 2% false discovery rate and 1.01% false negative rate. While H_4 client gives the 95.72% accuracy, 99.30% sensitivity, 86.36% specificity, 97.94% negative

Table 4
Performance evaluation of proposed RTS-DELM-based secure healthcare 5.0 system during training for the estimation of intrusion detection in a system using different statistical measurements at client side.

Client	Accuracy	Sensitivity	Specificity	Negative Predictive Value	False Positive Rate	False discovery Rate	False Negative Rate
H_1	0.9375	0.9825	0.8261	0.9500	0.1739	0.0667	0.0175
H_2	0.9472	0.9895	0.8407	0.9694	0.1593	0.0600	0.0105
H_3	0.9775	0.9899	0.9417	0.9700	0.0583	0.0200	0.0101
H_4	0.9572	0.9930	0.8636	0.9794	0.1364	0.0500	0.0070

predictive value, 13.64% false positive rate, 5% false discovery rate and 0.7% false negative rate.

Table 5 exhibited the suggested RTS-DELM-based secure healthcare 5.0 system for the prediction of intrusion in the system during the validation level. Through the validation phase, a total of 200 records are applied at each client side (H_1, H_2, H_3, H_4), which are technically separated into normal and attack records, respectively. It is seen that the detection system successfully achieves maximum accuracy for forecast the intrusion in a system at each client side. Table 5 shows the different statistical measures during validation level at each client side. As shown in Table 5, proposed system is effective in terms of accuracy to determine the intrusion at each client side. In addition, some other statistical measures are calculated also during validation level to justify the effectiveness of the proposed system. As we can see that in Table 5, H_1 client gives the 95% accuracy, 97.30% sensitivity, 88.64% specificity, 92% negative predictive value, 11.54% false positive rate, 4% false discovery rate and 2.7% false negative rate. While H_2 client gives the 93.50% accuracy, 95.75% sensitivity, 86.27% specificity, 88% negative predictive value, 13.73% false positive rate, 4.67% false discovery rate and 4.03% false negative rate. While H_3 client gives the 96.50% accuracy, 98.64% sensitivity, 90.57% specificity, 96% negative predictive value, 9.43% false positive rate, 3.33% false discovery rate and 1.36% false negative rate. While H_4 client gives the 94.50% accuracy, 96.64% sensitivity, 88.24% specificity, 90% negative predictive value, 11.76% false positive rate, % false discovery rate and 3.36% false negative rate.

4.2. Disease prediction

Table 6 exhibited the suggested healthcare 5.0 system entangled with federated learning approach for the prediction of Parkinson's disease in the patients during the validation level. Each client side (H_1, H_2, H_3, H_4) at training level trained their model on their local data and exported the learned model to cloud to form a global learned model. Furthermore, the global learned model imported from blockchain centered centralized server to validate the proposed system after verifying access through IDS. Through the validation phase, a total of 200 records are applied at server side, which are technically separated into negative and positive samples, respectively. It is seen that the forecasting system successfully achieve maximum accuracy for predict the disease in a patient at server side. Table 6 shows the different statistical measures during validation level at each client side and server side utilizing FL approach to justify the proposed healthcare 5.0 system entangled with federated learning approach in terms of accuracy. As presented in Table 6, proposed healthcare 5.0 system entangled with federated learning system is effective in terms of accuracy to predict the disease. In addition, some other statistical measures are calculated during validation level. As we can see that in Table 6, H_1 client gives the 92.50% accuracy, 96.97% sensitivity, 71.43% specificity, 83.33% negative predictive value, 28.57% false positive rate, 5.88% false discovery rate and 3.03% false negative rate during validation level. While H_2 client gives the 93% accuracy, 96.43% sensitivity, 75% specificity, 80% negative predictive value, 25% false positive rate, 4.71% false discovery rate and 3.57% false negative rate during validation level. While H_3 client gives the 95.50% accuracy, 97.63% sensitivity, 83.87% specificity, 86.67% negative predictive value, 16.63% false positive rate, 2.94% false discovery rate and 2.37% false negative rate during validation level. While H_4 client gives the 94.50% accuracy, 96.49% sensitivity, 82.76% specificity, 80% negative predictive value, 17.24% false positive rate, 2.94% false discovery rate and 3.51% false negative rate during validation level. Finally, at server side the proposed healthcare 5.0 system entangled with federated learning approach achieves the maximum accuracy as compare to each client. The proposed FL approach gives the 97% accuracy, 98.24% sensitivity, 90% specificity, 90% negative predictive value, 10% false positive rate, 1.76% false discovery rate and 1.76% false negative rate during validation level.

5. Discussion

Table 7 shows the comparison of the proposed model with previously published research. Chang et al. [38] achieves 84.5% accuracy using CNN. Sheibani et al. [46] achieves 90.6% accuracy using 10-fold cross-validation. Tracy et al. [47] achieves 90.1% accuracy using L2-regularized logistic regression, random forest. Sztahó et al. [48] achieves 89.3% accuracy using KNN, SVM-linear, SVM-RBF, ANN and DNN. Yaman et al. [49] achieves 91.25% accuracy using KNN, SVM with 10-fold cross-validation. Kuresan et al. [50] achieves 95.16% accuracy using HMM, SVM. While proposed healthcare 5.0 system entangled with federated learning approach model achieves 97% accuracy. As shown in Table 7, the proposed approach outclasses other approaches in terms of accuracy.

6. Conclusion

Smart healthcare based on IoMT has been extensively used in recent times to fully utilize clinical data to enhance the precision of disease prediction and medical treatment. With the increasing volume and variety of clinical data, there is a pressing need for effective mining methods to evaluate this data to aid disease diagnosis, give medical remedies, and enhance patient care. Nonetheless, it confronts several difficulties, including patient privacy breaches and a variety of adversary threats to data transmission. Additionally, the advancement of artificial intelligence and global epidemics have accelerated the adoption of smart healthcare, while introducing concerns about data privacy,

Table 5
Performance evaluation of proposed RTS-DELM-based secure healthcare 5.0 system during validation for the estimation of intrusion detection in a system using different statistical measurements at client side.

Client	Accuracy	Sensitivity	Specificity	Negative Predictive Value	False Positive Rate	False discovery Rate	False Negative Rate
H_1	0.9500	0.9730	0.8846	0.9200	0.1154	0.0400	0.0270
H_2	0.9350	0.9597	0.8627	0.8800	0.1373	0.0467	0.0403
H_3	0.9650	0.9864	0.9057	0.9600	0.0943	0.0333	0.0136
H_4	0.9450	0.9664	0.8824	0.9000	0.1176	0.0400	0.0336

Table 6
Performance evaluation of proposed healthcare 5.0 system entangled with federated learning approach during validation for the prediction of disease in a patient using different statistical measurements at server side.

Client	Accuracy	Sensitivity	Specificity	Negative Predictive Value	False Positive Rate	False discovery Rate	False Negative Rate
H_1	0.9250	0.9697	0.7143	0.8333	0.2857	0.0588	0.0303
H_2	0.9300	0.9643	0.7500	0.8000	0.2500	0.0471	0.0357
H_3	0.9550	0.9763	0.8387	0.8667	0.1613	0.0294	0.0237
H_4	0.9450	0.9649	0.8276	0.8000	0.1724	0.0294	0.0351
Proposed Approach based on FL (Server Side)	**0.9700**	**0.9824**	**0.9000**	**0.9000**	**0.1000**	**0.0176**	**0.0176**

Table 7
Performance evaluation of proposed RTS-DELM Based Architecture for the Estimation of Parkinson's Disease with previously published research.

	Machine Learning Methods	Outcome
Chang et al. [38]	Convolutional Neural Network)CNN)	84.5%
Sheibani et al. [46]	Ensemble learning with 10-fold cross-validation	90.6%
Tracy et al. [47]	L2-regularized logistic regression, random forest	90.1%
Sztahó et al. [48]	KNN, SVM-linear, SVM-RBF, ANN, DNN	89.3%
Yaman et al. [49]	KNN, SVM with 10-fold cross-validation	91.25%
Kuresan et al. [50]	HMM, SVM	95.16%
Proposed Model	**Healthcare 5.0 System Entangled with Federated Learning Approach**	**97%**

malicious cyberattack, and quality of service.

A secure healthcare 5.0 system based on blockchain technology entangled with federated learning techniques is being explored to increase predictive performance. Numerous analytical frameworks have been employed to determine the viability of this particular argument. The proposed RTS-DELM method is extremely effective. The proposed technique has a rate of accuracy of 93.22% for disease prediction and 96.18% for intrusion detection. Furthermore, it is acknowledged that developing a basic method is less expensive and faster. We are pleased with the preliminary findings and intend to expand our research by evaluating additional sets of data in the future. The proposed system's computational complexity is limited by the growing number of hidden layers. Future research will aim to identify and quantify the factors with greater precision. To optimize the performance of various configurations, the learning system will be retrained on a more frequent basis.

Declaration of competing interest

It declares there is no conflict of interest.

References

[1] C. Wilson, T. Hargreaves, R. Hauxwell-Baldwin, Benefits and risks of smart home technologies, Energy Pol. 103 (2017) 72–83, https://doi.org/10.1016/j.enpol.2016.12.047.

[2] B.L. Risteska Stojkoska, K.V. Trivodaliev, A review of Internet of Things for smart home: challenges and solutions, J. Clean. Prod. 140 (2017) 1454–1464, https://doi.org/10.1016/j.jclepro.2016.10.006.

[3] F. Folianto, Y.S. Low, W.L. Yeow, Smartbin: smart waste management system, in: 2015 IEEE 10th Int. Conf. Intell. Sensors, Sens. Networks Inf. Process, 2015, https://doi.org/10.1109/ISSNIP.2015.7106974. ISSNIP 2015.

[4] J. ho Park, M.M. Salim, J.H. Jo, J.C.S. Sicato, S. Rathore, J.H. Park, CIoT-Net: a scalable cognitive IoT based smart city network architecture, Human-Centric Comput, Inf. Sci. 9 (2019) 1–20, https://doi.org/10.1186/s13673-019-0190-9.

[5] M.R. Alam, M. St-Hilaire, T. Kunz, Peer-to-peer energy trading among smart homes, Appl. Energy 238 (2019) 1434–1443, https://doi.org/10.1016/j.apenergy.2019.01.091.

[6] Y. Mittal, P. Toshniwal, S. Sharma, D. Singhal, R. Gupta, V.K. Mittal, A voice-controlled multi-functional smart home automation system, in: 12th IEEE Int. Conf. Electron. Energy, Environ. Commun. Comput. Control (E3-C3), 2016, https://doi.org/10.1109/INDICON.2015.7443538. INDICON 2015.

[7] P. Wang, F. Ye, X. Chen, A smart home gateway platform for data collection and awareness, IEEE Commun. Mag. 56 (2018) 87–93, https://doi.org/10.1109/MCOM.2018.1701217.

[8] J. Shen, C. Wang, T. Li, X. Chen, X. Huang, Z.H. Zhan, Secure data uploading scheme for a smart home system, Inf. Sci. 453 (2018) 186–197, https://doi.org/10.1016/j.ins.2018.04.048.

[9] N. Komninos, E. Philippou, A. Pitsillides, Survey in smart grid and smart home security: issues, challenges and countermeasures, IEEE Commun. Surv. Tutorials. 16 (2014) 1933–1954, https://doi.org/10.1109/COMST.2014.2320093.

[10] S. Abbas, M.A. Khan, L.E. Falcon-Morales, A. Rehman, Y. Saeed, M. Zareei, A. Zeb, E.M. Mohamed, Modeling, simulation and optimization of power plant energy sustainability for IoT enabled smart cities empowered with deep extreme learning machine, IEEE Access 8 (2020) 39982–39997, https://doi.org/10.1109/ACCESS.2020.2976452.

[11] D. Nasonov, A.A. Visheratin, A. Boukhanovsky, Blockchain-based transaction integrity in distributed big data marketplace, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 2018, https://doi.org/10.1007/978-3-319-93698-7_43.

[12] R.A. Michelin, A. Dorri, M. Steger, R.C. Lunardi, S.S. Kanhere, R. Jurdak, A. F. Zorzo, Speedy Chain: a framework for decoupling data from blockchain for smart cities, in: ACM Int. Conf. Proceeding Ser., 2018, https://doi.org/10.1145/3286978.3287019.

[13] B. Xiong, K. Yang, J. Zhao, K. Li, Robust dynamic network traffic partitioning against malicious attacks, J. Netw. Comput. Appl. 87 (2017) 20–31, https://doi.org/10.1016/j.jnca.2016.04.013.

[14] C. Yin, J. Xi, R. Sun, J. Wang, Location privacy protection based on differential privacy strategy for big data in industrial internet of things, IEEE Trans. Ind. Inf. 14 (2018) 3628–3636, https://doi.org/10.1109/TII.2017.2773646.

[15] Z. Zheng, S. Xie, H.N. Dai, X. Chen, H. Wang, Blockchain challenges and opportunities: a survey, Int. J. Web Grid Serv. 14 (2018) 352–375, https://doi.org/10.1504/IJWGS.2018.095647.

[16] M. Rahouti, K. Xiong, N. Ghani, Bitcoin concepts, threats, and machine-learning security solutions, IEEE Access 6 (2018) 67189–67205, https://doi.org/10.1109/ACCESS.2018.2874539.

[17] B. Mohanta, P. Das, S. Patnaik, Healthcare 5.0: a paradigm shift in digital healthcare system using artificial intelligence, IOT and 5G communication, in: Proc. - 2019 Int. Conf. Appl. Mach. Learn. ICAML 2019, 2019, https://doi.org/10.1109/ICAML48257.2019.00044.

[18] E. Mbunge, B. Muchemwa, S. Jiyane, J. Batani, Sensors and healthcare 5.0: transformative shift in virtual care through emerging digital health technologies, Glob. Health J. 5 (2021) 169–177, https://doi.org/10.1016/j.glohj.2021.11.008.

[19] M. Bhavin, S. Tanwar, N. Sharma, S. Tyagi, N. Kumar, Blockchain and quantum blind signature-based hybrid scheme for healthcare 5.0 applications, J. Inf. Secur. Appl. 56 (2021), 102673, https://doi.org/10.1016/j.jisa.2020.102673.

[20] S. Aggarwal, R. Chaudhary, G.S. Aujla, N. Kumar, K.K.R. Choo, A.Y. Zomaya, Blockchain for smart communities: applications, challenges and opportunities, J. Netw. Comput. Appl. 144 (2019) 13–48, https://doi.org/10.1016/j.jnca.2019.06.018.

[21] M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, P. McCallum, A. Peacock, Blockchain technology in the energy sector: a systematic review of challenges and opportunities, Renew. Sustain. Energy Rev. 100 (2019) 143–174, https://doi.org/10.1016/j.rser.2018.10.014.

[22] G. Li, M. Dong, L.T. Yang, K. Ota, J. Wu, J. Li, Preserving edge knowledge sharing among IoT services: a blockchain-based approach, IEEE Trans. Emerg. Top. Comput. Intell. 4 (2020) 653–665, https://doi.org/10.1109/TETCI.2019.2952587.

[23] Z. Zhou, B. Wang, M. Dong, K. Ota, Secure and efficient vehicle-to-grid energy trading in cyber physical systems: integration of blockchain and edge computing, IEEE Trans. Syst. Man, Cybern. Syst. 50 (2020) 43–57, https://doi.org/10.1109/TSMC.2019.2896323.

[24] X. Du, B. Chen, M. Ma, Y. Zhang, Research on the application of blockchain in smart healthcare: constructing a hierarchical framework, J. Healthc. Eng. (2021), https://doi.org/10.1155/2021/6698122, 2021.

[25] B. Ihnaini, M.A. Khan, T.A. Khan, S. Abbas, M.S. Daoud, M. Ahmad, M.A. Khan, A smart healthcare recommendation system for multidisciplinary diabetes patients with data fusion based on deep ensemble learning, Comput. Intell. Neurosci. (2021), https://doi.org/10.1155/2021/4243700, 2021.

[26] M.A. Khan, Challenges facing the application of IoT in medicine and healthcare, Int. J. Comput. Integrated Manuf. 1 (2021), https://doi.org/10.54489/ijcim.v1i1.32.

[27] M.F. Khan, T.M. Ghazal, R.A. Said, A. Fatima, S. Abbas, M.A. Khan, G.F. Issa, M. Ahmad, M.A. Khan, An iomt-enabled smart healthcare model to monitor elderly

people using machine learning technique, Comput. Intell. Neurosci. (2021), https://doi.org/10.1155/2021/2487759, 2021.
[28] J. Xu, B.S. Glicksberg, C. Su, P. Walker, J. Bian, F. Wang, Federated learning for healthcare informatics, J. Healthc. Informatics Res. 5 (2021) 1–19, https://doi.org/10.1007/s41666-020-00082-4.
[29] Y. Li, B. Shan, B. Li, X. Liu, Y. Pu, Literature review on the applications of machine learning and blockchain technology in smart healthcare industry: a bibliometric analysis, J. Healthc. Eng. (2021), https://doi.org/10.1155/2021/9739219, 2021.
[30] S.Y. Siddiqui, I. Naseer, M.A. Khan, M.F. Mushtaq, R.A. Naqvi, D. Hussain, A. Haider, Intelligent breast cancer prediction empowered with fusion and deep learning, Comput. Mater. Continua (CMC) 67 (2021), https://doi.org/10.32604/cmc.2021.013952.
[31] H. Medjahed, D. Istrate, J. Boudy, J.L. Baldinger, B. Dorizzi, A pervasive multi-sensor data fusion for smart home healthcare monitoring, Fuzzy Syst. Conf. (2011), https://doi.org/10.1109/FUZZY.2011.6007636.
[32] W. Dai, T.S. Brisimi, W.G. Adams, T. Mela, V. Saligrama, I.C. Paschalidis, Prediction of hospitalization due to heart diseases by supervised learning methods, Int. J. Med. Inf. 84 (2015) 189–197, https://doi.org/10.1016/j.ijmedinf.2014.10.002.
[33] Y.J. Son, H.G. Kim, E.H. Kim, S. Choi, S.K. Lee, Application of support vector machine for prediction of medication adherence in heart failure patients, Healthc. Inform. Res. 16 (2010) 253–259, https://doi.org/10.4258/hir.2010.16.4.253.
[34] A. Tariq, L.A. Celi, J.M. Newsome, S. Purkayastha, N.K. Bhatia, H. Trivedi, J. W. Gichoya, I. Banerjee, Patient-specific COVID-19 resource utilization prediction using fusion AI model, Npj Digit. Med. 4 (2021) 1–9, https://doi.org/10.1038/s41746-021-00461-0.
[35] A. Sedik, A.M. Iliyasu, B.A. El-Rahiem, M.E. Abdel Samea, A. Abdel-Raheem, M. Hammad, J. Peng, F.E. Abd El-Samie, A.A. Abd El-Latif, Deploying machine and deep learning models for efficient data-augmented detection of COVID-19 infections, Viruses 12 (2020), https://doi.org/10.3390/v12070769.
[36] A. Qayyum, K. Ahmad, M.A. Ahsan, A. Al-Fuqaha, J. Qadir, Collaborative Federated Learning for Healthcare: Multi-Modal Covid-19 Diagnosis at the Edge, 2021 arXiv preprint arXiv:2101.07511.
[37] T.S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I.C. Paschalidis, W. Shi, Federated learning of predictive models from federated Electronic Health Records, Int. J. Med. Inf. 112 (2018) 59–67, https://doi.org/10.1016/j.ijmedinf.2018.01.007.
[38] Y. Chang, C. Fang, W. Sun, et al., A blockchain-based Federated Learning Method for Smart Healthcare, Computational Intelligence and Neuroscience 2021 (2021) 1–12, https://doi.org/10.1155/2021/4376418.
[39] S. Squarepants, Bitcoin: A Peer-To-Peer Electronic Cash System, SSRN Electron. J, 2022, https://doi.org/10.2139/ssrn.3977007.
[40] A. Rehman, A. Athar, M.A. Khan, S. Abbas, A. Fatima, Atta-Ur-Rahman, A. Saeed, Modelling, simulation, and optimization of diabetes type II prediction using deep extreme learning machine, J. Ambient Intell. Smart Environ. 12 (2020) 125–138, https://doi.org/10.3233/AIS-200554.
[41] M.A. Khan, A. Rehman, K.M. Khan, M.A. Al Ghamdi, S.H. Almotiri, Enhance intrusion detection in computer networks based on deep extreme learning machine, Comput. Mater. Continua (CMC) 66 (2021), https://doi.org/10.32604/cmc.2020.013121.
[42] A. Haider, M.A. Khan, A. Rehman, M. Ur Rahman, H.S. Kim, A real-time sequential deep extreme learning machine cybersecurity intrusion detection system, Comput. Mater. Continua (CMC) 66 (2020), https://doi.org/10.32604/cmc.2020.013910.
[43] M.A. Khan, S. Abbas, A. Rehman, Y. Saeed, A. Zeb, M.I. Uddin, N. Nasser, A. Ali, A machine learning approach for blockchain-based smart home networks security, IEEE Netw 35 (2021) 223–229, https://doi.org/10.1109/MNET.011.2000514.
[44] Oxford, Parkinsons Data Set, UCI Learn, Repos., 2007.
[45] A.G.M. Tavallaee, E. Bagheri, W. Lu, Canadian Institute for Cybersecurity, UNB, NSL-KDD Dataset, 2018.
[46] R. Sheibani, E. Nikookar, S. Alavi, An ensemble method for diagnosis of Parkinson's disease based on voice measurements, J. Med. Signals Sens. 9 (2019), https://doi.org/10.4103/jmss.JMSS_57_18.
[47] J.M. Tracy, Y. Özkanca, D.C. Atkins, R. Hosseini Ghomi, Investigating voice as a biomarker: deep phenotyping methods for early detection of Parkinson's disease, J. Biomed. Inf. 104 (2020), https://doi.org/10.1016/j.jbi.2019.103362.
[48] D. Sztaho, I. Valalik, K. Vicsi, Parkinson's disease severity estimation on Hungarian speech using various speech tasks, in: 2019 10th Int. Conf. Speech Technol. Human-Computer Dialogue, SpeD 2019, 2019, https://doi.org/10.1109/SPED.2019.8906277.
[49] O. Yaman, F. Ertam, T. Tuncer, Automated Parkinson's disease recognition based on statistical pooling method using acoustic features, Med. Hypotheses 135 (2020), https://doi.org/10.1016/j.mehy.2019.109483.
[50] H. Kuresan, D. Samiappan, S. Masunda, Fusion of wpt and mfcc feature extraction in Parkinsons disease diagnosis, Technol. Health Care 27 (2019) 363–372, https://doi.org/10.3233/THC-181306.

ELSEVIER

Privacy-preserving techniques for decentralized and secure machine learning in drug discovery

Aljoša Smajić, Melanie Grandits, Gerhard F. Ecker *

Department of Pharmaceutical Sciences, University of Vienna, Vienna, Austria

Data availability, data security, and privacy concerns often hamper optimal performance efficiency of machine learning (ML) techniques. Therefore, novel techniques for the utilization of private/sensitive data in the field of drug discovery have been proposed for ML model-building tasks. Some examples of the different techniques are secure multiparty computation, distributed deep learning, homomorphic encryption, blockchain-based peer-to-peer networking, differential privacy, and federated learning, as well as combinations of such techniques. In this paper, we present an overview of these techniques for decentralized ML to illustrate its benefits and drawbacks in the field of drug discovery.

Introduction

Machine learning (ML) and deep learning (DL) approaches are extensively used in drug discovery and development.[p(1),p(2)] Both techniques have revolutionized how new drugs are identified, and their importance can be seen by the recently signed FDA Modernization Act 2.0, which allows the application of alternative models to animal studies before human clinical trials for assessing the safety and effectiveness of a new drug.[3] Established predictive models can streamline the drug discovery process, analyze large datasets, identify off-target interactions, and predict the efficacy and safety of new drug candidates. Moreover, by providing ML and DL algorithms with large datasets of high quality, the performance of these models can be significantly increased. This allows the generation of models with a broader chemical space, which renders data availability a crucial aspect when applying such methods.

Unfortunately, datasets from pharmaceutical and biotech companies are usually not disclosed to the public, as they are considered valuable due to high investments and are central for developing new drugs and other bioactive molecules.[4] Data sharing and access also carries a risk of violation of other legitimate private interests, such as commercial interests. A violation of these interests can lead to a loss of competitive advantage, which has a negative impact on the company's revenue and profitability.

Collaboration between companies, individuals, or institutions regarding data sharing would speed up drug development and make it safer, less redundant, and less risky.[5] In addition, (i) benchmarking and comparison of different models and methods would be possible, (ii) existing models could be evaluated against new datasets, providing insights into their robustness and limitations, and (iii) potential risks and adverse effects associated with drugs and treatments could be identified. Nevertheless, conventional methods for data sharing are often restricted to centralized approaches. In such centralized settings, ML model training is performed on one central server that stores all data locally. Data must therefore be moved from one instance where it is processed and utilized for ML training to another. While this poses no issues regarding internal data exchange within the company, applying these techniques to third parties presents a major issue given that pharmaceutical company datasets are sensitive or confidential.

Additionally, when sharing pharmaceutical data with other institutions, centralized approaches have a higher risk of data leaks or breaches since they rely on a single central server to manage every step of the training procedure. Therefore, decentralized

* *Corresponding author.*

1359-6446/© 2023 Published by Elsevier Ltd.
https://doi.org/10.1016/j.drudis.2023.103820

ML for drug discovery was proposed.[6] This would make it possible for several parties to work together on developing a model without disclosing internal datasets, as is done traditionally.

Decentralized ML can be quite advantageous for some applications, like virtual screening and drug design. Researchers can increase the accuracy of virtual screening and speed up the drug development process by training models for compound activity prediction against therapeutic targets, using datasets spread across many pharmaceutical companies and research institutions. Moreover, when novel compounds are designed, knowledge regarding the chemical space used for model building may additionally be enriched for increased efficacy and decreased toxicity. Additionally, a privacy-preserved decentralized environment would enable the models to capture patterns in datasets outside their own resources.

To use datasets across numerous companies and institutions while protecting the confidentiality and privacy of individual data points, various strategies have been developed.[p(7),p(8)] Each of these techniques has benefits, but they are also constrained in certain aspects. In this review, we address these methodologies' most recent applications in the context of drug discovery. We discuss the applicability of the strategies, their significance in the early stages of drug discovery, as well as future potential, new developments, and advantages and disadvantages. Figure 1 shows a brief summary of the methodologies discussed in this review.

Techniques for privacy-preserved decentralized ML for drug discovery

Secure multiparty computation

Secure multiparty computation (SMPC) is an emerging cryptographic technology that was introduced in the late 1970s. Compared with traditional cryptography approaches, which ensure the security and integrity of communications, SMPC places more emphasis on protecting data from disclosure by other participants involved in the computation.[9] The idea behind SMPC is the development of methods that allow teams to work together on a task using internal inputs while keeping those inputs private. Data can be shared with participants since SMPC protocols provide guarantees comparable to those of a trusted party. Joint analyses are performed and only the results are sent back to the participating parties (Figure 2). This avoids direct data exchange or disclosure of any sensitive information between the participants by following specific cryptographic protocols. This approach can be used to address the shortcomings of federated data networks and does not require a trusted third party to be involved.[10] Moreover, unlike federated learning (FL), SMPC reveals the final results to the participants without disclosing intermediate information during the computation, providing a higher level of security.

Applications of SMPC in drug discovery

Nowadays, SMPC can be found in many domains such as finance, healthcare, and cybersecurity. In the field of drug discovery, as well as in other research fields in computational chemistry and biology, modern cryptographic techniques started to be introduced in 2017.[p(11),p(12),p(13)] SMPC allows pharmaceutical industries to keep private information safe by protecting data from being leaked to others, while also allowing multiple participants to collaboratively perform computations. One example of SMPC in the field of biomedical research is EasySMPC, a powerful no-code tool for practical and secure multiparty computation. This standalone desktop application makes use of the SMPC method Arithmetic Secret Sharing. This approach securely sums up predefined sets of variables among different participants during input sharing and output reconstruction and applies this method to a graphical user interface.[10] Moreover, recent computational and technical advances led to the introduction of quantitative structure–activity relationship (QSAR) approaches, used in combination with SMPC and for privacy-preserving drug target interactions (DTIs).[14] Ma and colleagues demonstrated the utilization of SMPC in combina-

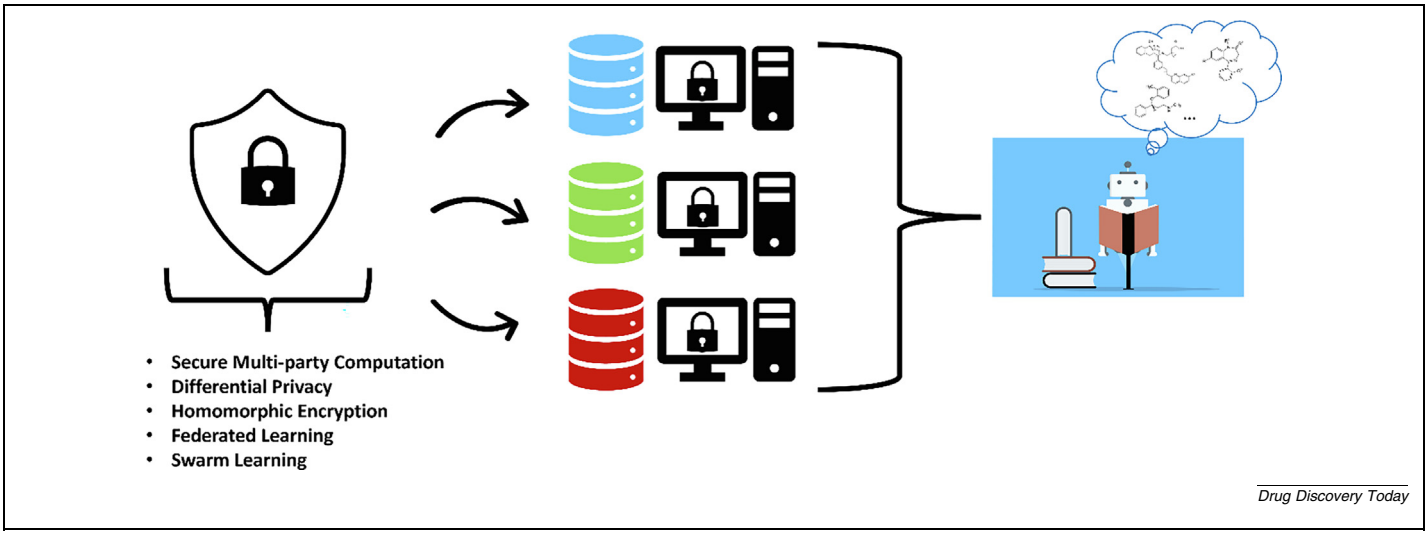


FIGURE 1
Simplified depiction of the existing privacy-preserved techniques applicable to drug discovery.

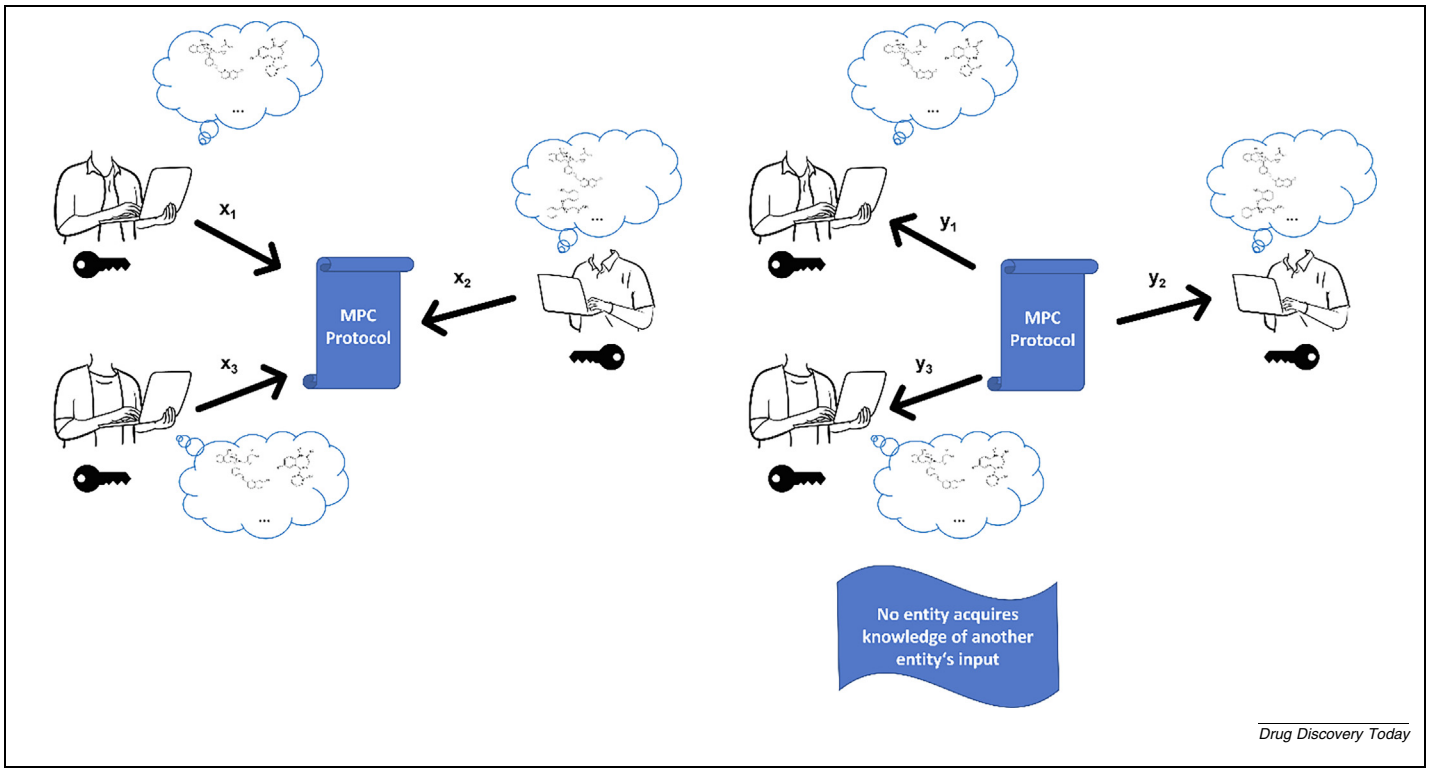


FIGURE 2
Schematic overview of secure multiparty computation.

tion with QSAR in an approach called QSARMPC.[14] The MPC is built on a DTINet with a two-hidden-layer neural network and is primarily intended for solving regression problems.[15] This approach has shown reasonably good performance when applied to QSAR under the MPC protocol. Additionally, a tool for DTI prediction based on MPC, called DTIMPC, was introduced by the same group for the prediction of novel DTIs from drug-related heterogenous information. QSARMPC was examined in terms of whether the use of protected private data will cause a drop in the prediction accuracy compared with general approaches, where public collaborations are using shared information. Finally, a slightly higher squared Pearson correlation coefficient (R^2) of 0.446 over 15 datasets was obtained by QSARMPC, whereas the general approach yielded an R^2 of 0.425. Regarding the DTI approach, two different scenarios were tested, one with all samples included and another with a ratio of 1:10 of positive:negative samples. A 10-fold cross-validation of DTIs with an imbalanced distribution of positive and negative entries was designed with the intention of simulating a practical implementation scenario. DTIMPC showed that pharmaceutical companies or institutions could achieve high-quality collaborations.[16] Moreover, in the field of genomic sequence comparison, a so-called secure two-party computation (S2PC) is commonly applied. This approach allows the alignment of two genome sequences in a secure manner. S2PC can be divided into two groups: homomorphic-encryption-based construction and garbled-circuit-based construction.[17] Another approach, Sequre, a high-performance framework for secure multiparty computation, is a cryptographic tool that allows computation based on sensitive biomedical data. Sequre has been applied to different tasks such as secure genome-wide association analysis, secure drug-target interaction prediction, and secure metagenomic binning. When compared with existing state-of-the art pipelines, Sequre achieved up to a sevenfold reduction in code length, while the overall execution time of the implemented pipelines was reduced three- to fourfold, and the network utilization was 17 % lower. Additionally, it achieved the best runtime performance in most of the cases while also being one of the simplest frameworks to use.[18] SMPC, as a relatively new technology, faces certain challenges. For example, in some cases regulations such as data protection and privacy laws do not consider the use of SMPC techniques, which creates uncertainty about the legality of using these methods and may leave organizations unsure about whether they can use such techniques without violating any regulations.[p(19),p(20)] In addition, communicating with relevant stakeholders about the properties of SMPC can be very challenging.[p(10),p(16),p(21),p(22),p(23),p(24),p(25)]

Differential privacy and differential private deep learning

Differential privacy (DP) can be considered as a decentralized technique when it is used to protect the privacy of sensitive data stored by multiple parties. This approach allows the participants to share data with each other while preserving the privacy of each individual's data. It provides a privacy guarantee that holds regardless of what someone knows or does when trying to perform data intrusions.[26] DP encompasses ML as well as basic statistical calculations, such as averages. It is important to note that DP is not a single instrument but rather a means of quantifying and managing privacy risks for which human-made

technological tools have been developed. The concept of DP was first introduced in 2006 by Dwork *et al.*, who asserted that the presence or absence of any individual entry in a dataset should not substantially affect the outcome of a computation performed on a dataset.[27] On an abstract level, any computation is considered differentially private if, for two datasets that differ by only one entry, the likelihood of obtaining a particular result is nearly the same. By applying calibrated noise to the output, the contribution of any individual in the dataset can be masked while still preserving the overall accuracy of the analysis/computation. Analyses using DP are different from conventional statistical analyses, which compute averages, medians, and linear regression equations without the addition of random noise. Therefore, DP analysis outcomes are not exact, but rather an *approximation* and can yield different results when performed multiple times. Even with the addition of noise and with DP, ML models can still achieve good overall performance when the introduced noise is carefully calibrated to balance privacy and utility. Moreover, the accuracy of DP ML models can be increased by carefully adjusting the privacy settings and utilizing innovative methods like adaptive privacy budgeting and privacy amplification. To compensate for the noise and keep accuracy levels comparable, larger datasets and advanced model architectures could be used.[p(28),p(29),p(30)]

Applications of DP in drug discovery

Up to now, DP has been applied mostly to drug sensitivity prediction and omics data. In the work of Honkela *et al.*, it was demonstrated that the utilization of DP improves drug sensitivity prediction, and useful predictors can be learned under powerful DP guarantees from moderately sized datasets.[31] Moreover, the authors illustrated improvements in the accuracy of drug sensitivity predictions using a robust private regression method.[31] In 2022, Islam *et al.* demonstrated that a combination of DP and DL, known as differential private deep learning, can be used for the prediction of breast cancer status, cancer type, and drug sensitivity in cancer cell lines using sensitive human genomic data while preserving individuals' privacy.[32] Perturbation of the models showed less sensitivity to changes in the input and inhibition of unauthorized access by making it difficult to identify or infer the raw input data used to train the models. Thus, raw privacy-sensitive input data can be used to build DL models that require large amounts of data.[32] Unfortunately, DP and differential private deep learning methods have not been applied to as many domains of drug discovery as other approaches, such as FL.

It is important to mention that privacy-aware data analysis is a trade-off between privacy and the utilization of the data in the analysis. Adding noise to the datasets can make it more difficult to extract meaningful information, and DP can therefore result in a loss of data utility. Another issue that arises is that it can be difficult to determine the appropriate level of noise to add to the datasets to establish the desired level of privacy. However, in ongoing research, DP has been used in FL.[33]

Homomorphic encryption and fully homomorphic encryption

Homomorphic encryption (HE) is another cryptographic technology that allows the computation of encrypted datasets, with some limitations. The concept of performing computations on encrypted data for computation tasks was first introduced as "privacy homomorphisms" by Rivest *et al.* in 1978.[34] Subsequently, an improved version named "fully homomorphic encryption" (FHE) was published by Craig Gentry in his 2009 thesis, which enables arbitrary computation over encrypted datasets (Figure 3).[p(35),p(36),p(37),p(38)] In FHE, addition and multiplication can be performed over encrypted data by describing arbitrary functions as Boolean circuits in an encryption scheme.[39] Moreover, improved versions followed and led to three generations of research.[p(39),p(40)] The "third generation" of FHE, which was introduced between 2012 and 2013, brought practical improvements that greatly boosted the efficiency of FHE schemes.[p(39),p(41)] Depending on the scheme chosen, operations, types of activation, architectures as well as the plaintext space can differ. Messages need to be encoded into the plaintext space as most schemes, such as Brakerski-Fan-Vercauteren (BFV), Brakerski-Gentry-Vaikuntanathan (BGV), Yet Another Somewhat Homomorphic Encryption (YASHE), and Fan and Vercauteren (FV), only support integers, while others such as Cheon-Kim-Kim-Song (CKKS) support real numbers, and the fast fully homomorphic encryption scheme over the Torus (TFHE) supports individual bits. The specific scheme chosen will determine the capabilities and limitations of the encryption system.[42]

These advancements also marked the release of the first open-source FHE implementations.[43] Technical improvements yielded a collection of open-source software libraries such as SEAL, HElib, TFHE, HEAAN, and PALISADE.[p(44),p(45),p(46),p(47),p(48)] Starting in 2017, a consortium of experts from industry, government, and academia allied and formed the Homomorphic Encryption Standardization Consortium. By promoting workshops, they developed a *Homomorphic Encryption Standard* in 2018, illustrating the security requirements for FHE applications. FHE was described by the consortium as being based on three models of homomorphic computation: Boolean circuits, modular (exact) arithmetic, and approximate number arithmetic.[49] Additionally, the implementation of neural networks for encrypted data has been proposed and applied for inference. Unfortunately, training becomes impractical due to high computational complexity. In addition, the same key is used for the encryption of weights as well as data, meaning that the server will have a model that it cannot access after training is performed.[42]

Furthermore, large-scale decentralized model building becomes impractical due to the complexity of the computations being performed on the datasets. Nevertheless, with further advances in hardware acceleration and optimized algorithms, this technique could become a potential key player for privacy-preserved decentralized model building. This could lead to new applications in Drug Discovery.

Federated learning

Each of the three techniques mentioned above is essential in preserving the privacy of the datasets and achieving the goal of avoiding any data breaches. Commonly, some of these techniques are combined with approaches such as FL.[p(50),p(51)]

With the help of a distinctive algorithm, federated averaging (FedAvg), which Google first revealed in 2016, it is possible to relocate computations of data while preserving any institution's data privacy. Models can be sent to organizations without direct access to the training data by combining local stochastic gradient

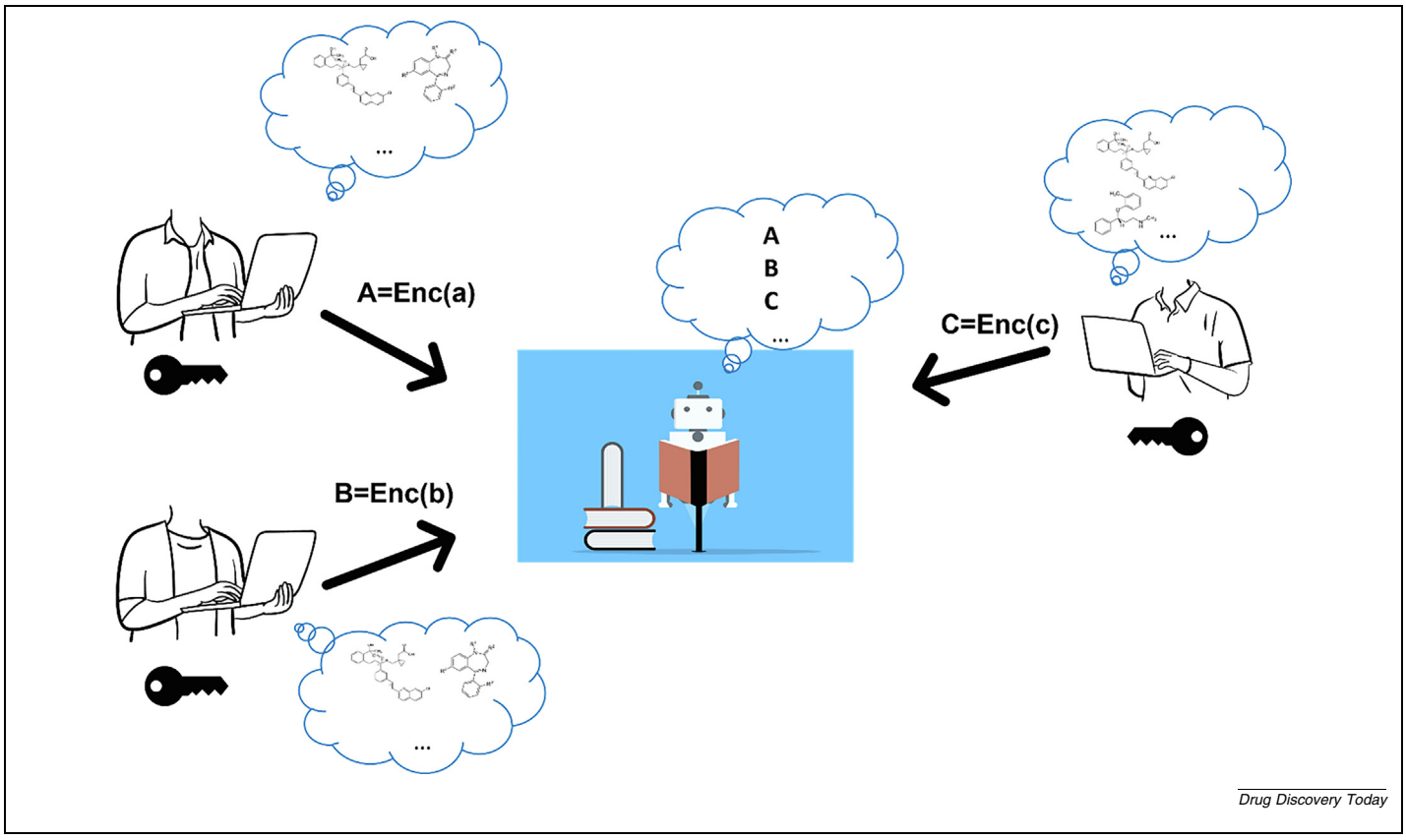


FIGURE 3
Simplified depiction of a fully homomorphic encryption.

descent (SGD) on each client with a server in charge of model averaging. Typically, each institution uses its own loss function to execute SGD. The global model can then be updated by sending the aggregated models back to the server (Figure 4). By using individual devices instead of a single device or a cloud server to store the datasets, the risks associated with privacy and security are reduced.[53] This is known as model-driven federated learning (MD-FL), which promises to maintain confidentiality by requiring certain information for model creation. FL methods enable a situation in which privacy concerns are respected and data are stored locally while only the parameters are sent to a central custodian.[9] Since its introduction in 2016, federated learning artificial intelligence (FL-AI) has been suggested and used for a wide range of techniques where data privacy and security are crucial. Additionally, FL has been used in conjunction with novel communication protocols, encryption ideas, and optimization methods.[50]

Depending on the size of participants, FL can be divided into two categories: cross-device FL and cross-silo FL. The latter, which involves knowledge sharing between a small number of entities, has been used extensively in drug discovery. Alternatively, FL may be divided into three categories: horizontal FL (HFL), vertical FL (VFL), and federated transfer learning (FTL), each of which is capable of handling a variety of learning tasks.[50] When users are spread among multiple clients, HFL is used, but the feature space remains unchanged. On the other hand, VFL is used for vertically partitioned data, where each device has access to a distinct subset of features but all devices share the same set of data samples. FTL can be used in situations when there are insufficient labeled data points available for a model to learn a specific task. In other words, FTL can be considered as the cross-section of HFL and VFL.

Applications of FL in drug discovery

Specifically in the domain of drug discovery, FL represents the most dynamic research approach. MELLODY (MachinE Learning Ledger Orchestration for Drug DiscoverY), an Innovative Medicines Initiative (IMI) project, has been developed with 17 public and private partners. Currently, it is the largest FL-based study, analyzing 20 million small compounds from over 40,000 biological assays from 10 pharmaceutical companies. The project used a massive multitask setup based on MD-FL, enabling each participant to contribute to a range of tasks. Across all the tasks a common trunk is shared that allows the model to learn a common representation of the datasets. Each task can be perceived as a separate "head" of the model that can learn unique characteristics of the other tasks. Transfer learning is further applied between the trunk and the heads, which enables the model to leverage the knowledge it has gained from one task to improve its performance on another task. For the evaluation of model performance, additional models were developed on datasets from one collaboration partner.[53] The MD-FL approach showed an average improvement of 2 % in terms of the coefficient of deter-

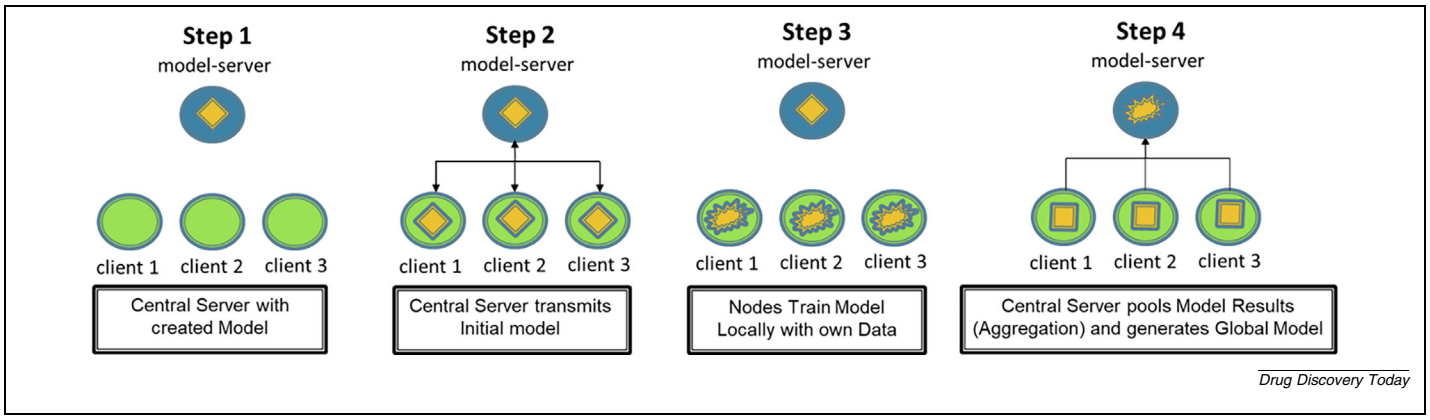


FIGURE 4

Simplified depiction of a federated learning approach. Figure adapted from McMahan et al.[52], published in Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS).[52]

mination (R^2) for regression tasks and 4 % in terms of area under the precision-recall curve (PR-AUC) for classification tasks when applied across tasks and partners. Additionally, the approach showed an average expansion of the applicability domain by 10 %.[54]

Another FL platform, Effiris,[54] uses a so-called data-driven FL (DD-FL) platform, which is based on private knowledge transfer. In this approach private models are trained on in-house data and a shared dataset is used for predicting the labels. The resulting predictions are shared with a central custodian that aggregates the predicted labels from all participants into a single consolidated label. The consolidated label is used for the creation of a federated dataset, which is in turn used to train a federated model. This dataset is shared between the participants to increase in-house datasets, allowing the construction of local models with better performance.[53]

In the work of Chen *et al.*, an FL setup for QSAR tasks was demonstrated. This approach was applied to 15 QSAR regression tasks from Kaggle, simulating a decentralized setting with two, three, or four participants.[55] Differences in data size between participants were analyzed after random data partitioning. Moreover, this study did not consider cases where the participants exhibited non-independent and identically distributed (IID) data. However, the FL model outperformed each individual participant irrespective of the data size gap, but it should be noted that the absence of non-IID may considerably impact modelling performance.[53] Other FL projects that have been applied to drug/molecular discovery include FedChem, FedGraphNN, FL-Disco, kMol, and the work of Xiong *et al.*[p(56),p(57),p(58),p(59),p(60),p(61),62] A detailed description of these projects and the application of FL can be found in the work of Thierry Hanser, which illustrates the benefits of and remaining challenges to FL application to molecular discovery.[55]

Swarm learning and its application in drug discovery

While FL requires a central coordinator, swarm learning (SL) is a completely independent approach as it uses blockchain-based peer-to-peer networking for exchanging parameters established by individually trained local models (Figure 5). Preauthorized participants can execute transactions, and appropriate authorization measures are applied for recognizing network participation.

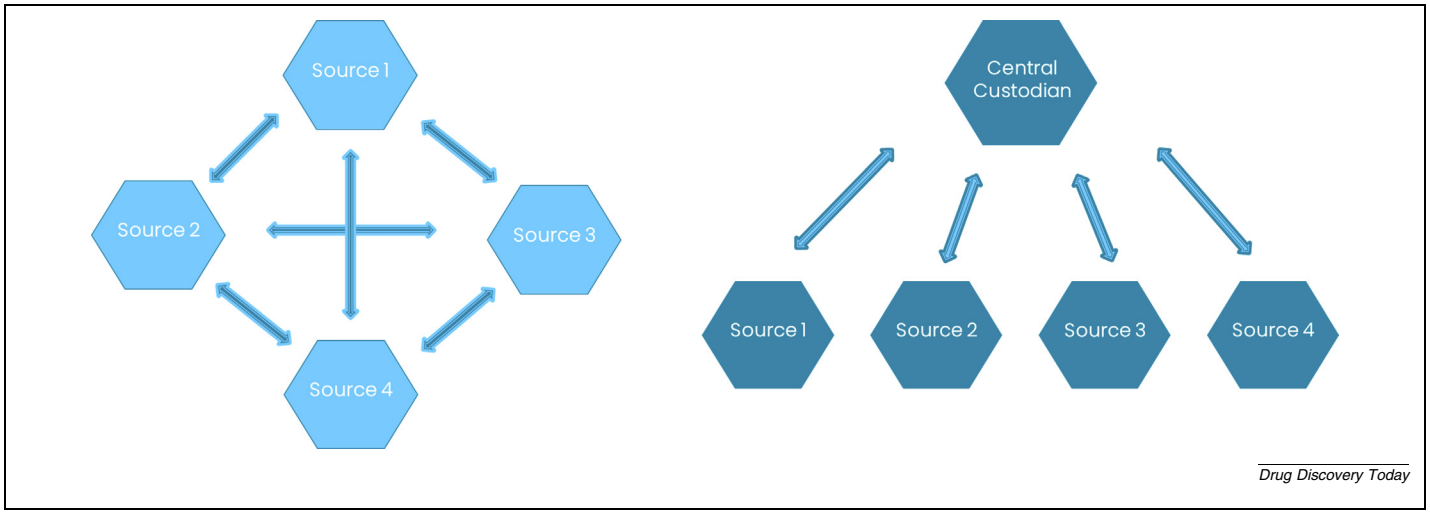


FIGURE 5

Basic representation and comparison of a swarm learning approach (left) and federated learning approach (right).

Blockchain smart contracts are used for enrolling new nodes, which obtain the model and perform local model training until defined criteria are met for synchronization. By using a swarm application programming interface (API), model parameters are transferred and merged to create an updated model similar to the FL approach. Three different use cases were selected to which SL was applied: prediction of leukemias by using peripheral blood mononuclear cell (PBMC) transcriptomes, identification of tuberculosis using blood transcriptomes, and identification of COVID-19.[63] Moreover, SL provides confidentiality by design and has the potential to incorporate advancements such as DP, functional encryption, and encrypted transfer learning.

Concluding remarks

In the context of pharmaceutical research, privacy-preserving decentralized approaches are crucial since they would allow for the enrichment of chemical and biological space for predictive modeling. When applied to collaborative activities, traditional and central ML techniques exhibited shortcomings. Up to now, few real-world implementations in the drug discovery domain have been introduced and deployed. The other implementations mentioned in this work were often simulated for benchmarking purposes. However, innovative, integrated methods and platforms have the potential to transform collaborative teamwork and may result in significant advancements in the areas covered in this work. Comparing the techniques, it can be seen that FL has gained considerable popularity and holds great importance as a research domain in combination with drug discovery. This can be demonstrated by examining two real-world implementations, namely MELLODY and Effiris. Moreover, the combination of multiparty privacy-protected ML techniques, such as HE, DP, and SMPC, in conjunction with decentralized data and federated ML, holds great importance in the field of drug discovery. As technology advances, these techniques will gain further significance.

Declarations of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Data availability

No data was used for the research described in the article.

Acknowledgment

The Pharmacoinformatics Research Group (Ecker lab) acknowledges funding provided by the Austrian Science Fund FWF AW012321 MolTag.

References

1. Vo AH, Van Vleet TR, Gupta RR, Liguori MJ, Rao MS. An overview of machine learning and big data for drug toxicity evaluation. *Chem Res Toxicol*. 2020;33:20–37. https://doi.org/10.1021/acs.chemrestox.9b00227.
2. Klambauer G, Hochreiter S, Rarey M. Machine learning in drug discovery. *J Chem Inf Model*. 2019;59:945–946. https://doi.org/10.1021/acs.jcim.9b00136.
3. S.5002 – 117th Congress (2021-2022): FDA Modernization Act 2.0. Congress.gov. Library of Congress. Accessed 5 February 2023. https://www.congress.gov/bill/117th-congress/senate-bill/5002
4. Rieke N et al. The future of digital health with federated learning. *NPJ Digit Med*. 2020;3:1–7. https://doi.org/10.1038/s41746-020-00323-1.
5. Institute of Medicine (US) Extending the Spectrum of Precompetitive Collaboration in Oncology Research: Workshop Summary. Washington (DC): National Academies Press (US); 2010. BENEFITS OF COLLABORATING. Accessed 8 March 2023. https://www.ncbi.nlm.nih.gov/books/NBK210038/
6. Innovative Medicines Initiative. MELLODDY: Machine learning ledger orchestration for drug discovery. Accessed 21 August 2023. https://www.imi.europa.eu/projects-results/project-factsheets/melloddy
7. Domingo-Ferrer J, Blanco-Justicia A. Privacy-preserving technologies. *Int Libr Ethics, Law Technol*. 2020;21:279–297. https://doi.org/10.1007/978-3-030-29053-5_14/COVER.
8. Hiwale M, Walambe R, Potdar V, Kotecha K. A systematic review of privacy-preserving methods deployed with blockchain and federated learning for the telemedicine. *Healthc Anal*. 2023;3. https://doi.org/10.1016/j.health.2023.100192 100192.
9. Shokri R, Shmatikov V. Privacy-preserving deep learning. *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 29 September 2015–02 October 2015. IEEE; 2016: 909–910. https://doi.org/10.1109/ALLERTON.2015.7447103
10. Wirth FN, Kussel T, Müller A, Hamacher K, Prasser F. EasySMPC: a simple but powerful no-code tool for practical secure multiparty computation. *BMC Bioinform*. 2022;23:1–17. https://doi.org/10.1186/s12859-022-05044-8.
11. Jagadeesh KA, Wu DJ, Birgmeier JA, Boneh D, Bejerano G. Deriving genomic diagnoses without revealing patient genomes. *Science*. 2017;357:692–695. https://doi.org/10.1126/science.aam9710.
12. Hie B, Cho H, Berger B. Realizing private and practical pharmacological collaboration. *Science*. 2018;362:347–350. https://doi.org/10.1126/science.aat4807.
13. Cho H, Wu DJ, Berger B. Secure genome-wide association analysis using multiparty computation. *Nat Biotechnol*. 2018;36:547–551. https://doi.org/10.1038/nbt.4108.
14. Ma R et al. Secure multiparty computation for privacy-preserving drug discovery. *Bioinformatics*. 2020;36:2872–2880. https://doi.org/10.1093/bioinformatics/btaa038.
15. Luo Y et al. A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information. *Nat Commun*. 2017;8:573. https://doi.org/10.1038/s41467-017-00680-8.
16. Bogdanov D, Kamm L, Laur S, Pruulmann-Vengerfeldt P. Secure multi-party data analysis: end user validation and practical experiments. *IACR Cryptol ePrint Arch*. 2013;1:826:1–826:19.
17. Zhao C, Zhao S, Zhao M, et al.. Secure multi-party computation: theory, practice and applications. *Inf Sci (Ny)*. 2019;476:357–372. https://doi.org/10.1016/j.ins.2018.10.024.
18. Smajlović H, Shajii A, Berger B, Cho H, Numanagić I. Sequre: a high-performance framework for secure multiparty computation enables biomedical data sharing. *Genome Biol*. 2023;24:1–18. https://doi.org/10.1186/s13059-022-02841-5.
19. HIPPA 1996. Health Insurance Portability and Accountability Act of 1996 (HIPAA). CDC. Health Insurance Portability and Accountability Act of 1996 (HIPAA). 2019. Accessed 21 August 2023. https://www.cdc.gov/phlp/publications/topic/hipaa.html
20. I (Legislative acts) REGULATIONS REGULATION (EU) 2016/679 of The European Parliament and of The Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repeali. Accessed 21 August 2023. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679&from=EN
21. Paverd AJ, Martin A, Brown I. Modelling and automatically analysing privacy properties for honest-but-curious adversaries with applications in the smart grid. 2014. Accessed 27 March 2023. https://www.cs.ox.ac.uk/people/andrew.paverd/casper/
22. Desai T, Ritchie F, Welpton R. Five Safes: designing data access for research. Economics working paper series. 2016. Accessed 27 March 2023. https://www.researchgate.net/publication/292975549_Five_Safes_designing_data_access_for_research

23. Evans D, Kolesnikov V, Rosulek M. *A Pragmatic Introduction to Secure Multi-Party Computation*. 2018. https://doi.org/10.1561/9781680835090.
24. Veeningen M, Chatterjea S, Horváth AZ, et al.. Enabling analytics on sensitive medical data with secure multi-party computation. *Stud Health Technol Inform*. 2018;247:76–80. https://doi.org/10.3233/978-1-61499-852-5-76.
25. Tõldsepp K, Pruulmann-Vengerfeldt P, Laud P. Usable and efficient secure multiparty computation—requirements specification based on the interviews. Deliverables in usable and efficient secure multiparty computation (UaESMC) Research Project 2015. Accessed 27 March 2023. https://www.usable-security.eu
26. Dwork C, Roth A, Dwork C, Roth A. The algorithmic foundations of differential privacy. *Found Trends R Theor Comput Sci*. 2014;9:211–407. https://doi.org/10.1561/0400000042.
27. Dwork C, McSherry F, Nissim K, Smith A. Calibrating noise to sensitivity in private data analysis. In: Halevi S, Rabin T, eds. *Theory of Cryptography*. *TCC 2006*. *Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer; 2006:265–284 https://doi.org/10.1007/11681878_14.
28. Ji Z, Lipton ZC, Elkan C. Differential Privacy and Machine Learning: a Survey and Review. 2014. Accessed 16 May 2023. https://arxiv.org/abs/1412.7584
29. Papernot N, Thakurta AG. How to deploy machine learning with differential privacy. NIST. Accessed 16 May 2023. https://www.nist.gov/blogs/cybersecurity-insights/how-deploy-machine-learning-differential-privacy
30. Wood A, Altman M, Bembenek A, et al.. Differential privacy: a primer for a non-technical audience. *Vand J Ent Tech l*. 2019;21:69. https://doi.org/10.2139/ssrn.3338027.
31. Honkela A, Das M, Nieminen A, Dikmen O, Kaski S. Efficient differentially private learning improves drug sensitivity prediction. *Biol Direct*. 2018;13:1–12. https://doi.org/10.1186/s13062-017-0203-4.
32. Islam MM, Mohammed N, Wang Y, Hu P. Differential private deep learning models for analyzing breast cancer omics data. *Front Oncol*. 2022;12:2816. https://doi.org/10.3389/fonc.2022.879607.
33. Aldaghri N, Mahdavifar H, Beirami A. Federated learning with heterogeneous differential privacy. 2021. Accessed 21 April 2023. https://arxiv.org/abs/2110.15252v2
34. Rivest R, Adleman L, Dertouzos M. On data banks and privacy homomorphisms. *Found Secur Comput*. 1978;4:169–180.
35. Gentry C. Toward basing fully homomorphic encryption on worst-case hardness. In: Rabin T, ed. *Advances in Cryptology – CRYPTO 2010*. *Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer; 2010:116–137 https://doi.org/10.1007/978-3-642-14623-7_7.
36. Gentry C. Computing arbitrary functions of encrypted data. *Commun ACM*. 2010;53:97–105. https://doi.org/10.1145/1666420.1666444.
37. Gentry C. *Fully Homomorphic Encryption Using Ideal Lattices*. 2009;169–178.
38. Gentry C. *A fully homomorphic encryption scheme*. Stanford, CA: Stanford University; 2009 [PhD Thesis].
39. Peikert C. A decade of lattice cryptography. *Found Trends Theor Comput Sci*. 2016;10:283–424. https://doi.org/10.1561/0400000074.
40. Kahrobaei D, Wood A, Najarian K. Homomorphic encryption for machine learning in medicine and bioinformatics. *ACM Comput Surv*. 2020. https://doi.org/10.1145/1122445.1122456.
41. Benarroch D, Brakerski Z, Lepoint T. FHE over the integers: decomposed and batched in the post-quantum regime. In: Fehr S, ed. *Public-Key Cryptography – PKC 2017*. *Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer; 2017:271–301 https://doi.org/10.1007/978-3-662-54388-7_10.
42. Podschwadt R, Takabi D, Hu P. SoK: Privacy-preserving Deep Learning with Homomorphic Encryption. arXiv:2112.12855, 2021. Accessed May 25, 2023. https://arxiv.org/abs/2112.12855
43. Wood A, Najarian K, Kahrobaei D. Homomorphic encryption for machine learning in medicine and bioinformatics. *ACM Comput Surv*. 2020;53:1–35. https://doi.org/10.1145/3394658.
44. Microsoft. GitHub - Microsoft/SEAL: Microsoft SEAL is an easy-to-use and powerful homomorphic encryption library; 2019. Accessed 24 May 2023. https://github.com/microsoft/SEAL%0Ahttps://github.com/Microsoft/SEAL
45. IBM. GitHub – homenc/HElib: HElib is an open-source software library that implements homomorphic encryption. It supports the BGV scheme with bootstrapping and the Approximate Number CKKS scheme. HElib also includes optimizations for efficient homomorphic evalu. Accessed 24 May 2023. https://github.com/homenc/HElib
46. GitHub – TFHE: Fast Fully Homomorphic Encryption Library over the Torus. Accessed 24 May 2023. https://github.com/tfhe/tfhe
47. Seoul National University. GitHub – snucrypto/HEAAN. Accessed 24 May 2023. https://github.com/snucrypto/HEAAN
48. PALISADE Homomorphic Encryption Software Library. 2021. Accessed 24 May 2023. https://palisade-crypto.org/
49. Standard – Homomorphic Encryption Standardization. Accessed 24 May 2023. https://homomorphicencryption.org/standard/
50. Heusinger M, Raab C, Rossi F, Schleif FM, Federated learning methods, applications and beyond. ESANN,. Proceedings – 29th European Symposium on Artificial Neural Networks. *Comput Intell Mach Learn*. 2021;2021:1–10. https://doi.org/10.14428/esann/2021.ES2021-4.
51. Truong N, Sun K, Wang S, Guitton F, Guo YK. Privacy preservation in federated learning: an insightful survey from the GDPR perspective. *Comput Secur*. 2021;110. https://doi.org/10.1016/j.cose.2021.102402 102402.
52. McMahan HB, Moore E, Ramage D, Hampson S, Agüera y Arcas B. Communication-efficient learning of deep networks from decentralized data. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017*. PMLR: W&CP volume 54; 2017.
53. Hanser T. Federated learning for molecular discovery. *Curr Opin Struct Biol*. 2023;79. https://doi.org/10.1016/j.sbi.2023.102545 102545.
54. Hanser T, Bastogne D, Basu A, et al. Using privacy-preserving federated learning to enable pre-competitive cross-industry knowledge sharing and improve QSAR models. In: *2022 Society of Toxicology (SOT) Annual Meeting*; 2022. Accessed 5 April 2023. https://www.lhasalimited.org/Public/Library/2022/SOT Posters/Using privacy-preserving federated learning to enable pre-competitive cross-industry knowledge sharing and improve QSAR models.pdf.
55. Chen S, Xue D, Chuai G, Yang Q, Liu Q. FL-QSAR: a federated learning based QSAR prototype for collaborative drug discovery. *bioRxiv*. 2020:2020.02.27.950592. https://doi.org/10.1101/2020.02.27.950592
56. Zhu W, Luo J, White AD. Federated learning of molecular properties with graph neural networks in a heterogeneous setting. *Patterns*. 2022;3. https://doi.org/10.1016/j.patter.2022.100521 100521.
57. He C, Balasubramanian K, Ceyani E, et al. FedGraphNN: a federated learning system and benchmark for graph neural networks; 2021. Accessed 5 June 2023. https://arxiv.org/abs/2104.07145.
58. Manu D, et al. FL-DISCO: federated generative adversarial network for graph-based molecule drug discovery. In: *IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. Munich, Germany; 2021:1-7. https://doi.org/10.1109/ICCAD51958.2021.9643440
59. GitHub – elix-tech/kmol: kMoL is a machine learning library for drug discovery and life sciences, with federated learning capabilities. Accessed 5 June 2023. https://github.com/elix-tech/kmol
60. Fate. Accessed 5 June 2023. https://fate.fedai.org/
61. Webank. FATE: An Industrial Grade Federated Learning Framework. Accessed 5 June 2023. https://fate.readthedocs.io/en/latest/%0Ahttps://fate.fedai.org/kubefate/
62. Xiong Z, Cheng Z, Lin X, et al. Facing small and biased data dilemma in drug discovery with enhanced federated learning approaches. *Sci China Life Sci*. 2022;65(3):529–539. https://doi.org/10.1007/s11427-021-1946-0.
63. Warnat-Herresthal S, Schultze H, Shastry KL, et al. Swarm Learning for decentralized and confidential clinical machine learning. *Nature*. 2021;594:265–270. https://doi.org/10.1038/s41586-021-03583-3.

Patterns

CellPress
OPEN ACCESS

Review

Privacy preservation for federated learning in health care

Sarthak Pati,[1,2,15] Sourav Kumar,[3,15] Amokh Varma,[3,15] Brandon Edwards,[4,15] Charles Lu,[3,5] Liangqiong Qu,[6] Justin J. Wang,[7] Anantharaman Lakshminarayanan,[8] Shih-han Wang,[4] Micah J. Sheller,[4] Ken Chang,[9] Praveer Singh,[10] Daniel L. Rubin,[7] Jayashree Kalpathy-Cramer,[10,16] and Spyridon Bakas[1,2,11,12,13,14,16,*]

[1]Center for Federated Learning in Medicine, Indiana University, Indianapolis, IN, USA
[2]Division of Computational Pathology, Department of Pathology and Laboratory Medicine, Indiana University School of Medicine, Indianapolis, IN, USA
[3]Department of Radiology, Athinoula A. Martinos Center for Biomedical Imaging, Massachusetts General Hospital, Boston, MA, USA
[4]Intel Corporation, Santa Clara, CA, USA
[5]Center for Clinical Data Science, Massachusetts General Hospital and Brigham and Women's Hospital, Boston, MA, USA
[6]Department of Statistics and Actuarial Science, University of Hong Kong, Hong Kong, China
[7]Department of Biomedical Data Science, Radiology, and Medicine (Biomedical Informatics), Stanford University, Stanford, CA, USA
[8]Institute for Infocomm Research, Agency for Science Technology and Research (A*STAR), Singapore, Singapore
[9]Department of Radiology, Stanford University, Stanford, CA, USA
[10]University of Colorado School of Medicine, Aurora, CO, USA
[11]Department of Biostatistics and Health Data Science, Indiana University School of Medicine, Indianapolis, IN, USA
[12]Department of Radiology and Imaging Sciences, Indiana University School of Medicine, Indianapolis, IN, USA
[13]Department of Neurological Surgery, Indiana University School of Medicine, Indianapolis, IN, USA
[14]Department of Computer Science, Luddy School of Informatics, Computing, and Engineering, Indiana University, Indianapolis, IN, USA
[15]These authors contributed equally
[16]Senior authors
*Correspondence: spbakas@iu.edu
https://doi.org/10.1016/j.patter.2024.100974

THE BIGGER PICTURE Significant improvements can be made to clinical AI applications when multiple health-care institutions collaborate to build models that leverage large and diverse datasets. Federated learning (FL) provides a method for such AI model training, where each institution shares only model updates derived from their private training data, rather than the explicit patient data. This has been demonstrated to advance the state of the art for many clinical AI applications. However, open and persistent federations bring up the question of trust, and model updates have raised considerations of possible information leakage. Prior work has gone into understanding the inherent privacy risks and into developing mitigation techniques. Focusing on FL in health care, we review the privacy risks and the costs and limitations associated with state-of-the-art mitigations. We hope to provide a guide to health-care researchers seeking to engage in FL as a new paradigm of secure and private collaborative AI.

SUMMARY

Artificial intelligence (AI) shows potential to improve health care by leveraging data to build models that can inform clinical workflows. However, access to large quantities of diverse data is needed to develop robust generalizable models. Data sharing across institutions is not always feasible due to legal, security, and privacy concerns. Federated learning (FL) allows for multi-institutional training of AI models, obviating data sharing, albeit with different security and privacy concerns. Specifically, insights exchanged during FL can leak information about institutional data. In addition, FL can introduce issues when there is limited trust among the entities performing the compute. With the growing adoption of FL in health care, it is imperative to elucidate the potential risks. We thus summarize privacy-preserving FL literature in this work with special regard to health care. We draw attention to threats and review mitigation approaches. We anticipate this review to become a health-care researcher's guide to security and privacy in FL.

INTRODUCTION

The health-care domain has always dealt with privacy concerns and threats due to the sensitive nature of the information in the domain. For example, there have always been unauthorized access to medical records (which can be mitigated by requiring string access controls, user authentication, and audit logs),[1] insider threats that lead to misuse or inappropriate disclosure of health-care data (which can be mitigated by specific training, implementing policies that allow data access to employees where it is needed, and monitoring access to data),[2] and data breaches and/or cyber attacks (which can be mitigated by encrypting data, implementing robust network security policies, and regular security monitoring).[3] Although these challenges are critical, they have been documented and identified and are well studied. For the purposes of this review, we will be focusing on the use of advanced computational techniques in health care, where the privacy issues are more nuanced and their associated mitigation strategies are not that well studied compared to other fields.[4] The use of such tools (such as artificial intelligence [AI]) can make addressing these concerns more complicated due to the possibility that interactions with the application may leak information about the data used to train it.[3] Since health-care data are almost always tied with specific regulatory provisions,[5–7] privacy concerns of AI applications in this domain need a deeper understanding of the technical issues at hand, especially to provide guidelines for computational researchers developing algorithms and solutions in this field. This article aims to provide privacy and security guidelines for both computational researchers developing AI solutions in health care and regulatory authorities, so that they are mindful of both traditional information security concerns.

AI approaches have shown great promise for augmenting clinical workflows.[8] However, large and diverse datasets are required to train robust and generalizable AI models for clinical applications.[9–15] One method to acquire sufficient data is through multi-institutional collaborations, currently following a paradigm of centrally sharing data, also known as "data pooling"[16–20] (Figure 1A). However, such centralized data collection is not always possible due to various factors, such as patient privacy concerns, prohibitive costs of central data management and storage, and institutional or even regional data-sharing policies.[21,22] Federated learning (FL) is an alternative approach to the data pooling paradigm for multi-institutional collaborations that begins to address some privacy concerns, since model learning is performed locally at each institution and only the resulting local model parameters are shared (Figure 1B).

Although FL allows for training an AI model across private data without requiring that data to be shared, there are still questions that remain regarding the need and the way one needs to protect against leakage of information about these private data via the model updates shared throughout the FL workflow. There is hope that through the incorporation of additional security and privacy technologies into FL, a level of security and privacy can be achieved that will enable a greater degree of trust in the resulting federation. Many factors in addition to trust can prevent institutions from participating in FL training, such as coordination and overhead of data preparation, institutional information security, and compute hardware requirements. However, the use of more secure and private FL frameworks toward increasing trust in the system is expected to enable a more diverse collection of clinical institutions willing participate in FL projects. The models that result from such collaborations can potentially benefit from the data diversity, resulting in better model generalizability. In particular, secure and private FL has hope to greatly benefit collaborations in domains with stringent policies around data sharing, such as is the case for health care.[14,21–28] Some literature exists reporting on general vulnerabilities of FL[29,30] and even exploring privacy and security concepts related to FL,[31–33] albeit without providing a focus on their implications in the health-care domain. We further note a survey of open-source frameworks enabling FL, although without adequately exploring concepts of privacy.[34] Thus far, works that survey privacy issues related to FL in medicine[26,35–39] have provided details about FL in health care while not adequately expanding on the specific privacy threat associated with each attack.

In this work, we provide an overview of the current privacy threats and associated threat mitigations for FL workflows[40] while keeping the health-care context in mind. We summarize the key factors involved in determining the nature of privacy violation that can be related to each threat. Finally, we categorize the privacy threat mitigation technologies into distinct categories, and for each group we discuss what threats they address, as well as the costs associated with each mitigation technique. We hope that model developers having domain expertise can use this work to make more informed decisions with regard to patient privacy when using FL to train their models, and we hope to provide the context necessary for researchers to design experiments that most effectively advance our understanding of the problems and potential solutions for their domain.

In order to facilitate the focus on privacy concerns to FL deployments specifically in the health-care setting, we start by briefly introducing the concept of FL in health care. We then list the primary assets in AI that need to be protected in such an FL schema and then proceed with introducing a taxonomy of the potential threats to these assets, using the well-known confidentiality, integrity, and availability ("CIA") triad.[41] Notably, for each compromise, we consider the way in which it could adversely affect the collaborators. Finally, we discuss the mitigation methods that exist for these threats by considering how they reduce the impact of the threat, as well as the costs that are associated with the usage of the mitigation.

FEDERATED LEARNING

FL describes a collaborative approach to train AI models across decentralized collaborators (e.g., client servers on health-care sites) without directly sharing any training data between them.[22–25] This approach differs from the standard/traditional approach followed during training of AI models, which typically assumes that the data and model reside on a single, centralized location (see Figure 1 for an illustration). FL allows multiple institutions to train a single aggregate model without explicitly sharing any individual institution's patient data outside of that institution.

For example, to train a neural network (NN) with FL, an NN architecture needs to be chosen and code to implement training on this architecture incorporated into the system code to be distributed to all collaborators. We consider the use case of tumor boundary detection[14,22,24] for our explanation, where the trained

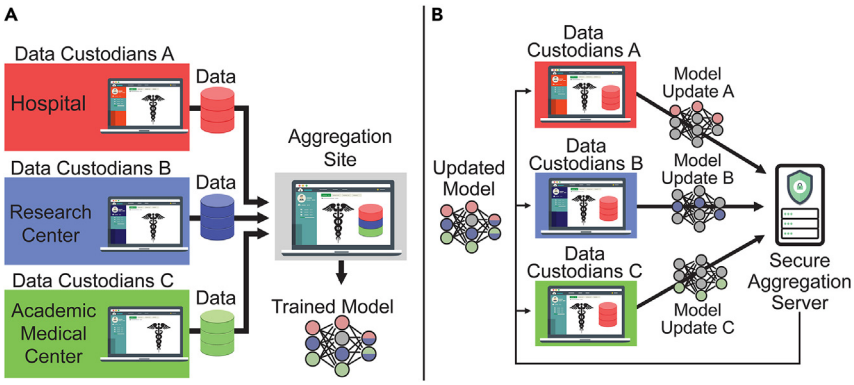


Figure 1. Illustration of different collaborative learning approaches

(A) Using a data-sharing paradigm, where data from the three individual data custodians are shared to a central data aggregation site for training and (B) using federated learning, where the training happens at each individual data custodian and only the model updates are shared to a secure aggregation server for combination.

AI model has an image as input and an image as output. The input image represents a clinically acquired scan, and the output is meant to identify regions of pathology associated with the presence of a cancerous tumor. The typical FL system consists of multiple participant sites, all independently connected via the network to a (central) "aggregator" node as depicted in Figure 1. To start the process, model initial weights are chosen at the aggregator as the initial global model and distributed via the network to all participants.

A typical "round" of federated training proceeds as follows. All participants perform training and validation on the aggregate model using their local data and compute resources and send their local model update as well as local validation scores to the aggregator node. The aggregator then aggregates all received local model updates and local validation scores, to form the updated aggregate model and aggregate validation scores for that round. This updated aggregate model is then sent back to all participants to initialize the next round of training. The complete course of FL training consists of multiple FL rounds, with a stopping/convergence condition and model selection criterion enforced by the aggregator using the model validation scores.

We take the opportunity to point out that model updates and validation scores should be viewed as a potential way to obtain information about the training input data. The local training described above consists of iteratively (1) passing batches of input images into the model, (2) measuring how well the model did at predicting the correct pathology locations, and (3) adjusting the model weights using small corrections obtained through the NN backpropagation process. Each batch of input data shifts the model weights using the information in that batch for performance to increase. This influence can potentially lead to information about the batches being detectable in the resulting model weights, as we will see in subsequent sections. The same is true for the validation metrics, although to a lesser degree. The intuition here is that patterns may be unique enough for one particular data sample that the results of training or validating on it could indicate its presence.

By being an approach in which data from one institute are never seen by another, it helps alleviate some privacy and security concerns, especially when dealing with sensitive medical data.[25,42] This potentially enables collaborations to be formed with larger and more diverse training data, which can result in trained models that generalize better.[22,43] In most FL algorithms, each institution independently performs training on their respective (local) data and provides the results of their computation (usually weights of a model) to be incorporated by the FL workflow (such as peer-to-peer aggregation or based on an aggregation server[25]; see Figure 1 for an illustration of FL using an aggregation server). Details of the typical central aggregator-based version that we provide as an example above can also differ depending on the implementation, for example, how model aggregation is performed or how the clients are selected in each round.[44–47]

FL has the potential to play a critical role in the next generation of privacy preservation strategies for health care, as evidenced by several recent high-impact studies.[14,22,24,25,27,48,49] The degree to which an FL system can be thought to be "secure and private" largely depends on the additional security and privacy technologies that it incorporates into the system. Concerns need to be addressed to mitigate malicious code execution, such as processes running at the collaborating institution's infrastructure that ex-filtrate their raw data or execute malicious forms of training. However, there are also threats from collaborators during FL that may attempt to extract information about the training data via the model updates that are shared between collaborators in the workflow. These updates may indeed leak information if in the hands of an adversary, as was suggested above.

SECURITY AND PRIVACY FOR FL IN HEALTH CARE

Importance of privacy in health care

Privacy attacks during the course of an FL collaboration result in exposure of data, model, or code (see Figure 2). Data privacy in health-care scenarios is crucial toward ensuring confidentiality and ethical handling of protected health information (PHI). Identifying robust mitigation techniques corresponding to specific threats, as well as implementing security measures, adopting encryption technologies, and adhering to privacy regulations (such as Health Insurance Portability and Accountability Act [HIPAA] from the United States[5] and General Data Protection Regulation [GDPR] from the European Union[6] or Digital Information Security in Healthcare Act [DISHA] from India[7]), is essential for safeguarding sensitive patient information and maintaining trust in health systems. Key health-care scenarios relevant to privacy preservation related to FL include (1) electronic health records (EHRs), which contain verbose textual data about a patient's medical history, procedures, diagnoses, medical scans, treatments, and medications, exposure of which would be a subject to patient confidentiality loss; (2) wearable devices, which collect health-related, fitness, and nutrition patient data directly associated with PHI; (3) local/institutional data repositories such as Biobanks and picture archiving and communication

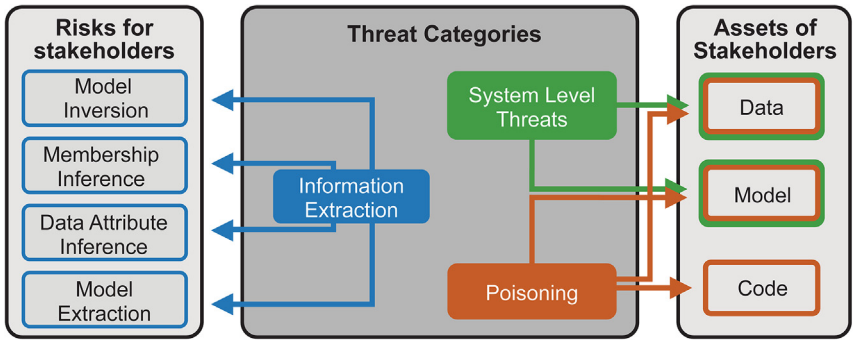


Figure 2. Illustration of overall privacy threat categories and their associated risks for stakeholders in a federated learning system

The "system-level threats" (in the green box) target the data and model; the "poisoning" attacks (in the orange box) attempt to expose the data, model, and code; and "information extraction" (in the blue box) targets model inversion, membership inference, data attribute inference, and model extraction.

systems (PACSs), including both data from routine clinical practice, clinical trials, and emergency services and unpublished research outcomes; (4) health insurance/billing; and (5) healthcare policy.

Assets

There are three AI-specific assets to consider with regard to security and privacy around FL deployments. These are (1) the entire data cohort to be used for training; (2) the quantitative performance evaluation metrics that signify model performance, generated against both the local and the aggregated model updates; and (3) the model parameters themselves, for both the local and the aggregated model updates. In addition, the system on which the AI system is being deployed includes three additional assets that should be considered: (1) the hardware on which all the computations (i.e., training and FL aggregation) occur, (2) the actual source code for model training and FL execution, and (3) any additional metadata or configuration information that can be defined either in memory or as files on disk. Table 1 offers a summary of each asset and the properties we propose to address in this work. In the following sections, we will proceed through each of the CIA properties and elaborate on their meaning. On a high level, hardware protection is expected to already be in place, participants are expected to report correct validation metrics with what we see as minimal privacy consequences otherwise, and we see minimal privacy impact to participants dropping out and/or network connections being lost and so do not address issues of unavailable FL system resources.

Confidentiality

By "confidentiality" we refer to the degree to which the asset is hidden from others. As an example, if collaborator A sends their model update to the aggregator using transport layer security (TLS), then A has some assurances as to the confidentiality of the update while it is being transmitted to the aggregator. Once the aggregator receives the update and the decryption in TLS is performed, the degree to which the update remains confidential depends on what the aggregator code logic does with the update (for example, it could simply broadcast it to others), as well as how protected the aggregator processes are (e.g., code, memory, hardware instructions) from inspection by other processes and users on the aggregator infrastructure. These issues are exacerbated in the case of peer-to-peer aggregation methods,[25] where each collaborator performs weight aggregation on the model updates received by a peer-collaborator.

The confidentiality of any part of the data cohort and model parameters is considered here, with a break in confidentiality of either representing either a privacy violation or a leak of intellectual property (IP). Exposure of the complete data cohort in general can be a privacy violation, and model parameters can be used in an effort to reverse engineer training data.[50,51] Both model parameters and data can be considered IP, as both have value to organizations. We also consider exposure of approximations to these assets (i.e., data and model parameters) as a break in confidentiality. In the health-care setting, an approximation of a medical image may violate privacy just as much as the originally acquired image, and an approximation to model parameters may preserve enough model utility as to continue to hold a great deal of value as IP. The confidentiality of quantitative evaluation measures of model performance will also be considered, because such scores can be used to approximately recover the parameters of the model being evaluated, which, as discussed earlier, can further lead to approximate recovery of the underlying data.

Although physical isolation of hardware as well as the confidentiality of system code and additional files may be a general concern, they are considered out of the scope of this review, which instead focuses on the privacy and security aspects of medical data to be used in FL.

Table 1. Security and privacy assets in an FL system, including the CIA properties we propose to address in this work

Asset	Confidentiality	Integrity	Availability
Training data	✓	✓	✗
Quantitative metrics	✓	✗	✗
Model parameters	✓	✓	✗
Hardware	✗	✗	✗
Source code	✓	✓	✗
Additional files or information	✓	✓	✗

Integrity

By "integrity" we mean the degree to which the asset is precisely what it was expected to be. As an example, collaborator A may want to establish the integrity of the code running on the compute infrastructure of collaborator B. In some rare cases, though, A may trust B and their infrastructure to the extent that A is confident of such integrity.

The integrity of system hardware, i.e., being able to rely on the proper execution of hardware for a given hardware state, will be considered as out of scope of this work. This work shall instead focus on the integrity of training data and model parameters, as

well as the integrity of system code and additional files and metadata. Although the integrity of model validation scores could be a concern in the general setting, we will not address this issue here, as the corruption of such scores would primarily be involved in an attack that was attempting to alter model selection (as such scores inform that process) and, as such, is not a significant concern in the health-care domain.

Availability

By "availability" we mean the degree to which an asset is available to use. As an example, a local model update may not be available at the aggregator if the network infrastructure at that collaborator is down, or the entire federation might get hampered if the network access at the aggregator level is lost.

Addressing availability issues for all listed assets in the previous section is considered out of the scope of this work, as we start with the assumption that the networking infrastructure for all collaborators in an FL system focusing on health care data is controlled by the respective clinical entities and, as such, is reliable and stable throughout the computation process.

THREATS TO PRIVACY DURING FL

In this section, we will be discussing the various threats to privacy in an FL system, an illustration of which can be found in Figure 2.

System-level threats

The threats in this category involve an adversary gaining access to either the raw data or the model weights. The adversary can acquire direct access through different means, such as privilege escalation and/or physical access, but always ends up with the requisite assets in their raw exact form. In contrast, all other threat categories in this section involve an adversary deriving approximations to these raw assets, such as extraction of training data information via the model weights[52] or manipulating their local data and/or training algorithm to exacerbate such an attack.

(1) Data ex-filtration: this involves an adversary obtaining access to the raw data (such as health-care scans or medical records) of one or multiple collaborators. The nature of privacy violation in this case is given by a patient's right to not have their data given to anyone whom they did not explicitly authorize to have them.[5,6]

(2) Model ex-filtration: this involves an adversary obtaining access to the weights and biases of a local model update or global model aggregate(s). The primary concern would be either that the model represents IP or that the model could be used to extract information about the raw data used to train it.[52] Therefore, the nature of the privacy violation in this case can be the peculation of IP or could be any of the privacy violations[5,6] that can come from a successful attack to extract information about the training data from the model.

Information extraction

There are different types of privacy attack objectives related to the extraction of information about the training data from the model weights during, or at the conclusion of, the model training process. Multiple studies show that rare or unique parts of the training data are unintentionally retained by NNs.[52–54] The trained model weights transferred from a local institution (as well as the aggregate models they become a part of) can therefore be potentially exploited by any user with access to the model, taking advantage of this unintended memorization to gain sensitive information about the dataset being used by other collaborators in a federated setting. Some of the examples of such threats are illustrated below, and clinical researchers should consider the privacy impact of each of these threats independently.

After the attacks that use model access to approximate information about the data used to train it, we include one more attack that instead uses access to the model validation scores to approximate the model itself. This is an attack that is of concern in a federation that was otherwise acting to control access to the model (as IP) and, in addition, is a concern because such an approximate model can also be further used to approximate the training data themselves.

(1) Model inversion: such attacks involve an adversary with the ability to query the model or observe a model update, constructing a data sample meant to approximate an actual sample in a collaborator's dataset.[55–57] Although the accuracy of these reconstructions can vary, the exposure of such a reconstruction may violate a patient's privacy if features in the reconstruction are highly correlated to the original samples (for example, chest radiographs[48]). One form of this attack is carried out by an adversary with access to any version of the model and has been demonstrated outside of the FL context. Fredrikson et al.[56] showed how an adversary may use the prediction confidence values to approximate associated faces in a facial recognition system. Zhang et al.[58] demonstrated that an adversary with access to the model weights can approximate training examples for various classes using some auxiliary knowledge, such as blurred images from each of those classes. Other forms of this attack utilize single FL model updates from a particular collaborator and may make certain assumptions about the setting and attempt to approximate aspects of local training, such as batch normalization statistics or what labels were used for the batches.[59] State-of-the-art versions of these stronger attacks have demonstrated pixel-perfect reconstruction of images[60] when the attacker has access to local updates created with few samples, so that each sample has more relative influence. However, most FL round model updates are processed using many local data samples so that individual sample influence is reduced. In these cases, it is more difficult to reconstruct an exact training data sample from a local model update and even harder to reproduce one from the global aggregate model(s) it is included in. An overview of model inversion attack implementations and defense approaches is already described in prior work.[61] Advancements in these attacks continue to be put forward, and works that demonstrate such attacks in the setting of FL for medical models that demonstrate successful approximation of

hidden batch normalization statistics, for example,[62] acknowledge the importance of understanding such threats in these settings.

(2) Membership inference: membership inference is the process by which an adversary has possession of a particular data sample and is attempting to infer whether it was included in the training set of the model.[50,51,63] Exposure of whether a specific patient's data sample was used in training may be sensitive information, for example, if the presence of that sample implied something about the sample custodian (i.e., the dataset consisted of information about known felons or the dataset consisted of patients with a certain type of cancer). Success in accurately predicting which samples were involved in training a model is correlated with the degree to which a model encodes sample-specific information during training.[64,65] Due to this fact, membership inference attack success measurements are thought of as building blocks for state-of-the-art tools for generically determining the amount to which a model leaks information about its training data.[66] Clinicians and researchers should therefore consider successful membership inference attacks as a privacy concern, regardless of how compelling the concern is regarding the leakage of membership information alone. Such success may indicate that other more concerning attacks may be successful as well. For more details regarding the various membership inference attacks and defenses, see Hu et al.[67].

(3) Data attribute inference: instead of attempting to recover an entire data sample, this type of attack is characterized when an adversary attempts to recover only a subset of the data attributes from particular samples in the training set or, alternatively, the adversary attempts to learn attributes of the training set as a whole (such as aggregate statistical information). Such attacks have been demonstrated outside of the FL setting,[68] as well as within the FL setting with the attacker being an FL participant that only has access to the aggregated model,[50] although the use of alternative aggregation functions other than a simple weighted average could make this more difficult. Here, the attacker may estimate the accuracy of the reconstruction over other possible alternate values for the unknown data fields. Measures of confidence in the reconstruction may play a role in the impact of its exposure. Take the case of an attack to disclose the value ("true" or "false") of the attribute indicating a positive diagnosis of a particular medical condition. A confidence measure of 80% for such an attack, conditioned on a specific gender, location, and age of patient, may be used to assert that 80% of the samples in the training set corresponding to that specific gender, location, and age had a positive test result, which (depending on the characteristics) could be considered as exposure of PHI (defined as any information about health status, provision of health care, or payment for health care that is created or collected by a covered entity [or a business associate of a covered entity] and can be linked to a specific individual[5]) and, hence, identifying specific subjects from the training data. Such confidence measures have indeed been considered in previous studies and are easier to interpret when the feature space is relatively limited, such as when using categorical data with numerical digits (demonstrated by the attack example seeking to recover social security numbers[52]), and alternatives to standard model averaging, such as having collaborators withhold a subset of their local update, have been demonstrated to influence the effectiveness of such attacks.[50] Such confidence measurements would be more challenging to obtain for training sets containing only high-resolution images, for example.

(4) Model extraction: the traditional forms of this attack use the ability to obtain model outputs associated with arbitrary inputs chosen by the adversary[69,70] in order to estimate the model weights after a number of such queries. This is especially related to "black box"-style attacks, where a black box system[71] (basically, any system that does not expose any aspect of the inference process, starting from the data processing to the model weights, and only provides inference results) is used to generate outputs for multiple inputs in the form of predictions or validation scores. Depending on the amount of information provided by validation scores during FL training, such attacks based on validation scores alone may be more limited in their ability to approximate the model parameters, and with increasing numbers of input samples, the adversary can obtain a closer approximation to the model under attack. Such efforts to attain model weights, despite an appearance that access to such weights is restricted, could result in loss of valuable IP to a participant during a medical model federation.

Poisoning

This type of threat is specifically relevant for an adversary that is part of an FL system, such as a collaborator. A collaborator during the FL training process can maliciously alter their computations (by virtue of changing the local code, or data, or model) in order to magnify the effect of one of the attacks listed above on the private assets of others in the system, for example, by increasing data memorization. As such behavior serves to modify the model behavior in a malicious way, this falls under the category of model poisoning. Such advanced attacks are demonstrated for standard FL (where aggregation is simple model averaging) in Nasr et al.[51] and Hitaj et al.[72] and provide for a significant increase in the severity of the baseline attack. Alternative mechanisms for model averaging, such as limiting the portion of the local updates shared to the aggregator or employing a more robust aggregation, such as median rather than mean, can have an effect on the adversary's success but also affect the convergence properties of the global model. As such, these alternatives need to be considered together with the potential drawbacks to final model utility. Therefore, model poisoning can be considered fundamentally as a security attack by nature, since the primary attack vector is system asset corruption.

APPROACHES TO MITIGATE PRIVACY THREATS DURING FL

The amount of information that is exposed by the attacks described in the previous section can depend on the adversary's

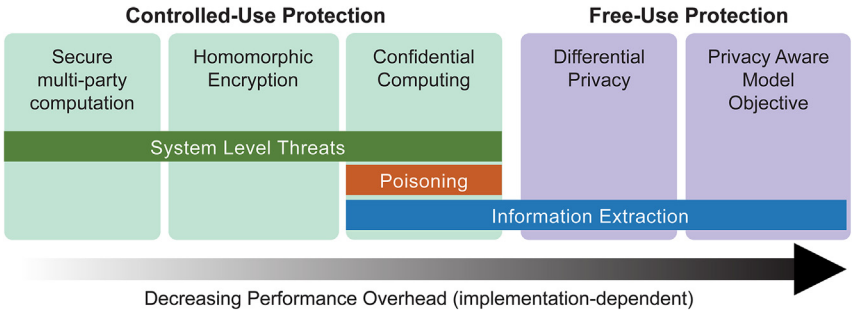


Figure 3. Various mitigation strategies and the respective privacy threats they aim to address

The "controlled-use protection" mechanisms of secure multi-party computation and homomorphic encryption are used to mitigate the system-level threats. The "free-use protection" mechanisms of differential privacy and privacy-aware model objective can be used to mitigate against information extraction. Confidential computing, on the other hand, can be used to counteract poisoning attacks in addition to the system-level threats and information extraction. The computational performance decreases as we move from secure multi-party computation to homomorphic encryption, confidential computing, differential privacy, and, finally, the privacy-aware model objective, being the most computationally inefficient.

level of access to the various assets in an FL system. As a basic requirement for considerations in health-care information technology (IT) infrastructures, we will assume that all FL systems will incorporate basic security measures, such as authentication to verify identities, support for encrypted communication, and other access control mechanisms meant to prevent exposure of assets to those who do not need to use them. However, since FL is a collaborative procedure, involving many parties who do need to handle assets in the system, the focus of the threat mitigations in this paper is to share assets and computational duties on those assets while mitigating the threats that exist when doing so with potentially untrusted parties. As an example, this includes the use of the final model, hosted by or with an untrusted user. As mentioned in the last section, such access can be sufficient for extracting model IP and/or carrying out membership or attribute inference attacks.

Some mitigations provide confidentiality of data while being used for computation, and others provide assurances that system code preapproved by participants matches identically to that being used at execution time. In addition, when an asset will be exposed due to requirements for its use, there are technologies to employ before the release of that asset to help mitigate a potential information extraction attack. We therefore consider two broad non-overlapping categories for the technologies that can be used during FL to mitigate the threats discussed in the previous section, which can be combined as needed (for an illustration, please refer to Figure 3): (1) controlled-use protection and (2) free-use protection technologies.

Controlled-use protection

These methods perform little to no alteration of the asset, but instead provide a way for potentially untrusted entities to perform computations on those assets without accessing the assets themselves, potentially controlling what computations are performed. None of the solutions in this category provide any protection against attempts to reverse engineer information from the outputs of the computations. For example, cryptography-based algorithms can be used to carry out remote algorithm execution with limitations on exposure of information to the remote parties during their calculation tasks (software-based confidentiality, such as homomorphic encryption [HE]).[73–75] In addition, specialized hardware solutions can provide computational resources while limiting the exposure of data used in the computation, by incorporating integrity checks during code execution to ensure that the appropriate code is being executed (hardware-based confidential computing [CC] with trusted execution).[76–78]

The first two solutions belonging to this category, i.e., secure multi-party computation (SMPC) and HE, allow for outsourcing computation with confidentiality of the inputs, intermediate results, and outputs and are implemented via software. These solutions may or may not provide assurances as to the integrity (correctness) of the computation. The other solution (i.e., confidential computing) is a hardware-based approach providing confidentiality of inputs, intermediate results, and outputs, while also providing assurances as to the integrity of the computation,[14] though usually with different assumptions as to in which circumstances protection is provided. CC solutions can generally help mitigate all threats listed in the previous section. Due to this broad threat coverage, combining CC with mitigation strategies that tackle broad information extraction threats (i.e., free-use protection category) can ensure the most robust security design. None of the solutions in this category provide any protection against reverse engineering of inputs to the computation itself. In general, there is a cost associated with these solutions that comes in the form of either increased computation and communication or special hardware requirements, in the case of CC.

Secure multi-party computation

SMPC[73–75] is an umbrella term that refers to a set of algorithms used to allow multiple entities to collectively calculate some function with controls as to what is exposed to the individual entities regarding both the inputs and the outputs of the function. As an example, suppose we wanted the aggregator in an aggregator-based FL[22,24,25] to know the output of a function of two inputs, and we want this to be computed using an input from each collaborator of a two-collaborator federation. The most basic example of an SMPC protocol here would be to use the trusted third party (TTP) protocol[79] to allow a third party (trusted by each collaborator and the aggregator) to take the inputs from each collaborator and send the output of the function applied to these inputs to the aggregator, without sharing either collaborator's inputs nor sharing the output to anyone except the aggregator. Using TTP, each collaborator would not learn anything new by participating in the protocol, and the aggregator only learns whatever can be deduced from the output of the function that was provided to it. Due to its simplicity, as well as the minimal

information exposure involved, TTP is the benchmark protocol used to evaluate the properties of all other SMPC protocols.[79] Ongoing research is exploring the use of SMPC as a privacy enhancement to FL on medical data either by helping to prevent malicious models or by improving the confidentiality of model aggregation[80,81] or by making progress on the overhead that is incurred by its use.[82,83] These protocols can incur high computational and network communication overhead costs, as significant computation can be required to obfuscate information by encrypting/encoding and splitting into parts to avoid recovery, and significant communication protocols can be involved in order to coordinate the compute on the information pieces as well as combining the results to recover the function output without revealing information to unwanted parties in the process.

Homomorphic encryption

Although SMPC allows multiple institutions to jointly evaluate a function without needing to share their respective private inputs, the design of an SMPC protocol needs to take into account the specific function whose output is desired from the protocol, and a good deal of the protocol itself is dedicated to obfuscating the inputs. In contrast, HE[84–87] is a type of data encryption (and therefore provides cryptographic guarantees of confidentiality) that allows for generic computation on the data when in its encrypted form. The result of a computation on the encrypted data, when decrypted, is identical (or very close) to the result of the same computation performed on the unencrypted data. One benefit over SMPC is that multiple adversaries cannot collude to significantly increase the threat. However, the encryption for HE requires keys, and so key management is a necessary consideration here. HE by design provides a robust privacy solution for the application of FL.[88] However, almost all efforts in this regard suffer from a huge computational cost, and even an incremental increase in the data size (or in the NN layers) leads to an exponential increase in runtime. As such, more work needs to be done on improving the computational efficiency to render this approach practical for modern NN architectures. Although we list this as a software-based approach, efforts are ongoing to provide hardware acceleration for HE[89,90] (https://www.darpa.mil/news-events/2021-03-08), which has the potential to significantly reduce the overhead of this solution. It can be said that HE in its present technological development is most suited to those applications that are not time sensitive, where it can offer an extremely secure form of privacy preservation solution. A recent work by Froelicher et al.[91] shows the success of HE in providing truly private federated evaluation for applications within oncology and medical genetics, and work by Chen et al.[92] demonstrates success using HE for model aggregation during FL when transfer learning is used to reduce the size of the model weights that are processed using HE.

Confidential computing

In the previous points, we described methods that use encryption, encoding, or secret sharing to increase data confidentiality during computation. Alternatively, such confidentiality can also be obtained by means that are hardware enabled with so-called hardware-based CC with trusted execution,[78] though usually there are different assumptions here as to in which circumstances protection is provided. In CC, processes can be run inside so-called "enclaves" that essentially serve as a trusted third party in the sense that we used in the points related to SMPC and HE (not even privileged users on the system can access the memory or alter the execution). Code to run an algorithm can be put into the enclave, encrypted data can be passed in and unencrypted inside the enclave, then the algorithm can execute on the inputs and the result can be encrypted and passed out. In contrast to the previous solutions, these enclaves generally have the ability to attest to the fact that the code run inside the enclave was precisely what was expected, providing assurance as to the correctness of the result. Trust in the CC itself depends on trust in the hardware vendor that designs and distributes it, and trust in attestation will depend on trust in those who implement the service that carries it out. This feature allows for trust in all components of the FL system provided they are run with CC.[26] For example, during FL, running local training at a particular collaborator with CC can help prevent an insider adversary on that compute infrastructure from modifying their training code to compute updates using only a few data samples, in order to increase the ease with which another adversary could extract information about those few samples from the local updates sent from this collaborator. In addition, CC is more scalable and provides faster computation time compared to SMPC and HE.

Free-use protection

These approaches do nothing to protect data confidentiality while the computations are conducted and do nothing to ensure that the computations proceed as intended. However, they have the advantage of limiting how much of the result of the computation can be used to infer information about the original inputs. Therefore, solutions in this category are ideal for mitigating the threats of the previous section in the information extraction category. The costs associated with these solutions are increased computation and a reduction in the asset's utility due to its modification. For example, we will see that differentially private model training is a modification of a standard training algorithm that reduces the ability for someone with free access to the resulting model to carry out an information extraction attack that exposes information about the training data used to create it.

Differential privacy

When using the mitigation strategies described in the previous section for model training during FL, we are able to prevent data exposure during training. However, these techniques do nothing to prevent an adversary from using the trained model to reverse engineer information about (memorized/learned) training data, as is possible in a membership inference attack.[93] Differentially private model training, however, is a common approach toward mitigating the degree to which a model memorizes individual contributions to the data during training. It does so by introducing randomization during training to obfuscate the influence of these individual contributions while being able to learn over the data as a whole. DP algorithms come with privacy guarantees that relate to the likelihood that any single data point can be detected.[93]

An algorithm can be loosely defined as "differentially private" if the output of the algorithm cannot be used to distinguish whether a particular contribution to the data is present in the dataset used as input for the algorithm training.[93] Common examples of what type of contribution DP considers are those of a single data record or contributions of whole collaborator datasets. While the concepts surrounding DP were generally developed for use in

data analytics, DP training algorithms have become a popular method for addressing user privacy concerns in AI.[94,95]

In the federated setting, DP algorithms can be used independently to train local model updates (local DP FL) or instead for the global consensus model aggregation (global DP FL). In local DP FL, each participating institution applies a DP training algorithm to perform their local training.[96] Here, the local model updates sent to the aggregator are produced with a DP privacy guarantee. This may be desired when the entity administering or running the aggregator is not trusted to prevent privacy attacks on its infrastructure. For global DP FL, the aggregation of model updates is made DP but there may be no guarantee with respect to the privacy of the local model updates handled by the aggregator.[97] If trust in the aggregator infrastructure is not already established, this may be done through the use of privacy solutions discussed in the previous section. Global DP FL is preferable to local, as it allows more data (all collaborator updates) to be combined before noising, which in principle improves the utility that is obtained for a given privacy level.[98]

Although DP has started gaining traction for deep-learning applications in medicine, it comes at the cost of a reduction in "model utility," which defines how well the model generalizes to new data when deployed,[99] as well as increased computation.[94] The model utility reduction comes from the noise addition during the training process, and the increased computation relates to potential changes in the way the training utilizes the underlying computational framework,[100] in addition to the potential need for more rounds of training. Importantly, DP training in FL could inhibit the use of data quality checks from specific collaborators, as privatized local model updates at the aggregator may mask signals that would otherwise indicate issues.

Survey articles for DP[101–104] can help to summarize the various approaches, best practices, and future research needed. However, more work is needed to understand the trade-offs associated with specific use cases, such as how much utility loss will be incurred at a given privacy level. We find in recent work[105–109] on DP FL training in medicine that federated training using DP (at $\epsilon = 4$, for example) can reach within 5% of the scores that would be achievable if DP were not used. As more research is done across different datasets and model architectures, a better understanding will form around how well these initial results will generalize. The privacy achieved by a DP algorithm should also be carefully considered. Most papers explore only ϵ values (lower indicates more privacy) greater than 1, and many explore ϵ values that are much greater. Since the worst-case odds of privacy exposure for an ϵ-DP algorithm is $e^{\epsilon} : 1$, the value $\epsilon = 4$, for example, is associated with worse than 50 : 1 odds of privacy exposure.

Another complicating factor is the difficulty for data custodians to understand the privacy guarantee associated with the use of DP training, as it is very technical. In addition, the likelihood of specific privacy threats is even less likely to be understood until more research is done. This makes the proper balance of utility loss against true privacy concerns difficult to reason about, which is a very important aspect for the design of a practical solution.

Privacy-aware model objective

An alternative to making model training DP during FL is to incorporate the incentive against susceptibility to privacy attack into the training objective. Here an "attack model" is used during training in order to simulate a privacy attack on the primary model. Adversarial training is then performed at each collaborator during each round, alternating between improving a locally held attack model and improving the primary model by attempting to minimize a privacy-aware model objective (PAMO) (for example, the sum of the primary model loss and a measure of success for the attack model).[110,111] Such approaches may utilize mutual information estimations in order to establish a measure of success in minimizing information leakage or could use other measures to determine this success. The privacy protections afforded by this approach are similar to that of DP; however, the measure of protection is usually empirically based, in contrast to the theoretical privacy guarantee provided by DP. The costs associated with this approach are that of potential loss in model utility as well as potentially increased computation and overall local training required, as the local training is more complex.

An information conservation approach

In this section we describe a general approach that is meant to address the concern of information leakage from model updates concerning the individual data samples used during training. The mitigation measures discussed here are ones that restrict or obscure information during training, but without formal analysis on how such measures affect information flow. As such, an empirical assessment of the utility loss in the model as a result of each technique must be weighed against the ease of use, as well as any loss of privacy (calculated with privacy risk scores coming from empirical privacy vulnerability tools such as those discussed in Murakonda and Shokri[66] and Jayaraman and Evans[112]).

Some examples are as follows:

(1) Privacy-goal-oriented training methods: perform local training in a way that has been demonstrated as resistant to subsequent privacy attacks against the data used in training. In this approach, the training is tailored (e.g., by loss function or data pipeline specifications) to reduce memorization of the training data during training. Such efforts are described in Liu et al.[113] The PAMO mitigations of the previous section that are not accompanied by a measure of privacy afforded by the training fall into this category.

(2) Partial weight sharing: instead of sharing all the weights of the model to the aggregator during FL, only a predefined percentage (largest components) of the model is shared.[43,49]

DISCUSSION AND CONCLUSIONS

In this work, we provide a taxonomy and a deeper understanding of current privacy threats with their associated mitigation approaches, by keeping the focus on the context of FL in a health-care setting. We have provided common definitions that could be used in this field, while giving the reader detailed summaries of possible violations of privacy and their strategies to mitigate them, along with a meaningful categorization for both.

Table 2. Comparison of privacy-enhancing techniques in terms of properties that would need to be considered for deployment

Technique	SMPC	HE	CC	DP	PAMO
Exposure of data in use	no	no	no	yes	yes
Integrity of data in use	no	no	no	yes	no
Result unprotected from information extraction	yes	yes	yes	no	no
Execution integrity	depends on protocol	no	yes	no	no
Performance overhead (implementation dependent)	high	high	medium	medium	medium
Mathematical parity to the original results	yes	yes	yes	no	no
Threats mitigated	system threats	system threats	system threats, information extraction, poisoning	information extraction	information extraction

Each row is the property and the head of each column is the name of a privacy-enhancing technique.

We have begun to explore the veracity of these techniques in the context of FL for health care, following the mounting evidence that FL represents a potential paradigm shift on how multiple health-care institutions can collaborate to develop AI models without sharing any of their local data.[14,21,22,24,49,97,114] We hope that, by building upon previous works in the field,[26,29–32,34] we have provided an opportunity to current and future researchers in the field of health-care informatics to make better informed decisions during model training to appreciate potential security issues.

All the privacy threats and threat mitigation technique categories discussed in this review have been encapsulated in Table 2. The appropriate techniques to employ for a specific case differ due to the variety of protections afforded by each.

Although SMPC, HE, and CC (which protect the assets during controlled use) provide confidentiality of input data, as well as the intermediate results ("exposure of data in use"), they do not provide protection against an adversary reverse engineering this information from the final results ("results unprotected from information extraction"). In contrast, the DP and PAMO mitigation strategies (which provide free-use protection) alone do not address the confidentiality of input or intermediate results, but instead have the advantage of limiting the amount with which the final result of the computation can be used to infer information about the original inputs. Mitigations in the CC category may provide assurances as to the correctness of the computations being performed ("execution integrity"), whereas those in the other categories generally do not.

Although the costs associated with mitigation solutions can vary significantly, in general, the categories SMPC, HE, PAMO, and DP incur the cost of increased computation. In addition, SMPC can incur the cost of increased communication, and CC implementations in general have specific hardware requirements associated with enablement of hardware-supported trusted execution environments.[115] Finally, mitigations in the categories DP and PAMO generally incur the cost of a reduction in model utility (e.g., classification accuracy).

Protecting patient privacy must always be one of the primary considerations of health-care institutions, and it becomes more important as more clinical sites are initiating or joining collaborations that leverage health-care data to train AI models for further precision medicine.[14,16–20,116,117] As outlined in Kairouz et al.,[31] privacy attacks within the FL setting are a cause for concern. A significant amount of experimentation on the associated threats and mitigations, crucially in realistic scenarios on real-world data, is required to understand how they play out in different settings where health-care models are trained using FL. Understanding of the costs and benefits of these additional privacy protections during FL for health care is also critical, as there is mounting evidence[14,21,22,24,49,97,114] that FL with additional privacy protections may represent a potential paradigm shift on how multiple health-care institutions can collaborate to develop AI models without sharing any of their local data.

Application of the threat mitigation techniques outlined in this paper poses particular challenges in health care. There are no turnkey implementations, as solutions require careful configuration (and in some cases iterative tuning) to be effective. It is difficult to ensure that solutions will be accepted by the stakeholder involved, as policies on data security vary widely by institution. Even performing FL with no additional privacy threat mitigation can be difficult in the health-care domain, since most data that could be used for these purposes reside in private institutions that lack incentives to participate in FL activities, and often institutions use operational systems for data handling that make connecting the data to FL platforms a difficult task. This is in addition to the requirement that institutions export the data cohort and de-identify them prior to starting any research. These issues, specific to the health-care setting, will need to be addressed as research activities in privacy-threat-mitigating machine learning (ML) model training for medicine are undertaken. Solutions for these issues require community-driven standards to be developed in concert with relevant stakeholders.[118] Otherwise, the legal risks of loss of patient privacy due to an improperly designed solution may outweigh the benefits of integrating AI models in clinical workflows.

We are still in the early days of considering these privacy concerns and using the security technologies highlighted in this review in large-scale FL deployments. There is a significant amount of research to be done, especially into how the incorporation of select privacy mitigations into a federated study will impact the stakeholders involved. As we discussed above, the costs and benefits of individual solutions are being explored

and improved in the literature against standard measures. The results are frequently dependent on specifics such as the trained algorithm, the data distributions involved, and the compute and physical network infrastructure to be used. More studies are needed to get a better understanding of how the current results will generalize in large-scale federations. In addition, it can be difficult for prospective FL participants to make decisions based only on the measures of privacy benefit currently reported for a solution in the literature. These measures may not map well to either regulation or common patient privacy concerns. Although some work exists, more research to help bridge this gap would be valuable for those trying to balance the concerns of stakeholders to a federation.[39,45,61,67,82,104,119]

In conclusion, this review encapsulates and illustrates some of the major research directions pertaining to privacy in FL, and we hope it can be used as a primer and reference for future research studies as security becomes a growing concern in the healthcare informatics community. Although a lot of work has been done in this area, more detailed experimentation of these methods in realistic scenarios with ample, diverse, and clinically relevant use cases will be essential for their proper quantification and subsequent evaluation for clinical deployment.

ACKNOWLEDGMENTS

Research reported in this publication was partly supported by the National Cancer Institute (NCI) of the National Institutes of Health (NIH) under award nos. U01CA242871, U24CA279629, and U01CA242879. The content of this publication is solely the responsibility of the authors and does not represent the official views of the NIH.

S.P. and S.B. conducted part of the work reported in this article at their current affiliation, as well as while they were affiliated with the Center for Artificial Intelligence and Data Science for Integrated Diagnostics (AI2D), the Center for Biomedical Image Computing and Analytics (CBICA), and the Departments of Radiology and of Pathology and Laboratory Medicine, Perelman School of Medicine at the University of Pennsylvania, Philadelphia, Pennsylvania 19104, USA. Fredrik Hans Skarstedt (Indiana University) helped create the figures.

AUTHOR CONTRIBUTIONS

All authors contributed to the writing and editing of the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

1. Moore, W., and Frye, S. (2019). Review of hipaa, part 1: history, protected health information, and privacy and security rules. J. Nucl. Med. Technol. *47*, 269–272. https://doi.org/10.2967/jnmt.119.227892.

2. Mercuri, R.T. (2004). The hipaa-potamus in health care data security. Commun. ACM *47*, 25–28.

3. Choi, Y.B., Capitan, K.E., Krause, J.S., and Streeper, M.M. (2006). Challenges associated with privacy in health care industry: implementation of hipaa and the security rules. J. Med. Syst. *30*, 57–64. https://doi.org/10.1007/s10916-006-7405-0.

4. Usynin, D., Rueckert, D., Passerat-Palmbach, J., and Kaissis, G. (2022). Zen and the art of model adaptation: Low-utility-cost attack mitigations in collaborative machine learning. Proc. Priv. Enhanc. Technol. *2022*, 274–290. https://doi.org/10.2478/popets-2022-0014.

5. Annas, G.J., et al. (2003). Hipaa regulations-a new era of medical-record privacy? N. Engl. J. Med. *348*, 1486–1490. https://doi.org/10.1056/NEJMlim035027.

6. Voigt, P., and Von dem Bussche, A. (2017). The eu general data protection regulation (gdpr). In A Practical Guide, 1st Ed., *10* (Springer International Publishing), pp. 3152676. https://doi.org/10.5555/3152676.

7. Haidar, M., and Kumar, S. (2021). Smart healthcare system for biomedical and health care applications using aadhaar and blockchain. In 2021 5th International Conference on Information Systems and Computer Networks (ISCON) (IEEE), pp. 1–5. https://doi.org/10.1109/ISCON52037.2021.9702306.

8. Topol, E.J. (2019). High-performance medicine: the convergence of human and artificial intelligence. Nat. Med. *25*, 44–56. https://doi.org/10.1038/s41591-018-0300-7.

9. Dunnmon, J.A., Yi, D., Langlotz, C.P., Ré, C., Rubin, D.L., and Lungren, M.P. (2019). Assessment of convolutional neural networks for automated classification of chest radiographs. Radiology *290*, 537–544. https://doi.org/10.1148/radiol.2018181422.

10. AlBadawy, E.A., Saha, A., and Mazurowski, M.A. (2018). Deep learning for segmentation of brain tumors: Impact of cross-institutional training and testing. Med. Phys. *45*, 1150–1158. https://doi.org/10.1002/mp.12752.

11. Chang, K., Beers, A.L., Brink, L., Patel, J.B., Singh, P., Arun, N.T., Hoebel, K.V., Gaw, N., Shah, M., Pisano, E.D., et al. (2020). Multi-institutional assessment and crowdsourcing evaluation of deep learning for automated classification of breast density. J. Am. Coll. Radiol. *17*, 1653–1662. https://doi.org/10.1016/j.jacr.2020.05.015.

12. Pati, S., Thakur, S.P., Bhalerao, M., Thermos, S., Baid, U., Gotkowski, K., Gonzalez, C., Guley, O., Hamamci, I.E., Er, S., et al. (2021a). Gandlf: A generally nuanced deep learning framework for scalable end-to-end clinical workflows in medical imaging. Preprint at arXiv. https://doi.org/10.48550/arXiv.2103.01006.

13. Thakur, S.P., Schindler, M.K., Bilello, M., and Bakas, S. (2022). Clinically deployed computational assessment of multiple sclerosis lesions. Front. Med. *9*, 797586. https://doi.org/10.3389/fmed.2022.797586.

14. Pati, S., Baid, U., Edwards, B., Sheller, M., Wang, S.-H., Reina, G.A., Foley, P., Gruzdev, A., Karkada, D., Davatzikos, C., et al. (2022). Federated learning enables big data for rare cancer boundary detection. Nat. Commun. *13*, 7346. https://doi.org/10.1038/s41467-022-33407-5.

15. Pati, S., Thakur, S.P., Hamamcı, İ.E., Baid, U., Baheti, B., Bhalerao, M., Güley, O., Mouchtaris, S., Lang, D., Thermos, S., et al. (2023). Gandlf: the generally nuanced deep learning framework for scalable end-to-end clinical workflows. Commun. Eng. *2*, 23. https://doi.org/10.1038/s44172-023-00066-3.

16. GLASS Consortium (2018). Glioma through the looking glass: molecular evolution of diffuse gliomas and the glioma longitudinal analysis consortium. Neuro Oncol. *20*, 873–884. https://doi.org/10.1093/neuonc/noy020.

17. Bakas, S., Ormond, D.R., Alfaro-Munoz, K.D., Smits, M., Cooper, L.A.D., Verhaak, R., and Poisson, L.M. (2020). iglass: imaging integration into the glioma longitudinal analysis consortium. Neuro Oncol. *22*, 1545–1546. https://doi.org/10.1093/neuonc/noaa160.

18. Davatzikos, C., Barnholtz-Sloan, J.S., Bakas, S., Colen, R., Mahajan, A., Quintero, C.B., Capellades Font, J., Puig, J., Jain, R., Sloan, A.E., et al. (2020). Ai-based prognostic imaging biomarkers for precision neuro-oncology: the respond consortium. Neuro Oncol. *22*, 886–888. https://doi.org/10.1093/neuonc/noaa045.

19. Bakas, S., Reyes, M., Jakab, A., Bauer, S., Rempfler, M., Crimi, A., Shinohara, R.T., Berger, C., Ha, S.M., Rozycki, M., et al. (2018). Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge. Preprint at arXiv. https://doi.org/10.48550/arXiv.1811.02629.

20. Baid, U., Ghodasara, S., Bilello, M., Mohan, S., Calabrese, E., Colak, E., Farahani, K., Kalpathy-Cramer, J., Kitamura, F.C., Pati, S., et al. (2021). The rsna-asnr-miccai brats 2021 benchmark on brain tumor segmentation and radiogenomic classification. Preprint at arXiv. https://doi.org/10.48550/arXiv.2107.02314.

21. Chang, K., Balachandar, N., Lam, C., Yi, D., Brown, J., Beers, A., Rosen, B., Rubin, D.L., and Kalpathy-Cramer, J. (2018). Distributed deep learning networks among institutions for medical imaging. J. Am. Med. Inf. Assoc. *25*, 945–954. https://doi.org/10.1093/jamia/ocy017.

22. Sheller, M.J., Edwards, B., Reina, G.A., Martin, J., Pati, S., Kotrotsou, A., Milchenko, M., Xu, W., Marcus, D., Colen, R.R., and Bakas, S. (2020). Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. Sci. Rep. *10*, 12598–12612. https://doi.org/10.1038/s41598-020-69250-1.

23. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., et al. (2016). Communication-efficient learning of deep networks from decentralized data. Preprint at arXiv. https://doi.org/10.48550/arXiv.1602.05629.

24. Sheller, M.J., Reina, G.A., Edwards, B., Martin, J., and Bakas, S. (2018). Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In International MICCAI Brainlesion Workshop (Springer), pp. 92–104. https://doi.org/10.1007/978-3-030-11723-8_9.

25. Rieke, N., Hancox, J., Li, W., Milletarì, F., Roth, H.R., Albarqouni, S., Bakas, S., Galtier, M.N., Landman, B.A., Maier-Hein, K., et al. (2020). The future of digital health with federated learning. NPJ Digit. Med. *3*, 119–127. https://doi.org/10.1038/s41746-020-00323-1.

26. Kaissis, G.A., Makowski, M.R., Rückert, D., and Braren, R.F. (2020). Secure, privacy-preserving and federated machine learning in medical imaging. Nat. Mach. Intell. *2*, 305–311. https://doi.org/10.1038/s42256-020-0186-1.

27. Roth, H.R., Chang, K., Singh, P., Neumark, N., Li, W., Gupta, V., Gupta, S., Qu, L., Ihsani, A., Bizzo, B.C., et al. (2020). Federated learning for breast density classification: A real-world implementation. In Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning (Springer), pp. 181–191. https://doi.org/10.1007/978-3-030-60548-3_18.

28. Qu, L., Balachandar, N., Zhang, M., and Rubin, D. (2022). Handling data heterogeneity with generative replay in collaborative learning for medical imaging. Med. Image Anal. *78*, 102424. https://doi.org/10.1016/j.media.2022.102424.

29. Mothukuri, V., Parizi, R.M., Pouriyeh, S., Huang, Y., Dehghantanha, A., and Srivastava, G. (2021). A survey on security and privacy of federated learning. Future Generat. Comput. Syst. *115*, 619–640. https://doi.org/10.1016/j.future.2020.10.007.

30. Bouacida, N., and Mohapatra, P. (2021). Vulnerabilities in federated learning. IEEE Access *9*, 63229–63249. https://doi.org/10.1109/ACCESS.2021.3075203.

31. Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Nitin Bhagoji, A., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. (2021). Advances and open problems in federated learning. FNT. in Machine Learning *14*, 1–210. https://doi.org/10.1561/2200000083.

32. Li, T., Sahu, A.K., Talwalkar, A., and Smith, V. (2020). Federated learning: Challenges, methods, and future directions. IEEE Signal Process. Mag. *37*, 50–60. https://doi.org/10.1109/MSP.2020.2975749.

33. Hatamizadeh, A., Yin, H., Molchanov, P., Myronenko, A., Li, W., Dogra, P., Feng, A., Flores, M.G., Kautz, J., Xu, D., et al. (2022a). Do gradient inversion attacks make federated learning unsafe?. Preprint at arXiv. https://doi.org/10.48550/arXiv.2202.06924.

34. Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., Liu, X., and He, B. (2023). A survey on federated learning systems: vision, hype and reality for data privacy and protection. IEEE Trans. Knowl. Data Eng. *35*, 3347–3366. https://doi.org/10.1109/TKDE.2021.3124599.

35. Aouedi, O., Sacco, A., Piamrat, K., and Marchetto, G. (2023). Handling privacy-sensitive medical data with federated learning: Challenges and future directions. IEEE J. Biomed. Health Inform. *27*, 790–803. https://doi.org/10.1109/JBHI.2022.3185673.

36. Xu, J., Glicksberg, B.S., Su, C., Walker, P., Bian, J., and Wang, F. (2021). Federated learning for healthcare informatics. J. Healthc. Inform. Res. *5*, 1–19. https://doi.org/10.1007/s41666-020-00082-4.

37. Prayitno Shyu, C.-R., Shyu, C.R., Putra, K.T., Chen, H.C., Tsai, Y.Y., Hossain, K.S.M.T., Jiang, W., and Shae, Z.Y. (2021a). A systematic review of federated learning in the healthcare area: From the perspective of data properties and applications. Appl. Sci. *11*, 11191. https://doi.org/10.3390/app112311191.

38. Antunes, R.S., André da Costa, C., Küderle, A., Yari, I.A., and Eskofier, B. (2022). Federated learning for healthcare: Systematic review and architecture proposal. ACM Trans. Intell. Syst. Technol. *13*, 1–23. https://doi.org/10.1145/3501813.

39. de Castro, L., Agrawal, R., Yazicigil, R., Chandrakasan, A., Vaikuntanathan, V., Juvekar, C., and Joshi, A. (2021). Does fully homomorphic encryption need compute acceleration?. Preprint at arXiv. https://doi.org/10.48550/arXiv.2112.06396.

40. Vassilev, A., Oprea, A., Fordyce, A., and Anderson, H. (2024). Adversarial machine learning: A taxonomy and terminology of attacks and mitigations. Tech. Rep. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.AI.100-2e2023.

41. Ham, J.V.D. (2021). Toward a better understanding of "cybersecurity". Digital Threats. *2*, 1–3. https://doi.org/10.1145/3442445.

42. Prayitno Shyu, C.-R., Shyu, C.R., Putra, K.T., Chen, H.C., Tsai, Y.Y., Hossain, K.S.M.T., Jiang, W., and Shae, Z.Y. (2021b). A systematic review of federated learning in the healthcare area: From the perspective of data properties and applications. Appl. Sci. *11*, 11191. https://doi.org/10.3390/app112311191.

43. Li, W., Milletarì, F., Xu, D., Rieke, N., Hancox, J., Zhu, W., Baust, M., Cheng, Y., Ourselin, S., Cardoso, M.J., et al. (2019). Privacy-preserving federated brain tumour segmentation. In International workshop on machine learning in medical imaging (Springer), pp. 133–141. https://doi.org/10.1007/978-3-030-32692-0_16.

44. Qi, P., Chiaro, D., Guzzo, A., Ianni, M., Fortino, G., and Piccialli, F. (2024). Model aggregation techniques in federated learning: A comprehensive survey. Future Generat. Comput. Syst. *150*, 272–293. https://doi.org/10.1016/j.future.2023.09.008.

45. Zhang, G., Liu, B., Zhu, T., Zhou, A., and Zhou, W. (2022). Visual privacy attacks and defenses in deep learning: a survey. Artif. Intell. Rev. *55*, 4347–4401. https://doi.org/10.1007/s10462-021-10123-y.

46. Smestad, C., and Li, J. (2023). A systematic literature review on client selection in federated learning. In Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering, pp. 2–11. https://doi.org/10.1145/3593434.3593438.

47. Huang, J., Hong, C., Liu, Y., Chen, L.Y., and Roos, S. (2023). Maverick matters: Client contribution and selection in federated learning. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (Springer), pp. 269–282. https://doi.org/10.1007/978-3-031-33377-4_21.

48. Kaissis, G., Ziller, A., Passerat-Palmbach, J., Ryffel, T., Usynin, D., Trask, A., Lima, I., Mancuso, J., Jungmann, F., Steinborn, M.-M., et al. (2021). End-to-end privacy preserving deep learning on multi-institutional medical imaging. Nat. Mach. Intell. *3*, 473–484. https://doi.org/10.1038/s42256-021-00337-8.

49. Dayan, I., Roth, H.R., Zhong, A., Harouni, A., Gentili, A., Abidin, A.Z., Liu, A., Costa, A.B., Wood, B.J., Tsai, C.-S., et al. (2021). Federated learning for predicting clinical outcomes in patients with covid-19. Nat. Med. *27*, 1735–1743. https://doi.org/10.1038/s41591-021-01506-3.

50. Melis, L., Song, C., De Cristofaro, E., and Shmatikov, V. (2019). Exploiting unintended feature leakage in collaborative learning. Preprint at arXiv. https://doi.org/10.1109/SP.2019.00029.

51. Nasr, M., Shokri, R., and Houmansadr, A. (2019). Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In 2019 IEEE symposium on security and privacy (SP) (IEEE), pp. 739–753. https://doi.org/10.1109/SP.2019.00065.

52. Carlini, N., Liu, C., Kos, J., Erlingsson, Ú., and Song, D. (2018). The secret sharer: Measuring unintended neural network memorization & extracting secrets. Preprint at arXiv. https://doi.org/10.48550/arXiv.1802.08232.

53. Thakkar, O.D., Ramaswamy, S., Mathews, R., and Beaufays, F. (2021). Understanding unintended memorization in language models under federated learning. In Proceedings of the Third Workshop on Privacy in

Natural Language Processing, pp. 1–10. https://doi.org/10.18653/v1/2021.privatenlp-1.1.

54. Song, C., Ristenpart, T., and Shmatikov, V. (2017). Machine learning models that remember too much. In Proceedings of the 2017 ACM SIGSAC Conference on computer and communications security, pp. 587–601. https://doi.org/10.1145/3133956.3134077.

55. Fredrikson, M., Lantz, E., Jha, S., Lin, S., Page, D., and Ristenpart, T. (2014). Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In Proceedings of the 23rd USENIX Conference on Security Symposium. SEC'14 USA (USENIX Association), pp. 17–32. https://doi.org/10.5555/2671225.2671227.

56. Fredrikson, M., Jha, S., and Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. CCS '15 (Association for Computing Machinery), pp. 1322–1333. URL:. https://doi.org/10.1145/2810103.2813677

57. Li, Z., Wang, L., Chen, G., Zhang, Z., Shafiq, M., and Gu, Z. (2023). E2egi: End-to-end gradient inversion in federated learning. IEEE J. Biomed. Health Inform. *27*, 756–767. https://doi.org/10.1109/JBHI.2022.3204455.

58. Zhang, Y., Jia, R., Pei, H., Wang, W., Li, B., and Song, D. (2020). The secret revealer: Generative model-inversion attacks against deep neural networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 253–261.

59. Huang, Y., Gupta, S., Song, Z., Li, K., and Arora, S. (2021). Evaluating gradient inversion attacks and defenses in federated learning. Adv. Neural Inf. Process. Syst. *34*, 7232–7241.

60. Zhu, L., Liu, Z., and Han, S. (2019). Deep leakage from gradients. Adv. Neural Inf. Process. Syst. *32*.

61. Song, J., and Namiot, D. (2022). A survey of the implementations of model inversion attacks. In International Conference on Distributed Computer and Communication Networks (Springer), pp. 3–16. https://doi.org/10.1007/978-3-031-30648-8_1.

62. Hatamizadeh, A., Yin, H., Molchanov, P., Myronenko, A., Li, W., Dogra, P., Feng, A., et al. (2022b). Do gradient inversion attacks make federated learning unsafe?. Preprint at arXiv. https://doi.org/10.48550/arXiv.2202.06924.

63. Liu, Y., Wen, R., He, X., Salem, A., Zhang, Z., Backes, M., De Cristofaro, E., Fritz, M., and Zhang, Y. (2022a). {ML-Doctor}: Holistic risk assessment of inference attacks against machine learning models. In 31st USENIX Security Symposium (USENIX Security 22), pp. 4525–4542.

64. Samala, R.K., Chan, H.-P., Hadjiiski, L., and Koneru, S. (2020). Hazards of data leakage in machine learning: a study on classification of breast cancer using deep neural networks. Medical Imaging 2020: Computer-Aided Diagnosis *11314*, 279–284. https://doi.org/10.1117/12.2549313.

65. Kaufman, S., Rosset, S., Perlich, C., and Stitelman, O. (2012). Leakage in data mining: Formulation, detection, and avoidance. ACM Trans. Knowl. Discov. Data *6*, 1–21. https://doi.org/10.1145/2382577.2382579.

66. Murakonda, S.K., and Shokri, R. (2020). Ml privacy meter: Aiding regulatory compliance by quantifying the privacy risks of machine learning. Preprint at arXiv. missingarXiv:2007.09339. https://doi.org/10.48550/arXiv.2007.09339.

67. Hu, H., Salcic, Z., Sun, L., Dobbie, G., Yu, P.S., and Zhang, X. (2022). Membership inference attacks on machine learning: A survey. ACM Comput. Surv. *54*, 1–37. https://doi.org/10.1145/3523273.

68. Ateniese, G., Mancini, L.V., Spognardi, A., Villani, A., Vitali, D., and Felici, G. (2015). Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. Int. J. Secur. Network. *10*, 137–150. https://doi.org/10.1504/IJSN.2015.071829.

69. Sanyal, S., Addepalli, S., and Babu, R.V. (2022). Towards data-free model stealing in a hard label setting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15284–15293. https://doi.org/10.1109/CVPR52688.2022.01485.

70. Orekondy, T., Schiele, B., and Fritz, M. (2019). Knockoff nets: Stealing functionality of black-box models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 4954–4963. https://doi.org/10.1109/CVPR.2019.00509.

71. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2018). A survey of methods for explaining black box models. ACM Comput. Surv. *51*, 1–42. https://doi.org/10.1145/3236009.

72. Hitaj, B., Ateniese, G., and Perez-Cruz, F. (2017). Deep Models under the gan: Information Leakage from Collaborative Deep Learning. https://doi.org/10.1145/3133956.3134012.

73. Yao, A.C. (1982). Protocols for secure computations. In 23rd annual symposium on foundations of computer science (sfcs 1982) (IEEE), pp. 160–164. https://doi.org/10.1109/SFCS.1982.38.

74. Goldreich, O. (1998). Secure multi-party computation. Manuscript. Preliminary version *78*, 110.

75. Shamir, A., Rivest, R.L., and Adleman, L.M. (1981). Mental poker. In The mathematical gardner (37–43) (Springer), pp. 37–43. https://doi.org/10.1007/978-1-4684-6686-7_5.

76. Sabt, M., Achemlal, M., and Bouabdallah, A. (2015). Trusted execution environment: what it is, and what it is not. In 2015 IEEE Trustcom/BigDataSE/ISPA, *1* (IEEE), pp. 57–64. https://doi.org/10.1109/Trustcom.2015.357.

77. Schneider, M., Masti, R.J., Shinde, S., Capkun, S., and Perez, R. (2022). Sok: Hardware-supported trusted execution environments. Preprint at arXiv. https://doi.org/10.48550/arXiv.2205.12742.

78. Consortium, C., et al. (2021). Confidential computing: Hardware-based trusted execution for applications and data. A Publication of The Confidential Computing Consortium.

79. Frikken, K.B. (2010). Secure multiparty computation. Algorithms and theory of computation handbook: special topics and techniques (ACM), p. 14. https://doi.org/10.5555/1882723.1882737.

80. Kalapaaking, A.P., Stephanie, V., Khalil, I., Atiquzzaman, M., Yi, X., and Almashor, M. (2022). Smpc-based federated learning for 6g-enabled internet of medical things. IEEE Network *36*, 182–189. https://doi.org/10.1109/MNET.007.2100717.

81. Kalapaaking, A.P., Khalil, I., and Yi, X. (2023). Blockchain-based federated learning with smpc model verification against poisoning attack for healthcare systems. Preprint at arXiv. https://doi.org/10.48550/arXiv.2304.13360.

82. Buyukates, B., So, J., Mahdavifar, H., and Avestimehr, S. (2022). Lightverifl: Lightweight and verifiable secure federated learning. In Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022), pp. 1–20. URL: https://openreview.net/pdf?id=WA7I-Fm4tmP

83. Huang, C., Yao, Y., Zhang, X., Teng, D., Wang, Y., and Zhou, L. (2022). Robust secure aggregation with lightweight verification for federated learning. In 2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) (IEEE), pp. 582–589. https://doi.org/10.1109/TrustCom56396.2022.00085.

84. Gentry, C., and Halevi, S. (2011). Implementing gentry's fully-homomorphic encryption scheme. In Annual international conference on the theory and applications of cryptographic techniques (Springer), pp. 129–148. https://doi.org/10.1007/978-3-642-20465-4_9.

85. Ahmed, E.-Y., and ELKETTANI, M.D. (2016). Fully homomorphic encryption: state of art and comparison. Int. J. Comput. Sci. Inf. Secur. *14*. https://doi.org/10.6084/M9.FIGSHARE.3362338.

86. Acar, A., Aksu, H., Uluagac, A.S., and Conti, M. (2018). A survey on homomorphic encryption schemes: Theory and implementation. ACM Comput. Surv. *51*, 1–35. https://doi.org/10.1145/3214303.

87. Stripelis, D., Saleem, H., Ghai, T., Dhinagar, N.J., Gupta, U., Anastasiou, C., Ver Steeg, G., Ravi, S., Naveed, M., Thompson, P.M., and Ambite, J.L. (2021). Secure neuroimaging analysis using federated learning with homomorphic encryption. Preprint at arXiv. https://doi.org/10.1117/12.2606256.

88. Ma, J., Naas, S.-A., Sigg, S., and Lyu, X. (2021). Privacy-preserving federated learning based on multi-key homomorphic encryption. Preprint at arXiv. https://doi.org/10.1002/int.22818.

89. Doröz, Y., Öztürk, E., and Sunar, B. (2014). Accelerating fully homomorphic encryption in hardware. IEEE Trans. Comput. *64*, 1–1521. https://doi.org/10.1109/TC.2014.2345388.

90. Cao, X., Moore, C., O'Neill, M., O'Sullivan, E., and Hanley, N. (2013). Accelerating fully homomorphic encryption over the integers with super-size hardware multiplier and modular reduction. Cryptology ePrint Archive. URL: https://eprint.iacr.org/2013/616

91. Froelicher, D., Troncoso-Pastoriza, J.R., Raisaro, J.L., Cuendet, M.A., Sousa, J.S., Cho, H., Berger, B., Fellay, J., and Hubaux, J.-P. (2021). Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption. Nat. Commun. *12*, 5910. https://doi.org/10.1038/s41467-021-25972-y.

92. Chen, Y., Qin, X., Wang, J., Yu, C., and Gao, W. (2020). Fedhealth: A federated transfer learning framework for wearable healthcare. IEEE Intell. Syst. *35*, 83–93. https://doi.org/10.1109/MIS.2020.2988604.

93. Dwork, C., and Feldman, V. (2018). Privacy-preserving prediction. In Conference On Learning Theory (PMLR), pp. 1693–1702.

94. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., and Zhang, L. (2016). Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, pp. 308–318. https://doi.org/10.1145/2976749.2978318.

95. Zhao, J., Chen, Y., and Zhang, W. (2019). Differential privacy preservation in deep learning: Challenges, opportunities and solutions. IEEE Access *7*, 48901–48911. https://doi.org/10.1109/ACCESS.2019.2909559.

96. Adnan, M., Kalra, S., Cresswell, J.C., Taylor, G.W., and Tizhoosh, H.R. (2022a). Federated learning and differential privacy for medical image analysis. Sci. Rep. *12*, 1953. https://doi.org/10.1038/s41598-022-05539-7.

97. Sadilek, A., Liu, L., Nguyen, D., Kamruzzaman, M., Serghiou, S., Rader, B., Ingerman, A., Mellem, S., Kairouz, P., Nsoesie, E.O., et al. (2021). Privacy-first health research with federated learning. NPJ Digit. Med. *4*, 132–138. https://doi.org/10.1038/s41746-021-00489-2.

98. Liu, H., Peng, C., Tian, Y., Long, S., Tian, F., and Wu, Z. (2022b). Gdp vs. ldp: A survey from the perspective of information-theoretic channel. Entropy *24*, 430. https://doi.org/10.3390/e24030430.

99. Pati, S., Baid, U., Zenk, M., Edwards, B., Sheller, M., Reina, G.A., Foley, P., Gruzdev, A., Martin, J., Albarqouni, S., et al. (2021b). The federated tumor segmentation (fets) challenge. Preprint at arXiv. https://doi.org/10.48550/arXiv.2105.05874.

100. Lee, J., and Kifer, D. (2021). Scaling up differentially private deep learning with fast per-example gradient clipping. In Proceedings on Privacy Enhancing Technologies. https://doi.org/10.2478/popets-2021-0008.

101. Shen, Z., and Zhong, T. (2021). Analysis of application examples of differential privacy in deep learning. Comput. Intell. Neurosci. *2021*, 4244040–4244115. https://doi.org/10.1155/2021/4244040.

102. Ficek, J., Wang, W., Chen, H., Dagne, G., and Daley, E. (2021). Differential privacy in health research: A scoping review. J. Am. Med. Inf. Assoc. *28*, 2269–2276. https://doi.org/10.1093/jamia/ocab135.

103. Jarin, I., and Eshete, B. (2022). Dp-util: comprehensive utility analysis of differential privacy in machine learning. In Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy, pp. 41–52. https://doi.org/10.1145/3508398.3511513.

104. Demelius, L., Kern, R., and Trügler, A. (2023). Recent advances of differential privacy in centralized deep learning: A systematic survey. Preprint at arXiv. https://doi.org/10.48550/arXiv.2309.16398.

105. Adnan, M., Kalra, S., Cresswell, J.C., Taylor, G.W., and Tizhoosh, H.R. (2022b). Federated learning and differential privacy for medical image analysis. Sci. Rep. *12*, 1953.

106. Malekzadeh, M., Hasircioglu, B., Mital, N., Katarya, K., Ozfatura, M.E., and Gunduz, D. (2021). Dopamine: Differentially private federated learning on medical data. Preprint at arXiv. https://doi.org/10.48550/arXiv.2101.11693.

107. Ziller, A., Usynin, D., Remerscheid, N., Knolle, M., Makowski, M., Braren, R., Rueckert, D., and Kaissis, G. (2021). Differentially private federated deep learning for multi-site medical image segmentation. Preprint at arXiv. https://doi.org/10.48550/arXiv.2107.02586.

108. Pfohl, S.R., Dai, A.M., and Heller, K. (2019). Federated and differentially private learning for electronic health records. Preprint at arXiv. https://doi.org/10.48550/arXiv.1911.05861.

109. Arasteh, S.T., Ziller, A., Kuhl, C., Makowski, M., Nebelung, S., Braren, R., Rueckert, D., Truhn, D., and Kaissis, G. (2023). Private, fair and accurate: Training large-scale, privacy-preserving ai models in medical imaging. Preprint at arXiv. https://doi.org/10.48550/arXiv.2302.01622.

110. Nasr, M., Shokri, R., and Houmansadr, A. (2018). Machine learning with membership privacy using adversarial regularization. In Proceedings of the 2018 ACM SIGSAC conference on computer and communications security, pp. 634–646. https://doi.org/10.1145/3243734.3243855.

111. Makhdoumi, A., Salamatian, S., Fawaz, N., and Médard, M. (2014). From the information bottleneck to the privacy funnel. In IEEE Information Theory Workshop (ITW 2014). IEEE, pp. 501–505. https://doi.org/10.1109/ITW.2014.6970882.

112. Jayaraman, B., and Evans, D. (2019). Evaluating differentially private machine learning in practice. In 28th USENIX Security Symposium (USENIX Security 19) (USENIX Association), pp. 1895–1912. https://doi.org/10.5555/3361338.3361469.

113. Liu, J., Oya, S., and Kerschbaum, F. (2021). Generalization techniques empirically outperform differential privacy against membership inference. Preprint at arXiv. https://doi.org/10.48550/arXiv.2110.05524.

114. Pham, Q.-V., Zeng, M., Ruby, R., Huynh-The, T., and Hwang, W.-J. (2021). Uav communications for sustainable federated learning. IEEE Trans. Veh. Technol. *70*, 3944–3948. https://doi.org/10.1109/TVT.2021.3065084.

115. Ekberg, J.-E., Kostiainen, K., and Asokan, N. (2014). The untapped potential of trusted execution environments on mobile devices. IEEE Secur. Priv. *12*, 29–37. https://doi.org/10.1109/MSP.2014.38.

116. Armato, S.G., 3rd, McLennan, G., McNitt-Gray, M.F., Meyer, C.R., Yankelevitz, D., Aberle, D.R., Henschke, C.I., Hoffman, E.A., Kazerooni, E.A., MacMahon, H., et al. (2004). Lung image database consortium: developing a resource for the medical imaging research community. Radiology *232*, 739–748. https://doi.org/10.1148/radiol.2323032035.

117. Thompson, P.M., Stein, J.L., Medland, S.E., Hibar, D.P., Vasquez, A.A., Renteria, M.E., Toro, R., Jahanshad, N., Schumann, G., Franke, B., et al. (2014). The enigma consortium: large-scale collaborative analyses of neuroimaging and genetic data. Brain Imaging Behav. *8*, 153–182. https://doi.org/10.1007/s11682-013-9269-5.

118. Karargyris, A., Umeton, R., Sheller, M.J., Aristizabal, A., George, J., Bala, S., Beutel, D.J., Bittorf, V., Chaudhari, A., Chowdhury, A., et al. (2021). Medperf: Open benchmarking platform for medical artificial intelligence using federated evaluation. Preprint at arXiv. https://doi.org/10.48550/arXiv.2110.01406.

119. Tonni, S.M., Vatsalan, D., Farokhi, F., Kaafar, D., Lu, Z., and Tangari, G. (2020). Data and model dependencies of membership inference attack. Preprint at arXiv. https://doi.org/10.48550/arXiv.2002.06856.

electronics

MDPI

Article

Secure and Flexible Privacy-Preserving Federated Learning Based on Multi-Key Fully Homomorphic Encryption

Jiachen Shen, Yekang Zhao, Shitao Huang and Yongjun Ren *

School of Computer Science, School of Cyber Science and Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China
* Correspondence: 002315@nuist.edu.cn

Abstract: Federated learning avoids centralizing data in a central server by distributing the model training process across devices, thus protecting privacy to some extent. However, existing research shows that model updates (e.g., gradients or weights) exchanged during federated learning may still indirectly leak sensitive information about the original data. Currently, single-key homomorphic encryption methods applied in federated learning cannot solve the problem of privacy leakage that may be caused by the collusion between the participant and the federated learning server, whereas existing privacy-preserving federated learning schemes based on multi-key homomorphic encryption in semi-honest environments have deficiencies and limitations in terms of security and application conditions. To this end, this paper proposes a privacy-preserving federated learning scheme based on multi-key fully homomorphic encryption to cope with the potential risk of privacy leakage in traditional federated learning. We designed a multi-key fully homomorphic encryption scheme, mMFHE, that encrypts by aggregating public keys and requires all participants to jointly participate in decryption sharing, thus ensuring data security and privacy. The proposed privacy-preserving federated learning scheme encrypts the model updates through multi-key fully homomorphic encryption, ensuring confidentiality under the CRS model and in a semi-honest environment. As a fully homomorphic encryption scheme, mMFHE supports homomorphic addition and homomorphic multiplication for more flexible applications. Our security analysis proves that the scheme can withstand collusive attacks by up to $N-1$ users and servers, where N is the total number of users. Performance analysis and experimental results show that our scheme reduces the complexity of the NAND gate, which reduces the computational load and improves the efficiency while ensuring the accuracy of the model.

Keywords: privacy preservation; federated learning; multi-key fully homomorphic encryption

Citation: Shen, J.; Zhao, Y.; Huang, S.; Ren, Y. Secure and Flexible Privacy-Preserving Federated Learning Based on Multi-Key Fully Homomorphic Encryption. *Electronics* **2024**, *13*, 4478. https://doi.org/10.3390/electronics13224478

Academic Editor: Subir Halder

Received: 9 October 2024
Revised: 29 October 2024
Accepted: 13 November 2024
Published: 14 November 2024

1. Introduction

The training of machine learning models usually requires the centralization of dispersed data to a single server or data center, a practice that not only presents technical challenges, such as communication costs and latency for large data transfers, but more importantly also raises significant concerns about privacy and data protection [1,2]. With the popularity of smartphones and various smart devices, a significant volume of individual data have been accumulated on the devices, which will be of great value if we can make use of them to enhance the quality of service and user experience under the premise of protecting privacy [3]. Google's research team developed the federated learning (FL) concept in response to these challenges [4,5]. The fundamental premise of FL is that every device (or node) trains a model locally with the data it holds, and only sends updates to the model (e.g., gradient or parameter updates) to the server. The server is tasked with aggregating the aforementioned updates, updating the global model, and subsequently disseminating the enhanced model to the participating devices. In this way, the data are not transmitted from the local device, which serves to safeguard the user's privacy. Since

its introduction in 2016, FL has evolved from a preliminary concept to an active research field and has found practical applications in healthcare [6,7], meta-universe [8,9], IoT [10], and many other areas.

However, it has been shown that in FL, model updates (e.g., gradients or weights) exchanged during model training may indirectly disclose confidential information about the original data, even though the data are retained locally [11,12]. For this reason, privacy-preserving federated learning (PPFL), as an extension of FL, further reduces potential privacy risks by introducing techniques such as differential privacy (DP), secure multi-party computation (SMPC), and homomorphic encryption (HE). DP protects the privacy of personal data by adding noise to perturb the training data or model parameters to ensure that modifications to individual data samples do not significantly affect the output results [13]. Nevertheless, the incorporation of noise inevitably impedes the rate of convergence of the model and diminishes the accuracy of the model within the context of aggregation [14]. SMPC, as a cryptographic method, permits the participation of multiple parties in the computation of a function while maintaining the confidentiality of their respective input data; however, SMPC usually requires complex cryptographic operations and frequent network communication, which may lead to higher computational costs and delays, especially in large-scale systems.

The core advantage of HE is its ability to perform computation while the data remain encrypted, which means that the data can be securely processed without decryption. And compared with DP and SMPC, HE can have a small communication overhead while maintaining model accuracy, but the traditional single-key HE in which all participants share the same encryption and decryption keys presents a significant risk to the privacy and security of the other participants in the event that a malicious user conspires with the server. On the other hand, extant privacy-preserving FL (PPFL) schemes based on multi-key homomorphic encryption (MKHE) in semi-honest environments are constrained in terms of security and application conditions. Specifically, in extreme cases, at least two participants (i.e., $k < N - 1$) are required not to collude with the attacker, where N is the total number of users and k is the number of users who collude with the server. The aforementioned limitations of existing PPFL schemes based on MKHE in semi-honest environments can be attributed to two primary factors. Firstly, the attacker's ability to prevent the leakage of private data is insufficient in terms of security. Secondly, they only support homomorphic addition but not homomorphic multiplication (or they require trustworthy hardware support to perform multiplication), which imposes many restrictions on the FL algorithms used in applications and is not flexible enough.

To address this, we propose a multi-key fully homomorphic encryption (MKFHE) scheme derived from GSW13 [15], named mMFHE, to enhance privacy protection in FL. As an FHE scheme, mMFHE is capable of supporting both the homomorphic addition and multiplication of encrypted data and ensures security by requiring encryption with an aggregated public key and decryption sharing. Additionally, mMFHE allows for embedding multiple plaintext messages within a single ciphertext, thereby improving the efficacy of the encryption and decryption procedures. In this paper, we present a PPFL scheme based on mMFHE. In this scheme, model updates are encrypted using homomorphic encryption before being shared for aggregation. Furthermore, the aggregated result can only be decrypted through the collaboration of all those who contributed to its generation. This renders the scheme resilient to assaults from participants and collusion attacks conducted by participants in conjunction with the server. Finally, we performed experiments based on simulations using actual data to assess the efficacy of mMFHE and the proposed PPFL scheme. Specifically, our contributions are the following:

1. We propose a multi-key FHE scheme, mMFHE, that ensures security by requiring encryption with an aggregated public key and decryption sharing. It supports the encryption, decryption, and homomorphic evaluation of multi-bit messages in a single operation, reducing the complexity of NAND gates and enhancing efficiency.

Additionally, it allows for homomorphic addition and multiplication on ciphertexts, increasing application flexibility.

2. We introduce a PPFL scheme based on mMFHE. This scheme employs multi-key FHE to encrypt model updates, thereby ensuring their confidentiality within a semi-honest environment. The proposed scheme exhibits robustness against attacks by participants and collusion attacks, even when involving up to $k = N - 1$ participants in conjunction with the server.
3. We analyze the security of the PPFL scheme based on mMFHE and perform a comparative analysis to evaluate its efficiency. The results of simulation-based experiments conducted on actual datasets demonstrate that our proposed PPFL scheme can reduce computational load while maintaining accuracy, thereby ensuring efficiency.

We introduce the research status of PPFL and the existing PPFL schemes based on HE in Section 2. Section 3 introduces some prerequisite knowledge related to the mMFHE scheme we proposed. Section 4 describes in detail the mMFHE and PPFL schemes based on mMFHE proposed by us. Section 5 presents a detailed security analysis of the PPFL scheme, and the performance analysis and simulation experiments are discussed in Section 6. We discuss some problems currently faced by PPFL based on mMFHE in Section 7 and give a summary of our work and future prospects in Section 8.

2. Related Work

2.1. PPFL

FL is a decentralized machine learning paradigm, first pioneered by McMahan et al. [4] in 2016, that allows several participants to collaboratively train a joint machine learning model without having to exchange or centrally store their individual datasets. The challenge of data silos, which is common in centralized machine learning, is effectively addressed by this approach. Since then, a number of efforts have emerged to improve FL in a variety of ways, from the design of the underlying algorithms to performance optimization and security enhancements. In the same year, the FedAvg algorithm was proposed [5], which allows each participant to train a model independently using their local data and then send their respective model parameters to a central server for averaging. This process is iterative until the model is in a state of convergence. FedAvg is the original and most basic FL algorithm and is widely used in a variety of application scenarios. In 2018, Li et al. [16] suggested the FedProx algorithm, in which a regular term is added to FedAvg in order to deal with the heterogeneity of the devices in the system. In 2020, Asad et al. [17] presented the FedOpt algorithm, which introduces more advanced optimization strategies such as adaptive gradient methods to optimize the updating and aggregation process of the global model. In 2023, Zhang et al. [18] proposed an Adaptive Locally Aggregated FL (FedALA) approach for personalizing client models in FL by capturing the information required in the global model. In 2024, A Raft consensus protocol based on Cauchy Reed–Solomon (CRS) codes was proposed for adaptive data maintenance in the metaverse of Yu et al. [19]. The protocol reduces the data storage requirements of nodes utilizing code erasure technology.

In FL, model updates (e.g., gradients or weights) exchanged during model training may indirectly disclose confidential information about the original data, despite the data being retained locally [11,12]. For this reason, privacy-preserving federated learning, as an extension of federated learning, further reduces potential privacy risks by introducing encryption and anonymization techniques. Most of the existing work related to privacy-preserving federated learning employs techniques such as DP [20–22], SMPC [23], and HE [24,25].

The DP process ensures the confidentiality of personal data by introducing noise to disrupt either the training data or model parameters, thereby ensuring that alterations to specific data samples do not significantly influence the output. In 2020, Wei et al. [26] put forth a new framework leveraging the principles of DP, which effectively mitigates the threat of information leakage by embedding artificial noise in the client-side parameters prior to aggregation. Concurrently, Truex et al. [27] integrated local differential privacy

(LDP) into federated learning, providing formal privacy guarantees and developing an innovative federated learning system, LDP-Fed. In 2023, He et al. advanced this field by introducing a local differential privacy scheme tailored to address the challenges of fine-grained range differences in weights across different layers of FL models, as well as the issue of privacy budget accumulation leading to budget explosion. Their method, ACS-FL, used adaptive cropping, weight compression, and the reorganization of parameters to train clustered FL models on disparate IoT data. However, the noise introduced by DP inherently impairs the speed of aggregation and diminishes the accuracy of the model during aggregation [28].

Compared to DP, SMPC allows several participants to collaboratively calculate a function without exposing their respective inputs through cryptographic methods and has a lower impact on model accuracy. Bonawitz et al. [25] were the first to introduce SMPC into federated learning for secure aggregation, constructing a confidentiality calculation process in combination with dual masking. In 2020, Li et al. [29] combined a single masking mechanism and a chained communication mechanism and proposed a new PPFL framework on the basis of chained SMPC, which improved the aggregation efficiency to some extent. And for the combined problem of privacy inference attacks and poison attacks that federated learning systems may suffer from, Gehlhar et al. [30] proposed an SMPC-based framework, SafeFL, aiming to evaluate the effectiveness of the FL technique in solving privacy inference and poison attacks. However, SMPC usually requires complex cryptographic operations and frequent network communication, which may lead to high computational cost and latency, especially in large-scale systems.

2.2. HE-Based PPFL

The core strength of HE is its ability to perform computation while the data remain encrypted, which means that the data can be securely handled without being decrypted. Zhang et al. [31] proposed a Privacy-Enhanced FL (PEFL) scheme that protects gradients on untrusted servers by encrypting the local gradients of the participant using the Paillier homomorphic cryptosystem. Li et al. [32] constructed an FL security framework on the basis of the threshold Paillier cryptosystem for use in IoT environments [33] and thereby mitigated the negative impact of untrusted users. He et al. [34] presented a privacy-protecting and low-latency FL scheme tailored for edge computing [35], which ensures the privacy of terminal devices by transferring parameters encrypted with an enhanced version of the Paillier homomorphic encryption algorithm, thereby avoiding the transmission of raw data to the edge node. Despite these advancements, this approach relies on single-key homomorphic encryption algorithms, where all devices utilize the same encryption and decryption keys. This architecture poses a significant risk to privacy and security, particularly in scenarios where a malicious participant conspires with the server, potentially compromising the confidentiality of other participants. To better implement the privacy preservation requirements in FL systems in multi-user environments, several works have introduced MKHE. Cai et al. [36] designed a TEE-based MKHE cryptosystem (EMK-BFV) to facilitate PPFL and optimize the operational performance. Ma et al. [37] designed an innovative PPFL scheme based on the multi-key CKKS algorithm learning scheme xMK-CKKS, while Walskaar et al. [38] proposed a more efficient federated learning scheme based on xMK-CKKS by incorporating the Flower framework. Zhang et al. [39] proposed a PPFL scheme VPFL based on the BCP cryptosystem, which is capable of verifying the user's identity and data integrity in a multi-key environment. However, all of the above works are deficient in terms of security; the schemes in reference [36–38] must have at least two participants who do not conspire with the server ($k < N - 1$) to preserve that the honest party's privacy is not compromised in the face of a user–server conspiracy attack in semi-honest environments, whereas [39] presents a two-server scheme that requires the two servers to not be complicit in order to ensure security. In addition, [37–39] only support homomorphic addition, while [36] supports multiplication, but it is essentially a direct-to-plaintext multiplication in a trusted execution environment, rather than homomorphic

multiplication, which requires trusted hardware support; therefore, the above schemes will impose many limitations on the federated learning algorithms used in terms of their applications, and they are not flexible enough.

Table 1 summarizes the above HE-based PPFL scheme and compares it with our proposed scheme, mainly including whether it supports homomorphic multiplication and the security in the face of collusion attacks between users and servers, where k is the number of users who collude with the server, and N is the total number of users participating in FL.

Table 1. Comparison of HE-based PPFL.

Scheme	Base	Homomorphic Addition	Homomorphic Multiplication	Security Against Collusion Attacks
[31,32,34]	Paillier	Yes	No	No
[36]	BFV	Yes	Support for plaintext multiplication in TEE.	$k < N-1$
[37]	MK-CKKS	Yes	No	$k < N-1$
[38]	MK-CKKS	Yes	No	$k < N-1$
[39]	BCP	Yes	No	Requires that the two servers cannot collude
Ours	mMFHE	Yes	Yes	$k = N-1$

3. Preliminaries

3.1. Definitions

For $n \in N$, we denote the set $\{1, \cdots, n\}$ by $[n]$. For any real number $x \in \mathbb{R}$, we define $\lfloor x \rfloor$ as the greatest integer less than or equal to x, and $\lfloor x \rceil := \lfloor x + \frac{1}{2} \rfloor$ as the integer closest to x. Matrices are represented by bold, uppercase letters: $\mathbf{A}$. We use ":=" for deterministic assignments.

Definition 1. *For a distribution $\{\chi_n\}_{n\in N}$ based on integers, if it satisfies $\mathsf{Pr}[|x| \geq B] = negl(\lambda)$, where $negl(\lambda)$ is a negligible function, then we call the distribution B-bounded.*

Theorem 1. *For a range of random variables $x_i (i \in N)$, if it obeys a $B-$bounded distribution, then the random variable $x = \frac{1}{N}\sum_{i=1}^{N} x_i$ also obeys the $B-$bounded distribution.*

Definition 2. *The statistical distance between two distributions A and B over a finite field Ω is $\Delta(X,Y) \overset{\text{stat}}{=} \frac{1}{2}\sum_{t\in\Omega} \mid A(t) - B(t) \mid$. A negligible $\Delta(A,B)$ implies $A \overset{\text{state}}{\approx} B$.*

Definition 3 (Learning with Errors, LWE)**.** *We consider the case of a secret vector $\boldsymbol{s}$ belonging to the discrete cube Z_q^n. The LWE distribution $Z_q^n \times Z_q$ over A_s, χ is established through the following definition: uniformly sample $\boldsymbol{a} \in Z_q^{n\times m}$ and $\boldsymbol{e} \leftarrow \chi$, and then output the pair $(\boldsymbol{a}, \boldsymbol{b} = \boldsymbol{a}\cdot\boldsymbol{s} + \boldsymbol{e}) mod\ q$.*

Definition 4 (*search.LWE$_{n,q,\chi,m}$*)**.** *For the secret vector $\boldsymbol{s} \in Z_q^n$ and given m independent samples $(\boldsymbol{a_i}, \boldsymbol{b_i}) \in Z_q^n \times Z_q$ to recover $\boldsymbol{s}$, these are selected from the distributions A_s, χ.*

Definition 5 (*Decision.LWE$_{n,q,\chi,m}$*)**.** *Given m independent samples $(\boldsymbol{a_i}, \boldsymbol{b_i}) \in Z_q^n \times Z_q$, these samples are selected from the following two distributions: (1) from A_s, χ; (2) drawn uniformly from $Z_q^n \times Z_q$. The advantage of being able to distinguish between these two types of selection is negligible.*

Definition 6 (Some-are-errorless LWE)**.** *Let $q \geq 1$, $n > 0$, and χ' be a distribution of errors over R. Define $T_q = \{0, \frac{1}{q}, \cdots, \frac{q-1}{q}\}$, where $q \in Z$. The distribution $A'_{s,\chi}$ over $T_q^n \times T_q$ is defined by*

uniformly selecting $\boldsymbol{a} \in T_q^n$ *and* $\boldsymbol{e} \leftarrow \chi'$*, and outputting* $(\boldsymbol{a}, \boldsymbol{b} = \boldsymbol{a} \cdot \boldsymbol{s} + \boldsymbol{e})$*. The some-are-errorless LWE problem concerns the distinction between two scenarios:*

(1) All samples are uniformly selected from $T_q^n \times T_q$*.*

(2) A random secret vector $\boldsymbol{s} \in T_q^n$ *is uniformly chosen, with the first* l *samples drawn from* $A'_{s,0}$ *and the remaining samples drawn from* $A'_{s,\chi}$*. In other words, the first* l *samples are of the form* $(\boldsymbol{a}, \boldsymbol{b} = \boldsymbol{a} \cdot \boldsymbol{s})$*, which are errorless, while the remaining samples* $(\boldsymbol{a_i}, \boldsymbol{b_i} = \boldsymbol{a_i} \cdot \boldsymbol{s} + \boldsymbol{e_i})$ *for each* $i > l$ *introduce a small error* $\boldsymbol{e}_i$*.*

Theorem 2. *For any* $n, l, q \geq 1 (l \ll n)$*, and an error distribution* χ'*, there exists a problem ranging from* $LWE_{n-1,q,\chi}$ *to* $LWE_{n,q,\chi}$*, a variant some-are-errorless problem. LWE polynomials reduce the success advantage of the problem by up to* $p - n$*, where* p *traverses all* q *prime factors. LWE polynomials, which reduces the success advantage of the problem by at most* $\sum_p p^{-n}$*, where* p *iterates over all the prime factors of* q*. The proof can be found in reference [40].*

Definition 7 (Key Homomorphism)**.** *Key homomorphism in cryptography refers to a property of cryptographic algorithms where transformations on keys can be correlated directly with transformations on ciphertexts [41]. This means that operations performed on keys can have equivalent operations on ciphertexts that preserve the structure of the data being encrypted.*

For example, if there is a cryptographic system that supports key homomorphism, and two keys k_1 and k_2, performing an operation on these keys (like addition or multiplication) to produce a new key k_3 will correlate with a similar operation on ciphertexts encrypted with k_1 and k_2 to produce a new ciphertext that would decrypt correctly under k_3.

In a system that supports multi-key homomorphism, there are several keys $k_1, k_2, \cdots, k_n$. Operations performed across these keys, such as combinations or aggregations (e.g., summation, multiplication), yield a new key k_{new}. The crucial aspect is that an equivalent operation on ciphertexts encrypted with $k_1, k_2, \cdots, k_n$ results in a new ciphertext that, when decrypted with k_{new}, reveals a data transformation that precisely reflects the key operations. This process can support complex data interactions securely and efficiently, providing significant flexibility in distributed cryptographic systems.

3.2. GSW13

To elucidate the distinctions between our proposed mMFHE scheme and prior works, we provide a detailed exposition of the GSW13. The GSW13, a scheme predicated on the LWE problem, is distinguished by its minimal ciphertext expansion during homomorphic operations. We employ the following formal representation for clarity:

Remark 1. *Let us consider the positive integers* m, m', n*, and* q *(with* $m > n\lceil logq \rceil$*) and a matrix* $\mathbf{T} \in \mathbb{Z}_q^{n \times m}$*. It can be demonstrated that there exists a matrix* $\mathbf{G}$*, belonging to the set of* $n \times m$ *matrices over the field of integers modulo* q*, and an inverse function* $\mathbf{G}^{-1}$*, such that* $\mathbf{G^{-1}}(\mathbf{T})$ *is a binary matrix. Furthermore, it can be demonstrated that the matrix* $\mathbf{GG^{-1}}(\mathbf{T})$ *is equal to* $\mathbf{T}$*. The multiplication of a matrix by* $\mathbf{G}$ *results in a bitwise combination of its elements. In contrast, the inverse of* $\mathbf{G}$*, denoted* $\mathbf{G^{-1}}$*, facilitates the bitwise decomposition of these elements. The operational specifics of the GSW13 are outlined as follows:*

- *GSW.Setup*$(1^\lambda, 1^d)$*: Initiate the setup by defining the lattice dimension* $n = n(\lambda, d)$*, where* λ *is the security parameter, and* d *is an integer specifying the maximum circuit depth permissible. Select a noise distribution* $\chi = \chi(\lambda, d)$*, bounded by* B_χ*, and determine a modulus* q *as* $q = B_\chi 2^{\omega(d\lambda log\lambda)}$*. This configuration is chosen to satisfy the Learning With Errors problem* $LWE_{n-1,q,\chi}$*. Set* $m = nlogq + \omega(log\lambda)$ *as the parameter defining the matrix dimensions.*
- *GSW.KeyGen: Generate a uniform random matrix* $\mathbf{B} \in \mathbb{Z}_q^{(n-1)\times m}$ *and a vector* $\mathbf{s}$ *in* $\mathbb{Z}_q^{n-1}$*. Compute the vector* $\mathbf{b}$ *as* $\mathbf{b} = \mathbf{sB} + \mathbf{e}$*, where* $\mathbf{e}$ *is an error vector sampled from a discrete*

distribution over $\mathbb{Z}_q$. *The public key is then given by* $\mathbf{A} = \begin{pmatrix} \mathbf{B} \\ \mathbf{b} \end{pmatrix} \in \mathbb{Z}_q^{n \times m}$, *and the private key is* $sk = \mathbf{t} = (-\mathbf{s}, 1) \in \mathbb{Z}_q^n$.

- *GSW.Encrypt: For a plaintext bit message* μ, *construct a uniform random matrix* $\mathbf{R} \in \{0,1\}^{m \times m}$, *and output the ciphertext* $\mathbf{C} = \mathbf{AR} + \mu\mathbf{G}$.
- *GSW.Decrypt: Define the vector* $\mathbf{w}$ *as consisting of zeros, except for the end position set to* $\lceil q/2 \rceil$. *For a given ciphertext* $\mathbf{C}$, *compute* $v = \mathbf{tCG^{-1}}(\mathbf{w^T}) \approx \mu\lceil q/2 \rceil$. *The decryption yields 0 if* v *is closer to 0 than to* $\lceil q/2 \rceil$, *and 1 otherwise.*
- *GSW.Evaluation: Define homomorphic operations as follows:*

$$\mathbf{ADD}(\mathbf{C}_1, \mathbf{C}_2)\text{: } \textit{Output } \mathbf{C}_1 + \mathbf{C}_2 \in \mathbb{Z}_q^{n \times m}.$$
$$\mathbf{MULT}(\mathbf{C}_1, \mathbf{C}_2)\text{: } \textit{Output } \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2) \in \mathbb{Z}_q^{n \times m}.$$
$$\mathbf{NAND}(\mathbf{C}_1, \mathbf{C}_2)\text{: } \textit{Output } \mathbf{G} - \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2).$$

For a more comprehensive analysis and construction details of $\mathbf{G}$ *and* $\mathbf{G}^{-1}$, *readers are referred to additional literature [15].*

4. FL Scheme Based on mMFHE

This section presents our MKFHE scheme, which is based on the key homomorphism of GSW13 under the CRS model, as well as the multi-bit FHE scheme, which is inspired by GSW13 presented by Li et al. [42], and proves that our scheme also satisfies the key linear homomorphism.

4.1. mMFHE

Assuming a CRS model and given the security parameters λ, we set t to the quantity of secret keys and the total number of bits in the message. The i-th participant is designated P_i, $i \in [N]$, and N is the number of participants. It is stipulated that each participant has t messages $\mu_j \in \{0,1\}$, $j \in [t]$. We now give the formal details.

- Setting parameters: $\mathsf{params} \leftarrow \mathsf{Setup}(1^\lambda, 1^L)$:
  $\mathsf{Setup}(\cdot)$ takes as input the safety parameter λ and the maximum depth L of the circuit. The mode is $q = q(\lambda)$, the dimension of the lattice $n = n(\lambda)$, $m = m(\lambda, L) = O(n \log q)$, and the distribution of the errors $\chi = \chi(\lambda, d)$ such that $(m, n, q, \chi) - LWE$ With the assumption that the security of at least 2^λ is achieved against a known attack, a uniformly randomized matrix $\mathbf{B} \leftarrow \mathbb{Z}_q^{n \times m}$ (as the common string) is chosen. Furthermore, let $l = \lfloor \log q \rfloor + 1$, $M = (n + t) \cdot l$ and output $\mathbf{params} = (n, q, \chi, m)$, $\mathbf{B}$.
- Key generation: $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{params})$: For the j-th message μ_j of the i-th participant P_i, select the sample $\mathbf{a}_j^T = (a_{j,1}, \cdots, a_{j,n}) \in \mathbb{Z}_q^{1 \times n}$ and output $\mathsf{sk}_j := \mathbf{s}_j = (\mathbf{I}_j \mid -\mathbf{a}_j^T)^T \in \mathbb{Z}_q^{(n+t) \times 1}$. The important thing to note here is that $\mathbf{v}_j = \mathsf{PowerOf2}(\mathbf{s}_j)$. Most importantly, the private key matrix $\mathbf{sk}_i := \boldsymbol{S}_i = [\mathsf{sk}_1, \cdots, \mathsf{sk}_t] = [\mathbf{s}_1, \cdots, \mathbf{s}_t] \in \mathbb{Z}_q^{(n+t) \times t}$; choose $\mathbf{e}_j \leftarrow \chi^{m \times 1}$, $j \in [t]$, then calculate $\mathbf{b}_j = \mathbf{B} \cdot \mathbf{a}_j + \mathbf{e}_{(p \bmod q)}$, and output $\mathbf{pk}_i := \boldsymbol{A}_i = [\mathbf{b}_1 \mid \cdots \mid \mathbf{b}_t \mid \mathbf{B}] \in \mathbb{Z}_q^{m \times (n+t)}$, where $\mathbf{pk}$ has size $\mathcal{O}(nm \cdot \log^2 q)$. Finally, we observe that $\boldsymbol{A} \cdot \mathbf{s}_i = \mathbf{e}_i$ and $\boldsymbol{A} \cdot \boldsymbol{S} = [\mathbf{e}_1, \cdots, \mathbf{e}_t]$.
- Encryption: $\mathbf{C} \leftarrow \mathsf{Enc}(\mathsf{params}, \mathsf{pk}, \mathbf{M})$: To encrypt a t-bit message, where each bit u_j belongs to the set $0, 1$ and $j \in [t]$, we commence by sampling a uniform matrix $\mathbf{R}$ from the set $\mathbf{R} \leftarrow \{0,1\}^{m \times M}$. Subsequently, the individual bits of the message are embedded into a diagonal matrix $\mathbf{U}$, expressed as $\mathbf{U} = \mathrm{diag}(u_1, \cdots, u_t)$. This matrix $\mathbf{U}$ serves as the basis for constructing the plaintext matrix. The precise method for forming the plaintext matrix will be elaborated as follows:

$$\mathbf{M} = \begin{pmatrix} \mathbf{U}_{t \times t} & \mathbf{0}_{t \times n} \\ \mathbf{0}_{t \times n} & \mathbf{E}_{n \times n} \end{pmatrix} \in \{0,1\}^{(n+t) \times (n+t)}, \tag{1}$$

where the matrices $\mathbf{U} \in \mathbb{Z}_q^{t\times t}$ and $\mathbf{E} \in \{0,1\}^{n\times n}$ are diagonal matrices, i.e., $\mathbf{U} = \mathrm{diag}(u_1, \cdots, u_t)$ and $\mathbf{E} = \mathrm{diag}(1, \cdots, 1)$, which are also the two division matrices of the plaintext matrix $\mathbf{M}$. Upon receipt of the public keys from all participants, each participant is tasked with computing the aggregate public key: $PK = \mathbf{A} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{A}_i$. Calculate and write $\mathbf{C} = \mathbf{M}\cdot\mathbf{G} + \mathbf{A}^T\cdot\mathbf{R} \pmod q \in \mathbb{Z}_q^{(n+t)\times M}$, $\mathbf{G} = \mathrm{BitDecom}\mathbf{p}^{-1}(\mathbf{I}_{n+t})$ $= (g^T \otimes \mathbf{I}_{n+t}) \in \mathbb{Z}_q^{(n+t)\times(n+t)\cdot l}$, where $\mathbf{I}_{n+t}$ denotes the $(n+t)$ dimensional unit matrix; hence, $g^T = [2^0, 2^1, \cdots, 2^{l-1}] \in \mathbb{Z}_q^l, l = \lceil \log q \rceil = \lfloor \log q \rfloor + 1$, for $m \geqslant n\lceil \log q \rceil$, i.e., $m = \mathcal{O}(n(\log q))$.

- Evaluation: Upon receipt of the ciphertexts from all participants, each participant is evaluated for deterministic homomorphism based on the circuit CIR, and a final ciphertext $\hat{\mathbf{C}}$ contains Add and Mult in more detail:
  $\mathsf{Add}(\mathbf{C}_1, \mathbf{C}_2)$: Output:

$$\hat{\mathbf{C}} = \mathbf{C}_1 + \mathbf{C}_2 = (\mathbf{M}_1 + \mathbf{M}_2)\mathbf{G} + \mathbf{A}^T(\mathbf{R}_1 + \mathbf{R}_2) \in \mathbb{Z}_q^{(n+t)\times M} \tag{2}$$

  $\mathsf{Mult}(\mathbf{C}_1, \mathbf{C}_2)$: Outputting the matrix product, as $\mathbf{C}_2 = \mathbf{M}_2\cdot\mathbf{G} + \mathbf{A}^T\cdot\mathbf{R}_2$, one obtains

$$\begin{aligned}\mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2) &= \left(\mathbf{M}_1\cdot\mathbf{G} + \mathbf{A}^T\cdot\mathbf{R}_1\right)\cdot\mathbf{G}^{-1}(\mathbf{C}_2) = \mathbf{M}_1\cdot\mathbf{C}_2 + \mathbf{A}^T\cdot\mathbf{R}_1\cdot\mathbf{G}^{-1}(\mathbf{C}_2)\\ &= \mathbf{M}_1\mathbf{M}_2\cdot\mathbf{G} + \mathbf{A}^T\mathbf{R}_1\cdot\mathbf{G}^{-1}(\mathbf{C}_2) + \mathbf{M}_1\mathbf{A}^T\mathbf{R}_2 \in \mathbb{Z}_q^{(n+t)\times M}\end{aligned} \tag{3}$$

  Additionally, this configuration facilitates the computation of homomorphic NAND gates. The operation is executed by generating the output from the expression $\mathbf{G} - \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2)$.
- Decryption $\mathbf{U} \leftarrow \mathsf{Dec}(\mathsf{params}, \mathsf{sk}, \hat{\mathbf{C}})$: We denote a matrix

$$\mathbf{W}^T = \left(\begin{array}{ccc|c} \lceil q/2\rceil, & \cdots, & 0 & \mathbf{0}^{1\times n}\\ \vdots & \ddots & \vdots & \vdots\\ 0, & \cdots, & \lceil q/2\rceil & \mathbf{0}^{1\times n}\end{array}\right) \in \mathbb{Z}_q^{t\times(n+t)} \tag{4}$$

  Participant P_i constructs its own key matrix $S_i = (\mathbf{s}_1, \cdots, \mathbf{s}_t) = \left(\frac{\mathbf{I}}{-\mathbf{t}_1, \cdots, -\mathbf{t}_t}\right) \in \{0,1\}^{(n+t)\times t}$. According to χ, select random vector $\sigma_j'' \in \mathbb{Z}_q^{n+t-tl}, \sigma_j' = (0, ..., 0, \sigma_j'') \in \mathbb{Z}_q^{n+t}$, and let the random vector matrix $\sigma_i = (\sigma_1', ..., \sigma_j', ..., \sigma_t') \in \mathbb{Z}_q^{(n+t)\times t}$ Generate and publish decryption shares $d_i = S_i^T\cdot\hat{\mathbf{C}} + \sigma_i \in \mathbb{Z}_q^{(n+t)\times t}$. After obtaining all the decryption shares d_i from the other participants, P_i computes

$$D = \frac{1}{N}\sum_{i=1}^{N} d_i = \frac{1}{N}\sum_{i=1}^{N} S_i^T\cdot\hat{\mathbf{C}} + \frac{1}{N}\sum_{i=1}^{N}\sigma_i \triangleq \mathbf{S}^T\hat{\mathbf{C}} + \sigma \tag{5}$$

  where $\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N} S_i$, $\mathbf{S}^T\hat{\mathbf{C}} = \mathbf{S}^T\mathbf{A}^T\mathbf{R} + \mathbf{S}^T\mathbf{M}\mathbf{G} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{e}_{i,1}, ..., \boldsymbol{e}_{i,t})^T\mathbf{R} + \mathbf{S}^T\mathbf{M}\mathbf{G}$ and $\mathbf{V} = \mathbf{S}^T\hat{\mathbf{C}}\cdot\mathbf{G}^{-1}(\mathbf{W}^T)$.
  Output decrypted message $\mathbf{U} = \left|\lfloor\frac{\mathbf{V}}{q/2}\rceil\right|$, where

$$\begin{aligned}\mathbf{V} &= \mathbf{S}^T\hat{\mathbf{C}}\cdot\mathbf{G}^{-1}(\mathbf{W}^T)\\ &= \lceil\frac{q}{2}\rceil\cdot\begin{pmatrix} u_{1,1} & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & u_{t,t}\end{pmatrix} + \frac{1}{N}\begin{pmatrix}\mathbf{e}_1^T\mathbf{R}\\ \vdots\\ \mathbf{e}_t^T\mathbf{R}\end{pmatrix}\cdot\mathbf{G}^{-1}(\mathbf{W}^T)\\ &= \lceil\frac{q}{2}\rceil\cdot\mathbf{U}_{t\times t}\end{aligned} \tag{6}$$

4.2. Key Homomorphism of mMFHE

In order to facilitate comprehension, this section will introduce and prove the key homomorphism of mMFHE in the CRS model.

Theorem 3. *Consider an environment in the GSW13 encryption scheme where all participants use the same random matrix* $\mathbf{B}$ *to generate their public keys. In this case, the GSW13 scheme is able to realize the additive homomorphism property of the key. Specifically, if all participants* P_i*, for* $i \in [N]$*, construct their own public key as* $\mathbf{A}_i$ *and accumulate these public keys as* $pk = \mathbf{A} = \sum_{i=1}^{N} \mathbf{A}_i$ *for encrypting a single-bit message* μ*, the resulting cipher text is* $\mathbf{C} = \sum_{i=1}^{N} \mathbf{A}_i\mathbf{R} + \mu\mathbf{G}$*. In this configuration, the corresponding valid private key* $sk = \mathbf{t} = \sum_{i=1}^{N} \mathbf{t}_i$ *is able to correctly decrypt the ciphertext* $\mathbf{C}$ *into the message* μ*. The proof of this theorem was given in reference [40], and below, we will show that it still holds in mMFHE.*

Proof. Assume that the participant P_i generates private key matrix $\mathbf{sk}_i := \boldsymbol{S}_i = [\mathsf{sk}_1, \cdots, \mathsf{sk}_t] = [\mathbf{s}_1, \cdots, \mathbf{s}_t]$ according to the $\mathsf{mFHE.KeyGen}()$ algorithm, where $\mathbf{s}_j = (\mathbf{I}_j \mid -\mathbf{a}_j^T)^T$, $\mathbf{a}_j^T = (a_{j,1}, \cdots, a_{j,n}) \in \mathbb{Z}_q^{1\times n}$ are randomly selected samples; the public key matrix is $\mathbf{pk}_i := \boldsymbol{A}_i = [\mathbf{b}_1 \mid \cdots \mid \mathbf{b}_t \mid \mathbf{B}]$, where $\mathbf{b}_j = \mathbf{B} \cdot \mathbf{a}_j + \mathbf{e}_{p \bmod q}$, $\mathbf{e}_j \leftarrow \chi^{m\times 1}, j \in [t]$. Then,

$$\begin{aligned}
\mathbf{A} \cdot \mathbf{S} &= \sum_{i=1}^{N} \boldsymbol{A}_i \cdot \sum_{i=1}^{N} \boldsymbol{S}_i \\
&= \left(\sum_{i=1}^{N} \boldsymbol{b}_{1,i}, ..., \sum_{i=1}^{N} \boldsymbol{b}_{t,i}, N \cdot \boldsymbol{B}\right) \left(\sum_{i=1}^{N} \boldsymbol{s}_{1,i}, ..., \sum_{i=1}^{N} \boldsymbol{s}_{t,i}\right) \\
&= \left(\sum_{i=1}^{N} \boldsymbol{b}_{1,i}, ..., \sum_{i=1}^{N} \boldsymbol{b}_{t,i}, N \cdot \boldsymbol{B}\right) \left(\sum_{i=1}^{N} (\boldsymbol{I}_1| - \boldsymbol{a}_1^T)^T, ..., \sum_{i=1}^{N} (\boldsymbol{I}_t| - \boldsymbol{a}_t^T)^T\right) \\
&= N \sum_{i=1}^{N} (\boldsymbol{e}_{1,i}, ..., \boldsymbol{e}_{t,i}) \approx \mathbf{0}
\end{aligned} \tag{7}$$

In the case that all error vectors $\mathbf{e}_{i,j}$ are small enough, the above equation holds, i.e., we can obtain the plaintext message matrix $\mathbf{U}$ according to the decryption algorithm of mMFHE. It can be seen that under the assumption that $\mathbf{B}$ is CRS, mMFHE still satisfies the additive key homomorphism.

However, the validity of the above theorem relies on two key assumptions: one is that the number of participants N must be kept small, and the other is that all the error vectors $\mathbf{e}_{i,j}$ also need to be sufficiently small. These conditions pose rather stringent constraints in practical applications. In view of this, we propose to consider another form of key homomorphism for a wider range of application scenarios. □

Theorem 4. *The GSW13 encryption scheme possesses the linear homomorphism property of the key. More specifically, if the same setup as before is used and the public key is defined as* $pk = \mathbf{A} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{A}_i$*, and this public key is used to encrypt a single bit message* μ *to obtain the ciphertext* $\mathbf{C}$*, then the corresponding valid private key is* $sk = \mathbf{t} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{t}_i$*. This private key can accurately decrypt the ciphertext* $\mathbf{C}$*, thus recovering the original message* μ*.*

Proof. Similarly, we have

$$\begin{aligned}
\mathbf{A} \cdot \mathbf{S} &= \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{A}_i \cdot \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{S}_i \\
&= \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{e}_{1,i}, ..., \boldsymbol{e}_{t,i}) \approx \mathbf{0}
\end{aligned} \tag{8}$$

It can thus be shown that mMFHE satisfies the linear homomorphism property of the key. □

4.3. Threat Model

In contemporary encryption protocols, it is commonly postulated that participants align with the "honest but curious" model. In accordance with this assumption, participants faithfully execute the prescribed protocol while simultaneously seeking any feasible means to derive confidential data contained within the output generated throughout the protocol's execution.

In our research, we employ MKFHE to safeguard data privacy within a federated learning framework. Specifically, we posit that both the server and all remote participants operate under the honest-but-curious assumption. This implies that while they conscientiously adhere to the protocol's specifications, they concurrently endeavor to derive personal information about other participants from the shared data during the course of the protocol's execution. Additionally, we entertain the possibility of collusion among the participants and the server.

To elucidate, we consider a specific adversarial scenario with $k = N - 1$ participants, where N denotes the total number of participants and k the number of conspirators, collaborating with the server to compromise the confidentiality of a targeted participant. This scenario highlights the potential vulnerabilities and the requisite safeguards necessary in the design of secure federated learning systems.

4.4. Our PPFL Scheme

Building upon the mMFHE scheme proposed in the preceding section of this document, we introduce a privacy-centric FL scheme. This scheme is predicated on the assumption that all participants actively engage and contribute to the model training process in each iteration.

With reference to the FedAvg algorithm, the general process of our PPFL scheme is illustrated in Figure 1. In our proposed PPFL scheme, based on the mMFHE scheme and in referencing the FedAvg algorithm, the process is outlined as shown in Figure 1. During each model aggregation round, clients train the received global model using their local data. Clients independently decide the number of training rounds and parameters based on their resources and data volume. After training, clients encrypt their model parameters using aggregated public key PK and send the encrypted results back to the central server.

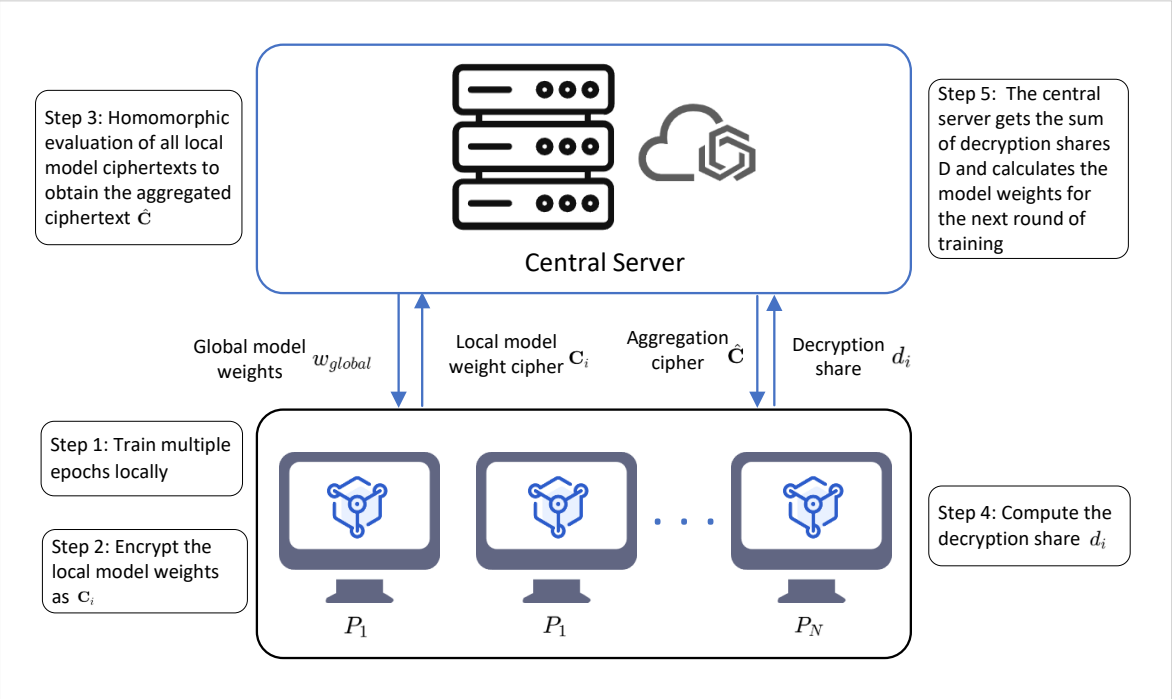


Figure 1. Model of mMFHE-based PPFL scheme.

The central server collects all encrypted model parameters from the clients and performs a ciphertext aggregation to obtain the aggregated result ciphertext $\hat{\mathbf{C}}$. Then, clients use their private key matrices $\mathbf{sk}_i$ and the $\hat{\mathbf{C}}$ from the server to compute decryption shares d_i and send these shares back to the central server. The server aggregates all decryption shares to obtain the plaintext form of the model aggregation result and updates the global model using the average aggregation method. This process is repeated until the model meets the predetermined performance standards or completes the specified number of training rounds. The aforementioned flow of encryption and decryption is illustrated in Figures 2 and 3, and a detailed description of each step is provided below.

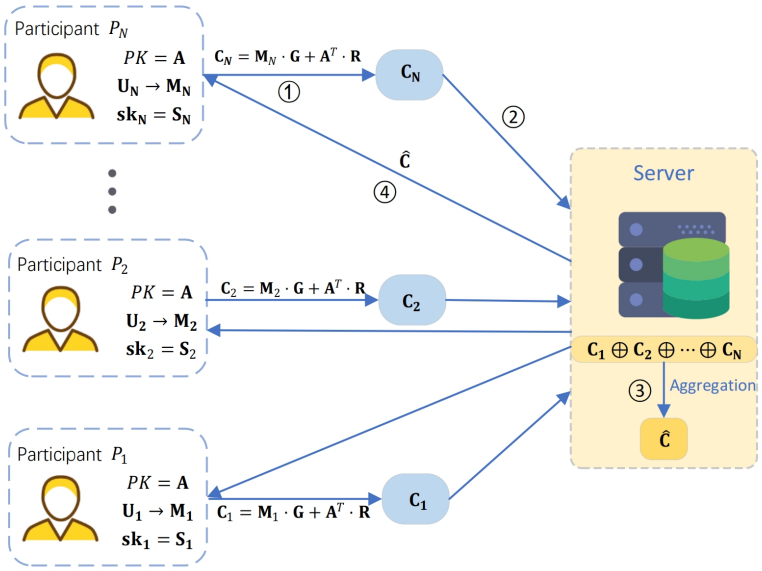


Figure 2. Encryption process.

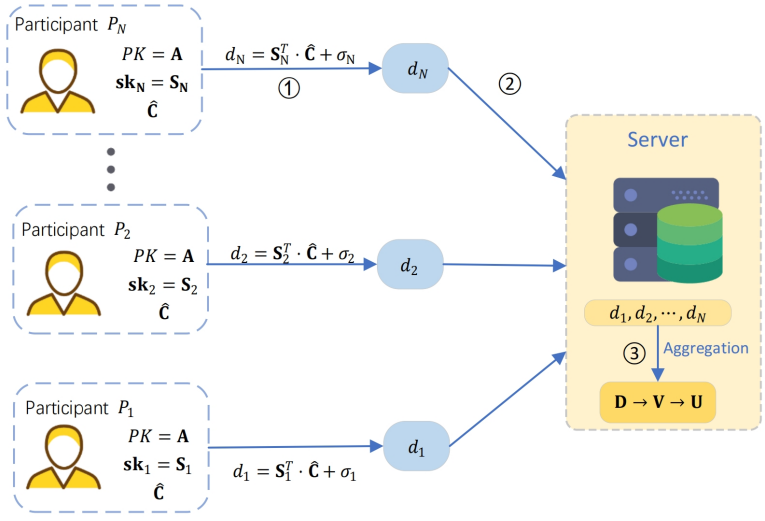


Figure 3. Decryption process.

Initialization: The central server performs the initialization of the global model; executes the Setup$(\cdot)$ function; sets the ciphertext mode q, the lattice dimensions n, m, the error distribution χ, and the length of the message t according to λ and L; selects the uniform random matrix $\mathbf{B} \leftarrow \mathbb{Z}_q^{n \times m}$; and returns public parameters params $= (n, q, \chi, m)$, public

matrix $\mathbf{B}$, and t. Each remote client P_i ($i \in [N]$) selects the sample $\mathbf{a}_j^T = (a_{j,1}, \cdots, a_{j,n}) \in \mathbb{Z}_q^{1\times n}$ in order to generate the private key matrix:

$$\mathbf{sk}_i := \boldsymbol{S}_i = [\mathsf{sk}_1, \cdots, \mathsf{sk}_t] = [\mathbf{s}_1, \cdots, \mathbf{s}_t] \in \mathbb{Z}_q^{(n+t)\times t} \tag{9}$$

where $\mathsf{sk}_j := \mathbf{s}_j = (\mathbf{I}_j \mid -\mathbf{a}_j^T)^T \in \mathbb{Z}_q^{(n+t)\times 1}$ is the j-th bit of the message corresponding to the key, $j \in [t]$. Subsequently, select $\mathbf{e}_j \leftarrow \chi^{m\times 1}$ and compute $\mathbf{b}_j = \mathbf{B} \cdot \mathbf{a}_j + \mathbf{e}_{p \ mod \ q}$ to generate and return the public key matrix to the centralized server:

$$\mathbf{pk}_i := \boldsymbol{A}_i = [\mathbf{b}_1 \mid \cdots \mid \mathbf{b}_t \mid \mathbf{B}] \in \mathbb{Z}_q^{m\times(n+t)} \tag{10}$$

The central server receives the public key matrix from all participating clients, calculates

$$PK = \mathbf{A} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{A}_i \tag{11}$$

and returns the aggregated public key PK to all participants (clients).

Step 1: Local Training: At the beginning of each aggregation round, each participant P_i receives the global model weights w_{global} from the central server. They then utilize their own locally held data to train the model, and after enough epochs, the participant P_i generates a locally held model with the weights w_i.

Step 2: Model Weight Encryption: Let the t-bit message $u_i \in \{0,1\}$ be the encoded plaintext input of w_i and generate the corresponding plaintext matrix $\mathbf{M}_i$ (see Section 4.1 above); participant P_i samples a homogeneous matrix $\mathbf{R}_i \leftarrow \{0,1\}^{m\times M}$ and encrypts $\mathbf{M}_i$ with the aggregation public key $PK = \mathbf{A}$ to obtain the ciphertext

$$\mathbf{C}_i = \mathbf{M}_i \cdot \mathbf{G} + \mathbf{A}^T \cdot \mathbf{R} \pmod q \in \mathbb{Z}_q^{(n+t)\times M} \tag{12}$$

and sends C_i to the central server.

Step 3: Homomorphic Evaluation: The central server performs homomorphic evaluation on the ciphertexts after receiving the ciphertexts with model weights sent by each participant. For ease of understanding, here in this paper, there is an example of homomorphic addition, which yields

$$\hat{\mathbf{C}} = \sum_{i=1}^{N} \mathbf{C}_i = \sum_{i=1}^{N} \mathbf{M}_i\mathbf{G} + \mathbf{A}^T \sum_{i=1}^{N} \mathbf{R}_i \in \mathbb{Z}_q^{(n+t)\times N}; \tag{13}$$

The server then sends $\hat{\mathbf{C}}$ to all participants. And if we want to perform homomorphic multiplication, in the case of two participants,

$$\mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2) = \mathbf{M}_1\mathbf{M}_2 \cdot \mathbf{G} + \mathbf{A}^T\mathbf{R}_1 \cdot \mathbf{G}^{-1}(\mathbf{C}_2) + \mathbf{M}_1\mathbf{A}^T\mathbf{R}_2 \in \mathbb{Z}_q^{(n+t)\times M} \tag{14}$$

Step 4: Calculation of decryption share: In MKFHE, the decryption of the ciphertext necessitates the input of all participating members; in order to decrypt $\hat{\mathbf{C}}$, it is necessary for each participant P_i to calculate its decryption share d_i using its private key matrix S_i:

$$d_i = \boldsymbol{S}_i^T \cdot \hat{\mathbf{C}} + \sigma_i \in \mathbb{Z}_q^{(n+t)\times t} \tag{15}$$

where $\sigma_i = \left(\sigma'_1, \ldots, \sigma'_j, \ldots, \sigma'_t\right) \in \mathbb{Z}_q^{(n+t)\times t}$ is the matrix of random vectors, and σ'_j is the error vector chosen from χ.

Step 5: Model Aggregation: The server calculates the aggregation of the decryption shares after obtaining the decryption shares d_i of all participants and obtains

$$D = \frac{1}{N}\sum_{i=1}^{N} d_i = \frac{1}{N}\sum_{i=1}^{N} S_i^T \cdot \hat{\mathbf{C}} + \frac{1}{N}\sum_{i=1}^{N} \sigma_i \triangleq \mathbf{S}^T\hat{\mathbf{C}} + \sigma \tag{16}$$

where $\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N} S_i$. Then, the server computes $\mathbf{V} = \mathbf{S}^T\hat{\mathbf{C}} \cdot \mathbf{G}^{-1}(\mathbf{W}^T)$, obtains the decrypted message $\mathbf{U} = \left| \lfloor \frac{\mathbf{V}}{q/2} \rceil \right|$, decodes $\mathbf{U}$ to obtain the aggregation of the local model weights of each participant, and computes the new weight, which will be used as the global model weight in the next round.

5. Security Analysis

This section presents a discussion of the manner in which the presented scheme ensures the confidentiality of the model weights within the FL system. This, in turn, ensures the privacy of the data hosted on the distributed devices by each FL participant. In order to characterize the security of our scheme for each potential adversary in the CRS model, we will employ the following theorem.

Theorem 5 (Semantic Safety)**.** *mMFHE is IND-CPA-safe if the parameters params* $= (n, q, \chi, m, t)$ *are chosen to align with the difficulty presumption of the* $LWE_{n,m,q,\chi}$ *problem, and* $m = O(n \log q)$.

Proof. (1) The public key matrix $\boldsymbol{A} = [\mathbf{b}_1 \mid \cdots \mid \mathbf{b}_t \mid \mathbf{B}]$ and the $m \times (n+t)$-rank matrices uniformly chosen from $\mathbb{Z}_q$ are statistically indistinguishable. A detailed proof of this is given in reference [42]. (2) The ciphertext matrices $\mathbf{C}$ and the $(n+t) \times M$-rank matrices selected uniformly from $\mathbb{Z}_q$ are computationally indistinguishable. A detailed proof of this is given in reference [43]. □

Theorem 6 (Security Against Honest-but-Curious Servers)**.** *In the framework of our proposed scheme, it is established that a server, operating under the honest-but-curious model, is incapable of deducing any private information from any of the remote participants.*

Proof. In the mMFHE-based PPFL scheme, the remote participant P_i will send two kinds of information to the server, i.e., C_i and d_i. In Step 2, P_i will encrypt its own locally held model gradient to obtain C_i using mMFHE and send it to the server, where $C_i = \mathbf{M}_i \cdot \mathbf{G} + \mathbf{A}^T \cdot \mathbf{R}(\mathrm{mod} q)$. From Theorem 5, mMFHE is IND-CPA-secure, so C_i will not disclose to the server any of the M_i information.

In Step 4, P_i partially decrypts the aggregated ciphertext $\hat{\mathbf{C}}$ using the private key matrix S_i to obtain the decryption share $d_i = S_i^T \cdot \hat{\mathbf{C}} + \sigma_i$ and sends it to the server, which aggregates the decryption shares of all parties to obtain $D = \mathbf{S}^T\hat{\mathbf{C}} + \sigma$. Since the first $(n+t) \times t$ components in σ_i and σ are subject to B_χ boundedness, these two equations form the some-are-errorless LWE problem. Therefore, in Steps 4 and 5, P_i publishes its d_i without revealing its private key or the aggregation key. After the aggregation is decrypted, the server is only able to ascertain the total sum of the model weights, rather than the specific weights of the individual participants.

It thus follows that our scheme ensures that the individual weights remain confidential, thereby guaranteeing the privacy of data belonging to various participants distributed across remote devices. It is assured that the server, upon receiving information, remains unable to deduce any private details regarding the participants. □

Theorem 7 (Security for Honest-but-Curious Users)**.** *Within the framework of the mMFHE-based PPFL scheme, it is established that an honest-but-curious user is incapable of deriving any private information about other users by intercepting shared information.*

Proof. In our proposed model, the model update for each participant P_i is secured using the mMFHE encryption. Each device independently selects an S_i to generate a corresponding A_i. Concurrently, all participants collaborate to obtain an aggregated public key, which is then employed for the encryption of their model weights. To further enhance security, an error is introduced into the decryption share, thereby protecting the keys of each user. With Theorem 5, the mMFHE has been proven to be IND-CPA-secure. Consequently, an honest-but-curious user is unable to extract meaningful information from data uploaded by other participants, thus maintaining confidentiality across the network. □

Theorem 8 (Security Against User–Server Collusion:)**.** *Assume a scenario where $k \leq N-1$ users engage in collusion with the server, yet do not disclose the model updates from other users, where N represents the total number of users and k is the number of users in collusion with the server.*

Proof. In addressing a scenario with a static semi-malicious adversary A involving exactly $N-1$ corrupt users conspiring with the server, we establish a Probabilistic Polynomial-Time (PPT) simulator, denoted as Sim, and designate P_h to represent the sole honest participant. The simulator Sim undertakes the following operations on behalf of the honest party:

In Step 2, the simulator Sim employs a zero value as a placeholder for the genuine input of the honest participant P_h during the encryption process. Subsequently, Sim acquires the inputs and private keys of the $N-1$ compromised parties from the "evidence tape". These inputs are then supplied to an "ideal machine", from which the output y is derived. Additionally, Sim is capable of retrieving the homomorphically computed ciphertext $\hat{\mathbf{C}}$ from the server.

Using this gathered information, Sim computes the simulated partial decryption result for the honest party P_h as follows:

$$\boldsymbol{\rho}'_h \leftarrow Sim(y, \hat{\mathbf{C}}, h, \{\mathbf{sk}_i\}_{i\in[N]\setminus\{h\}}) \tag{17}$$

In Step 4, rather than forwarding the authentic decryption result, Sim transmits this simulated partial decryption result to the server. We define a series of hybrid games to prove the indistinguishability between the real and simulated scenarios, i.e., $IDEAL_{F,Sim,Z} \overset{\text{comp}}{\approx} REAL_{\pi,A,Z}$, where Z represents a specific environment. In this context, the game $REAL_{\pi,A,Z}$ represents the execution of protocol π_f in the real environment Z with a semi-honest adversary. The game $HYB_{\pi,A,Z}$ is essentially the same as the game $REAL_{\pi,A,Z}$, but assumes that P_h obtains all private keys $\{\mathbf{sk}_i\}$ for $i \in [N]\setminus\{h\}$ after Step 2, and in Step 4, it sends the simulated partial decryption $\rho'_h \leftarrow Sim(y, \hat{\mathbf{C}}, h, \{\mathbf{sk}_i\}_{i\in[N]\setminus\{h\}})$ to the server instead of the real decryption. In contrast, the game $IDEAL_{F,Sim,Z}$ replaces the real input of P_h with 0 for encryption and sends it to the server in Step 2, while the other Steps remain consistent with the game $HYB_{\pi,A,Z}$. □

Lemma 1. $REAL_{\pi,A,Z} \overset{\text{stat}}{\approx} HYB_{\pi,A,Z}$

Proof. The distinction between the two scenarios is marked by the substitution of the simulated decryption ρ'_h for the actual partial decryption ρ_h conducted by participant P_h. Therefore, let $\mathbf{V} = \mathbf{U}\lceil\frac{q}{2}\rceil + \boldsymbol{e}'$, with the simulated decryption algorithm being

$$\begin{aligned}\boldsymbol{\rho}'_h &= N\cdot\mathbf{U}\cdot\lceil\frac{q}{2}\rceil + N\boldsymbol{e}' - \sum_{i\neq h}\mathbf{S}_i\hat{\mathbf{C}}\mathbf{G}^{-\mathbf{1}}(\mathbf{W}^T) + \boldsymbol{\sigma}'_h \\ &= N\cdot\mathbf{U}\cdot\lceil\frac{q}{2}\rceil + N\boldsymbol{e}' + \boldsymbol{\sigma}'_h - \sum_{i\neq h}\mathbf{V}_i\end{aligned} \tag{18}$$

where $\boldsymbol{e}' \leftarrow \chi$, $\boldsymbol{\sigma}'_h \leftarrow \chi$.

As $\mathbf{V} = \frac{1}{N}\sum_{i\in[N]}\mathbf{V}_i = \mathbf{U}\cdot\lceil\frac{q}{2}\rceil + \boldsymbol{e}' \Rightarrow N\boldsymbol{e}' = \sum_{i\in[N]}\mathbf{V}_i - N\cdot\mathbf{U}\cdot\lceil\frac{q}{2}\rceil$, the real decryption of P_h is

$$\begin{aligned}\boldsymbol{\rho}_h &= \mathbf{V}_h + \boldsymbol{\sigma}_h \\ &= \sum_{i\in[N]}\mathbf{V}_i - \sum_{i\neq h}\mathbf{V}_i + \boldsymbol{\sigma}_h \\ &= N\boldsymbol{e}' + N\cdot\mathbf{U}\cdot\lceil\frac{q}{2}\rceil - \sum_{i\neq h}\mathbf{V}_i + \boldsymbol{\sigma}_h\end{aligned} \tag{19}$$

where $\boldsymbol{\sigma}_h \leftarrow \chi$.

It can be readily observed that the values $\boldsymbol{\sigma}_h$ and $\boldsymbol{\sigma}'_h$ are statistically indistinguishable, thereby establishing that $\boldsymbol{\rho}_h$ and $\boldsymbol{\rho}'_h$ are likewise indistinguishable from one another. Therefore, it is established that $REAL_{\pi,A,Z} \overset{\text{stat}}{\approx} HYB_{\pi,A,Z}$. □

Lemma 2. $HYB_{\pi,A,Z} \overset{\text{comp}}{\approx} IDEAL_{F,Sim,Z}$

Proof. In these two games, only the ciphertext generated by P_h differs. In accordance with Theorem 5, the ciphertexts are deemed computationally indistinguishable, thereby rendering the two games indistinguishable as well. By Lemmas 1 and 2, we can conclude that $IDEAL_{F,Sim,Z} \overset{\text{comp}}{\approx} REAL_{\pi,A,Z}$. This means that even if the semi-honest adversary A corrupts $N-1$ users and colludes with the server, A still cannot infer the private information of the only honest participant P_h. □

6. Performance Analysis and Experimentation

6.1. Performance Analysis

This section presents an analysis of the performance of the proposed mMFHE protocol and makes some comparisons with existing related schemes.

mMFHE is implemented based on the GSW13 scheme. Since homomorphic NAND is realized through a combination of homomorphic addition and homomorphic multiplication, we can evaluate its efficiency by analyzing the complexity of NAND. In comparison to other FHE schemes, the mMFHE scheme exhibits a time complexity of $\tilde{O}(N(nd)^{\omega})$ for evaluating NAND gates, where n represents the lattice dimension, d denotes the depth of the NAND circuit being evaluated, and $\omega < 2.3727$ is a fixed constant [44]. In parallel research, detailed in reference [45], another LWE-based FHE scheme, referred to as Bv11, serves as the foundation for constructing SMC protocols via threshold decryption. The Bv11 scheme achieves a complexity of $\tilde{O}(n^3 d)$ for evaluating NAND gates. This positions it as marginally less efficient than mMFHE, particularly when the lattice dimension n is large. More importantly, the ciphertexts in mMFHE are in matrix form. This implies a lower expansion rate for ciphertext size, reduced time consumption, and the ability to avoid evaluation key operations in homomorphic evaluation, which are often the most time- and space-consuming aspects of FHE and related applications.

In recent years, several FHE schemes based on GSW13 have emerged, such as in [15,40]. These schemes follow a "matrix cascading" approach to construct joint ciphertexts, resulting in large ciphertext sizes and computational assumptions. Moreover, the aforementioned schemes are single-bit; if a participant's input is t bits, the two schemes need to be executed t times. In contrast, the scheme proposed in this paper only requires a single execution, making it more time-efficient than existing schemes. The details are shown in Table 2, where "CTE Ratio" represents the ciphertext expansion ratio, n is the lattice dimension, N is the number of users, EK indicates whether the key needs to be evaluated, and "NAND TCP" is the time complexity consumed by each NAND gate.

Table 2. Comparison of computational performance of related FHE schemes.

Scheme	Base	CTE Ratio	EK	NAND TCP
[15]	GSW13	$O(1)$	No	$\tilde{O}(tN(nd)^w)$
[46]	NTRU	$O(1)$	Yes	Depend on N
[45]	Bv11	$(n+1)\log q$	Yes	$\tilde{O}(n^3d)$
[40]	GSW13	$O(1)$	No	$\tilde{O}(t(nd)^w)$
Ours	GSW13	$O(1)$	No	$\tilde{O}((nd)^w)$

6.2. Experimentation and Evaluation

In this study, we tested and evaluated the performance of mMFHE- and mMFHE-based PPFL, respectively. In addition, a series of comparative tests were conducted to demonstrate the performance.

We implemented our scheme and tested its performance using the following settings:

Simulation experiment settings: The hardware used was 13th Gen Intel®Core™ i9-13900HX 2.20 GHz with 32 GB of memory. We used a Ubuntu Server 18.04 LTS as the server operating system and Ubuntu 18.04 LTS as the user operating system. We used three datasets: FEMINIST [47], EMINIST [48], and FashionMINIST [49], as shown in Table 3, to train the model. We used a fully connected neural network (CNN) as the model. The input layer consisted of 784 nodes. The output layer used the softmax activation function. The hidden layer had a five-layer structure and used the ReLu activation function. Our model was trained using the Adam optimizer [50] with a learning rate of 0.01 and 20 local epochs in one round of aggregation. The value of the security parameter q was $2^{31} - 1$.

Table 3. Datasets.

Dataset	Input Size	Sample Num	Partion
FEMINIST	28×28	805,263	3500 users
EMINIST	28×28	814,255	Custom
FashionMINIST	28×28	70,000	Custom

To verify the effectiveness of our scheme, we implemented MK-CKKS-based FL, TEE-based FL, and traditional FL to compare them with the mMFHE-based FL to show the performance of our proposed scheme in terms of computational cost, memory overhead, communication cost, and accuracy. Traditional federated learning has no additional privacy protection for model updates. MK-CKKS-based federated learning encrypts model updates through MK-CKKS, but there is a privacy risk due to the use of noise flooding technology during decryption.

Computational cost: We compared mMFHE with MK-CKKS in terms of homomorphic addition, homomorphic multiplication, encryption, and decryption. MK-CKKS is a multi-key version of the MKFHE scheme CKKS. Federated learning based on MK-CKKS is updated through the MK-CKKS encryption model, but there is a privacy risk due to the use of noise flooding technology during decryption. As can be seen from Figure 4, the cost of mMFHE in the encryption and decryption stages is roughly the same as that of MK-CKKS, but it has certain advantages overall. We discuss the computational cost when the number of addition and multiplication operations changes in Figures 5 and 6. We note that the time required for mMFHE to perform homomorphic addition is linearly related to the number of executions, and is better than the MK-CKKS scheme. The multiplication efficiency of mMFHE is slightly lower than that of MK-CKKS at the beginning, but as the number of multiplication operations increases, MK-CKKS needs to perform complex operations such as the rescaling of the noise, which leads to a decrease in multiplication efficiency and is lower than the multiplication execution efficiency of mMFHE.

Memory consumption: Compared with mMFHE and MK-CKKS, TEE-based FL migrates the entire privacy process to secure memory. MK-CKKS and mMFHE use multi-key HE running in ordinary memory to protect privacy. As shown in Figure 7, we tested the

memory overhead of the three. Due to the lower ciphertext expansion rate of mMFHE, the memory overhead of mMFHE is greater than the secure memory of TEE-based FL, but less than the ordinary memory of MK-CKKS. Although the memory overhead of TEE-based FL is smaller than that of mMFHE, it cannot guarantee security in the face of collusion attacks between users and servers. In addition, TEE needs to use an expensive page swap mechanism in secure memory. Therefore, we believe that the memory overhead of mMFHE is acceptable as a price to pay for higher security.

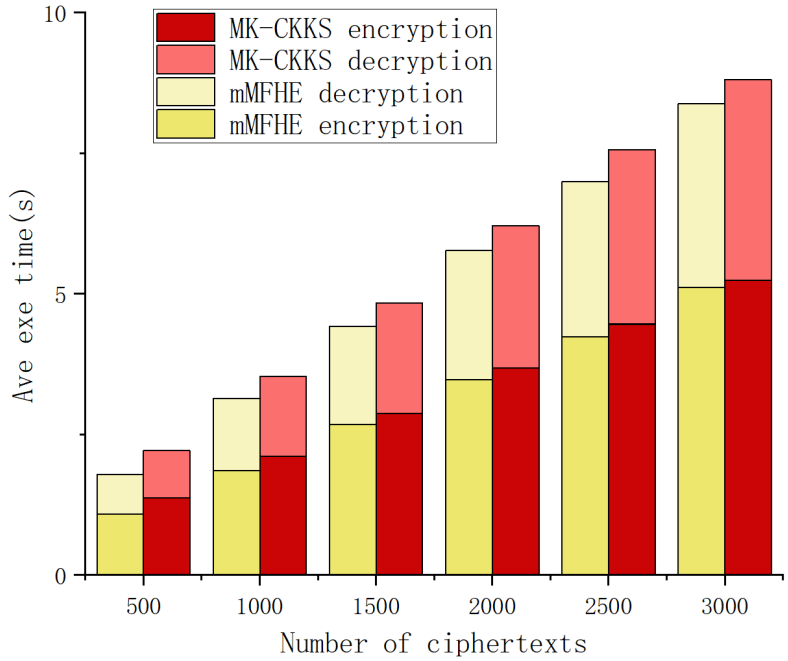


Figure 4. Encryption and decryption.

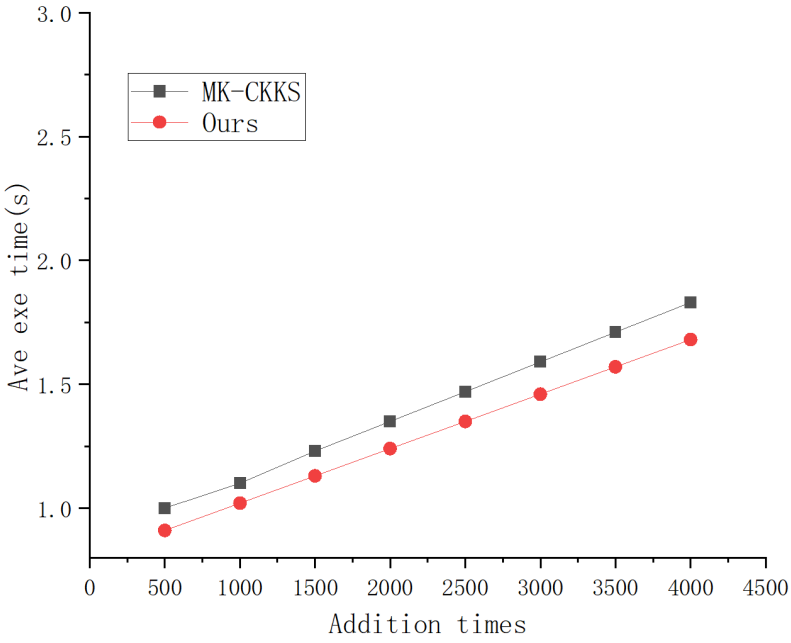


Figure 5. Addition.

Communication overhead: We further evaluated the total communication overhead of a round in mMFHE-based PPFL. The data transmitted from the user to the server were mainly encrypted in the model. We tested the communication cost when the model size is at 10^4 to 10^6. Table 4 lists the ciphertext and communication consumption corresponding

to the number of model parameters, and the communication overhead was about 32.2 MB when the model size reached 7027860, which is within the acceptable range.

Accuracy: Model accuracy usually refers to the proportion of correct predictions made by the global model on the test dataset. We evaluated the model accuracy of mMFHE-based PPFL and compared it with the traditional FL scheme and MK-CKKS-based FL. As shown in Figure 8,when we increase the local epoch to L = 40, the mMFHE-based scheme provides an accuracy of 97.1%, which is very close to the accuracy of the federated learning scheme (97.9%) and higher than that of MK-CKKS. This is because MK-CKKS uses approximate calculations when encrypting and has the problem of error accumulation, which also proves the accurate decryption of the mMFHE scheme.

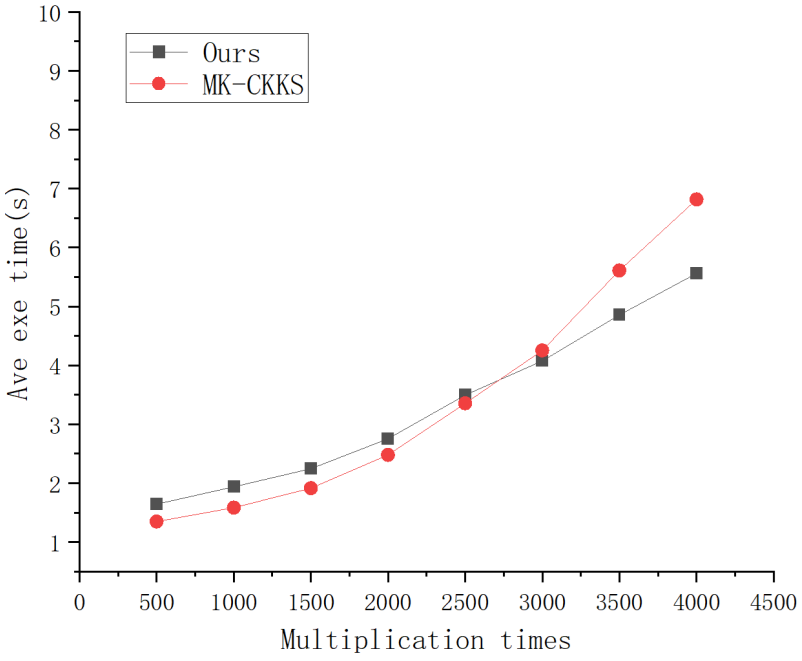


Figure 6. Multiplication.

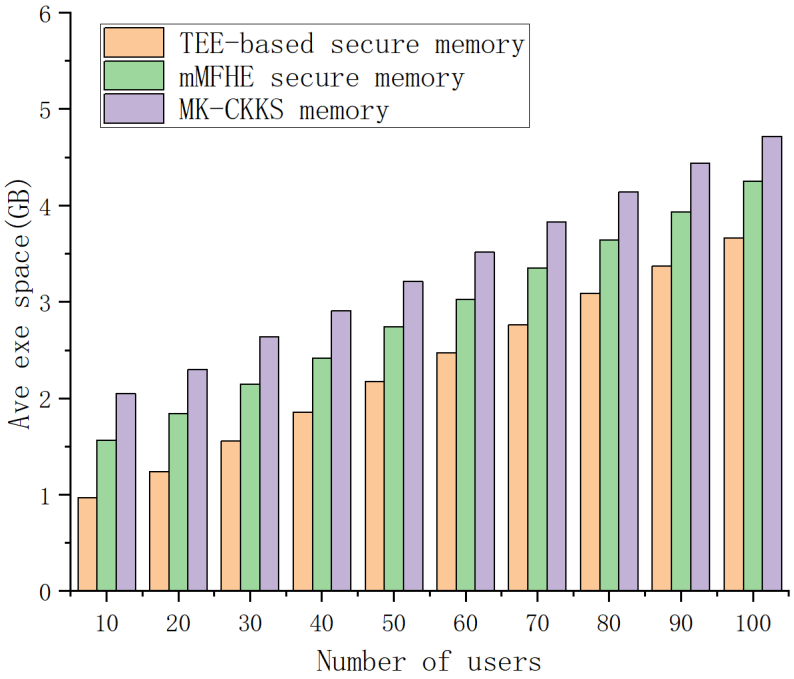


Figure 7. Memory consumption.

Table 4. Communication overhead

Model Size	Ciphertexts Num	Communication Overhead
50,670	63	232 KB
616,420	762	2.8 MB
7,027,860	8660	32.2 MB

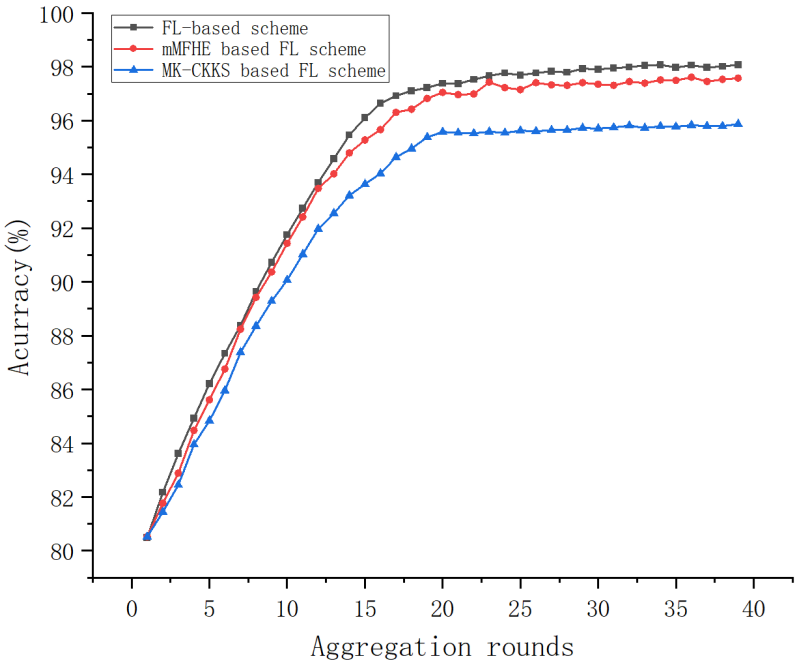


Figure 8. Model accuracy.

7. Discussion

We will discuss some of the current problems faced by mMFHE-based FL with future research directions in this section.

Transmission penalties for encrypted data: In FL based on HE, the encrypted transmission of the data does not directly result in penalties. HE is a form of encryption that allows computations to be performed directly on encrypted data and results in encrypted results, a process that does not require decrypting the data. This provides strong guarantees for protecting data privacy and is particularly suitable for use in scenarios where data privacy is critical, such as in FL. However, the use of HE introduces some other challenges and costs, such as computational efficiency vs. latency. HE operations are typically more complex and time-consuming than non-encryption operations, which can increase the overall computational requirements and latency of the learning task, and we have also shown in our experiments the time spent on the encryption and decryption processes—which is not required in non-encrypted FL. In addition, advanced encryption techniques require more computational resources, and we also show in our experiments the memory consumption required by mMFHE. These factors may impose "penalties" in terms of system performance and cost, especially in resource-limited devices or applications that require real-time response, which also leads to the lack of real-time applicability of mMFHE compared to TEE-based FL.

Reducing complexity: To address the above-mentioned problems of mMFHE in terms of computational efficiency and overhead, we plan to explore the possibility of applying hybrid cryptography in FL in our subsequent research, i.e., combining HE and other types of cryptography (e.g., symmetric encryption or SMPC) to share the computational load. For example, HE could be used for highly computationally demanding parts, while more efficient encryption schemes could be used for other parts. Alternatively, a selective

encryption strategy can be applied to select the appropriate encryption granularity and strength based on the sensitivity of the data and the computational requirements, thus reducing the unnecessary computational burden.

Key management: In MKFHE-based FL, managing multiple keys during multi-party decryption may lead to errors, which remains a challenging issue. We plan to introduce the trusted execution environment as the key management center (KMC) in the subsequent improvement scheme, which is responsible for key generation and management, and select an appropriate key distribution protocol to ensure the security of the key distribution process from the KMC to the user side.

Scalability: In mMFHE-based FL, scalability becomes an important issue as the number of clients increases. The proposed system may face complexity issues as the number of participants increases. To address this issue, we plan to introduce a dynamic participant management mechanism that holds dynamic participant joining and exiting. Through dynamically adjusting the number and load of participants, the computational load and network communication pressure of the system can be effectively balanced. This can be combined with a key management mechanism, where new participants need to go through a secure registration process to obtain the necessary authentication and authorization when joining. In mMFHE, this typically involves distributing or generating a set of keys (public and private) and sharing the public key with the rest of the system. When a participant exits, a logout mechanism is required to ensure that their keys and sensitive data are no longer used by the system and to remove their public key from any shared lists or databases.

Interoperability: Interoperability is a key issue in mMFHE-based FL systems, especially when multiple different organizations or platforms are involved. Addressing interoperability issues mainly involves ensuring effective communication and data sharing between different systems while protecting data privacy and security. In our subsequent work, we plan to design a scalable and flexible system architecture that can accommodate different types of computing nodes and data storage solutions. Such an architecture should support modular and plug-in extensions to facilitate the integration of new technologies or third-party services.

8. Conclusions and Future Work

We propose a PPFL on the basis of MKFHE. Specifically, we designed an MKFHE scheme, mMFHE, based on GSW13, which is secure under the CRS model. To deal with the privacy leakage risks associated with GSW13 as a single-key FHE in FL involving multiple users, we improved it through its key homomorphism, transforming it into MKFHE, which supports encryption using an aggregated public key and shared decryption. Additionally, mMFHE supports embedding multiple plaintext messages into a single ciphertext, thereby enhancing encryption and decryption efficiency while maintaining a low ciphertext expansion rate. Moreover, our security analysis demonstrates that the proposed PPFL scheme can resist collusion attacks involving up to $k = N - 1$ users and the server. Performance analysis and experiments further validate its efficiency.

While our scheme effectively defends against collusion attacks between participants and the server in FL, the introduction of MKFHE also brings new challenges, such as the issue of user offline status during the FL process. If a participant goes offline during model training, the absence of their partial decryption results will cause the aggregation decryption of that round of ciphertext to fail. Given the widespread application of FL in fields such as the Internet of Things (IoT), addressing the user offline issue will be a focus of our future work.

Author Contributions: Conceptualization, Y.Z.; methodology, Y.Z.; validation, J.S. and Y.Z.; resources, Y.R.; writing—original draft preparation, J.S. and Y.Z.; writing—review and editing, Y.Z. and S.H.; visualization, Y.R.; supervision, Y.R.; funding acquisition, Y.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China grant number 62072249.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhu, L.; Liu, Z.; Han, S. Deep leakage from gradients. In Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019) , Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
2. Hitaj, B.; Ateniese, G.; Perez-Cruz, F. Deep models under the GAN: Information leakage from collaborative deep learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 603–618.
3. Vatter, J.; Mayer, R.; Jacobsen, H.A. The evolution of distributed systems for graph neural networks and their origin in graph processing and deep learning: A survey. *ACM Comput. Surv.* **2023**, *56*, 1–37. [CrossRef]
4. McMahan, H.B.; Yu, F.; Richtarik, P.; Suresh, A.; Bacon, D. Federated learning: Strategies for improving communication efficiency. In Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016; pp. 5–10.
5. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, Ft. Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
6. Rieyan, S.A.; News, M.R.K.; Rahman, A.M.; Khan, S.A.; Zaarif, S.T.J.; Alam, M.G.R.; Hassan, M.M.; Ianni, M.; Fortino, G. An advanced data fabric architecture leveraging homomorphic encryption and federated learning. *Inf. Fusion* **2024**, *102*, 102004. [CrossRef]
7. Mantey, E.A.; Zhou, C.; Anajemba, J.H.; Arthur, J.K.; Hamid, Y.; Chowhan, A.; Otuu, O.O. Federated learning approach for secured medical recommendation in internet of medical things using homomorphic encryption. *IEEE J. Biomed. Health Inform.* **2024**, *28*, 3329–3340. [CrossRef] [PubMed]
8. Hou, X.; Wang, J.; Jiang, C.; Meng, Z.; Chen, J.; Ren, Y. Efficient federated learning for metaverse via dynamic user selection, gradient quantization and resource allocation. *IEEE J. Sel. Areas Commun.* **2023**, *42*, 850–866. [CrossRef]
9. Ren, Y.; Lv, Z.; Xiong, N.N.; Wang, J. HCNCT: A cross-chain interaction scheme for the blockchain-based metaverse. *ACM Trans. Multimed. Comput. Commun. Appl.* **2024**, *20*, 1–23. [CrossRef]
10. Issa, W.; Moustafa, N.; Turnbull, B.; Sohrabi, N.; Tari, Z. Blockchain-based federated learning for securing internet of things: A comprehensive survey. *ACM Comput. Surv.* **2023**, *55*, 1–43. [CrossRef]
11. Aono, Y.; Hayashi, T.; Wang, L.; Moriai, S. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Trans. Inf. Forensics Secur.* **2017**, *13*, 1333–1345.
12. Melis, L.; Song, C.; De Cristofaro, E.; Shmatikov, V. Exploiting unintended feature leakage in collaborative learning. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019; pp. 691–706.
13. Sun, L.; Wang, Y.; Ren, Y.; Xia, F. Path signature-based xai-enabled network time series classification. *Sci. China Inf. Sci.* **2024**, *67*, 170305. [CrossRef]
14. Ren, Y.; Zhu, F.; Wang, J.; Sharma, P.K.; Ghosh, U. Novel vote scheme for decision-making feedback based on blockchain in internet of vehicles. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 1639–1648. [CrossRef]
15. Mukherjee, P.; Wichs, D. Two round multiparty computation via multi-key FHE. In *Advances in Cryptology—EUROCRYPT 2016, Proceedings of the 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, 8–12 May 2016*; Proceedings 31; Springer: Berlin/Heidelberg, Germany, 2012; pp. 735–763.
16. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst.* **2020**, *2*, 429–450.
17. Asad, M.; Moustafa, A.; Ito, T. Fedopt: Towards communication efficiency and privacy preservation in federated learning. *Appl. Sci.* **2020**, *10*, 2864. [CrossRef]
18. Zhang, J.; Hua, Y.; Wang, H.; Song, T.; Xue, Z.; Ma, R.; Guan, H. Fedala: Adaptive local aggregation for personalized federated learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37, pp. 11237–11244.
19. Yu, X.; Liu, R.; Nkenyereye, L.; Wang, Z.; Ren, Y. ACRS-Raft: A Raft Consensus Protocol for Adaptive Data Maintenance in the Metaverse Based On Cauchy Reed-Solomon Codes. *IEEE Trans. Consum. Electron.* **2024**, *70*, 3792–3801. [CrossRef]
20. Zhang, C.; Li, S.; Xia, J.; Wang, W.; Yan, F.; Liu, Y. {BatchCrypt}: Efficient homomorphic encryption for {Cross-Silo} federated learning. In Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC 20), Online, 15–17 July 2020; pp. 493–506.
21. Madi, A.; Stan, O.; Mayoue, A.; Grivet-Sébert, A.; Gouy-Pailler, C.; Sirdey, R. A secure federated learning framework using homomorphic encryption and verifiable computing. In Proceedings of the 2021 Reconciling Data Analytics, Automation, Privacy, and Security: A Big Data Challenge (RDAAPS), Hamilton, ON, Canada, 18–19 May 2021; pp. 1–8.

22. Stripelis, D.; Saleem, H.; Ghai, T.; Dhinagar, N.; Gupta, U.; Anastasiou, C.; Ver Steeg, G.; Ravi, S.; Naveed, M.; Thompson, P.M.; et al. Secure neuroimaging analysis using federated learning with homomorphic encryption. In Proceedings of the 17th International Symposium on Medical Information Processing and Analysis, Campinas, Brazil, 17–19 November 2021; Volume 12088, pp. 351–359.
23. Lindell, Y. Secure multiparty computation. *Commun. ACM* **2020**, *64*, 86–96. [CrossRef]
24. Acar, A.; Aksu, H.; Uluagac, A.S.; Conti, M. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Comput. Surv.* **2018**, *51*, 1–35. [CrossRef]
25. Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H.B.; Patel, S.; Ramage, D.; Segal, A.; Seth, K. Practical secure aggregation for privacy-preserving machine learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 1175–1191.
26. Wei, K.; Li, J.; Ding, M.; Ma, C.; Yang, H.H.; Farokhi, F.; Jin, S.; Quek, T.Q.; Poor, H.V. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3454–3469. [CrossRef]
27. Truex, S.; Liu, L.; Chow, K.H.; Gursoy, M.E.; Wei, W. LDP-Fed: Federated learning with local differential privacy. In Proceedings of the third ACM International Workshop on Edge Systems, Analytics and Networking, Heraklion, Greece, 27 April 2020; pp. 61–66.
28. Hu, R.; Guo, Y.; Li, H.; Pei, Q.; Gong, Y. Personalized federated learning with differential privacy. *IEEE Internet Things J.* **2020**, *7*, 9530–9539. [CrossRef]
29. Li, Y.; Zhou, Y.; Jolfaei, A.; Yu, D.; Xu, G.; Zheng, X. Privacy-preserving federated learning framework based on chained secure multiparty computing. *IEEE Internet Things J.* **2020**, *8*, 6178–6186. [CrossRef]
30. Gehlhar, T.; Marx, F.; Schneider, T.; Suresh, A.; Wehrle, T.; Yalame, H. SafeFL: MPC-friendly framework for private and robust federated learning. In Proceedings of the 2023 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 25 May 2023; pp. 69–76.
31. Zhang, J.; Chen, B.; Yu, S.; Deng, H. PEFL: A privacy-enhanced federated learning scheme for big data analytics. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6.
32. Li, Y.; Li, H.; Xu, G.; Huang, X.; Lu, R. Efficient privacy-preserving federated learning with unreliable users. *IEEE Internet Things J.* **2021**, *9*, 11590–11603. [CrossRef]
33. Ren, Y.; Leng, Y.; Qi, J.; Sharma, P.K.; Wang, J.; Almakhadmeh, Z.; Tolba, A. Multiple cloud storage mechanism based on blockchain in smart homes. *Future Gener. Comput. Syst.* **2021**, *115*, 304–313. [CrossRef]
34. He, C.; Liu, G.; Guo, S.; Yang, Y. Privacy-preserving and low-latency federated learning in edge computing. *IEEE Internet Things J.* **2022**, *9*, 20149–20159. [CrossRef]
35. Ren, Y.; Leng, Y.; Cheng, Y.; Wang, J. Secure data storage based on blockchain and coding in edge computing. *Math. Biosci. Eng.* **2019**, *16*, 1874–1892. [CrossRef] [PubMed]
36. Cai, Y.; Ding, W.; Xiao, Y.; Yan, Z.; Liu, X.; Wan, Z. SecFed: A Secure and Efficient Federated Learning Based on Multi-Key Homomorphic Encryption. *IEEE Trans. Dependable Secur. Comput.* **2023**, *21*, 3817–3833. [CrossRef]
37. Ma, J.; Naas, S.A.; Sigg, S.; Lyu, X. Privacy-preserving federated learning based on multi-key homomorphic encryption. *Int. J. Intell. Syst.* **2022**, *37*, 5880–5901. [CrossRef]
38. Walskaar, I.; Tran, M.C.; Catak, F.O. A practical implementation of medical privacy-preserving federated learning using multi-key homomorphic encryption and flower framework. *Cryptography* **2023**, *7*, 48. [CrossRef]
39. Zhang, Q.; Jing, S.; Zhao, C.; Zhang, B.; Chen, Z. Efficient federated learning framework based on multi-key homomorphic encryption. In *Advances on P2P, Parallel, Grid, Cloud and Internet Computing, Proceedings of the 16th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC-2021), Fukuoka, Japan, 28–30 October 2021*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 88–105.
40. Wang, H.; Feng, Y.; Ding, Y.; Tang, S. A multi-key SMC protocol and multi-key FHE based on some-are-errorless LWE. *Soft Comput.* **2019**, *23*, 1735–1744. [CrossRef]
41. Gentry, C.; Sahai, A.; Waters, B. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology–CRYPTO 2013, Proceedings of the 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, 18–22 August 2013*; Proceedings, Part I; Springer: Berlin/Heidelberg, Germany, 2013; pp. 75–92.
42. Li, Z.; Ma, C.; Zhou, H. Multi-key FHE for multi-bit messages. *Sci. China Inf. Sci.* **2018**, *61*, 029101. [CrossRef]
43. Li, Z.; Ma, C.; Morais, E.; Du, G. Multi-bit Leveled Homomorphic Encryption via-Based. In Proceedings of the International Conference on Information Security and Cryptology, Beijing, China, 4–6 November 2016; pp. 221–242.
44. Sun, L.; Li, C.; Ren, Y.; Zhang, Y. A Multitask Dynamic Graph Attention Autoencoder for Imbalanced Multilabel Time Series Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 11829–11842. [CrossRef]
45. Asharov, G.; Jain, A.; López-Alt, A.; Tromer, E.; Vaikuntanathan, V.; Wichs, D. Multiparty computation with low communication, computation and interaction via threshold FHE. In *Advances in Cryptology–EUROCRYPT 2012, Proceedings of the 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, 15–19 April 2012*; Proceedings 31; Springer: Berlin/Heidelberg, Germany, 2012; pp. 483–501.
46. López-Alt, A.; Tromer, E.; Vaikuntanathan, V. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing, New York, NY, USA, 19–22 May 2012; pp. 1219–1234.

47. Caldas, S.; Duddu, S.M.K.; Wu, P.; Li, T.; Konečnỳ, J.; McMahan, H.B.; Smith, V.; Talwalkar, A. Leaf: A benchmark for federated settings. *arXiv* **2018**, arXiv:1812.01097.
48. Cohen, G.; Afshar, S.; Tapson, J.; Van Schaik, A. EMNIST: Extending MNIST to handwritten letters. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 2921–2926.
49. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv* **2017**, arXiv:1708.07747.
50. Kingma, D.P. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.

Homomorphic Encryption-Based Privacy-Preserving Federated Learning in IoT-Enabled Healthcare System

Li Zhang, Jianbo Xu, Pandi Vijayakumar, *Senior Member, IEEE*, Pradip Kumar Sharma, *Senior Member, IEEE*, and Uttam Ghosh, *Senior Member, IEEE*

***Abstract*—In this work, the federated learning mechanism is introduced into the deep learning of medical models in Internet of Things (IoT)-based healthcare system. Cryptographic primitives, including masks and homomorphic encryption, are applied for further protecting local models, so as to prevent the adversary from inferring private medical data by various attacks such as model reconstruction attack or model inversion attack, etc. The qualities of the datasets owned by different participants are considered as the main factor for measuring the contribution rate of the local model to the global model in each training epoch, instead of the size of datasets commonly used in deep learning. A dropout-tolerable scheme is proposed in which the process of federated learning would not be terminated if the number of online clients is not less than a preset threshold. Through the analysis of the security, it shows that the proposed scheme satisfies data privacy. Computation cost and communication cost are also analyzed theoretically. Finally, skin lesion classification using training images provided by the HAM10000 medical dataset is set as an example of healthcare applications. Experimental results show that compared with existing schemes, the proposed scheme obtained promising results while ensuring privacy preserving.**

***Index Terms*—Federated learning, homomorphic encryption, privacy-preserving, convolutional neural networks, IoT-enabled healthcare system.**

I. Introduction

THE Internet of Things (IoT) is a new paradigm with a wide range of applications. IoT-connected healthcare applications provide real-time monitoring and smart medical IoT devices that are synchronized to a smartphone app, allowing doctors to obtain medical data from their patients at any time or location. It also provides computer-assisted diagnostics, medical image analysis, and remote medical support, etc. It enables medical centers to operate more efficiently, and enables patients to receive better care. There are certain advantages to utilizing IoT-based healthcare methods, which might enhance the treatment quality and efficiency, and therefore the health of patients [1].

The development of Big Data and artificial intelligence has made it convenient to carry out joint researches in various fields. Especially in the medical field, the sharing of electronic health data benefits doctors to predict some diseases and formulate related treatment plans [2]. To some extent, data sharing promotes the development of the entire medical industry. However, the sensitivity of medical data has become an obstacle to resource sharing. Most of the medical institutions are located in disperse geographical locations and abide by different administrative management. Therefore, they are reluctant to share patient medical data by taking the risk of violating privacy ethics or even losing the economic benefits, such as leaking information about AIDS patients, publishing ingredients of patented therapeutic drugs, etc. In other fields, privacy issues are also urgent problems needed to be resolved.

Thanks to paradigm-shifting advancements in machine learning, computer-aided diagnostic techniques have already reached unprecedented levels. In this context, as a common type of cancer that originates in the epidermal layer when abnormal cells are exposed to ultraviolet radiation, skin cancer has gotten significant attention, and deep learning algorithms have attained a level of precision that is equivalent to that of trained dermatologists [1], [3].

For IoT-enabled healthcare system, in pursuing the huge benefits data sharing brings but unwilling to violate privacy, scholars have proposed a federated learning framework, which allows data to contribute to federated machine learning despite being kept in local repositories [4], [5]. Federated learning provides broad prospects for the application of artificial intelligence in different industries. However, in terms of privacy preservation, federated learning is not foolproof. There are still some challenges in applying federated learning to real application scenarios, such as IoT-based healthcare systems [6]–[8]. Although local data is not directly shared, models trained on these data may also be snooped by malicious adversaries or

Manuscript received 9 October 2021; revised 3 May 2022; accepted 18 June 2022. Date of publication 30 June 2022; date of current version 20 September 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 61872138, and in part by the Natural Science Foundation of Hunan Province under Grant 2021JJ30278. Recommended for acceptance by Dr. Varun G Menon. *(Corresponding authors: Jianbo Xu; Pandi Vijayakumar.)*

Li Zhang and Jianbo Xu are with the School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China, and also with Hunan Key Laboratory for Service Computing and Novel Software Technology, Hunan University of Science and Technology, Xiangtan 411201, China (e-mail: lzhang@mail.hnust.edu.cn; jbxu@hnust.edu.cn).

Pandi Vijayakumar is with the Department of Computer Science and Engineering, University College of Engineering Tindivanam, Tindivanam, Tamilnadu 604001, India (e-mail: vijibond2000@gmail.com).

Pradip Kumar Sharma is with the Department of Computing Science, University of Aberdeen, Aberdeen AB243UE, U.K. (e-mail: pradip.sharma@abdn.ac.uk).

Uttam Ghosh is with Computer Science and Data Science, Meharry Medical College, Nashville, TN 37208 USA (e-mail: ghosh.uttam@ieee).

Digital Object Identifier 10.1109/TNSE.2022.3185327

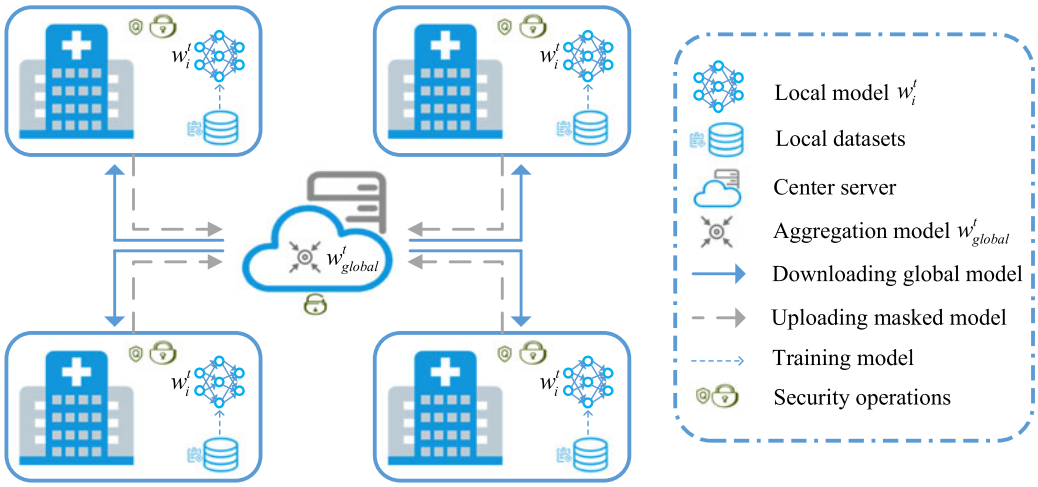


Fig. 1. The system overview for federated learning on medical data.

honest but curious parties when local models are aggregated into a center. Moreover, under the circumstance of knowing the local model, snoopers may adopt some attacks to restore the original data, which indirectly leads to information leakage. Therefore, the federated learning mechanism with privacy preservation has attracted more and more attentions. In addition, there are several technical issues in federated learning application, such as how to deal with client dropouts, and how to ensure the accurate global model and low computation and communication overhead, etc.

Although some privacy preservation schemes for federated learning have been proposed in recent years, such as [9]–[11]. Most of them have some limitations more or less. For example, both [10] and [11] proposed homomorphic encryption for federated learning. However, both of them could not solve the problem of dropout clients. In [9], Bonawitz *et al.* proposed a dropout-tolerable scheme, and the masked parameters are transferred, but there is no effective method for improving the accuracy of the model. Therefore, from the perspective of the sensitivity of medical data, the paper proposes a privacy preservation scheme for the federated learning of multiple medical institutions in IoT healthcare applications, as can be seen in Fig. 1. The scheme introduces cryptographic primitives such as homomorphic encryption and secure multi-party computing into the framework of federated learning, so that the privacy of data can still be ensured even if there are collusions among honest but curious participants. Furthermore, in order to alleviate the cost problems that ubiquitously existed in traditional cryptography-based schemes, the paper proposed an improved scheme according to an existing privacy preservation algorithm. Simultaneously, the heterogeneous characteristics of data in different medical institutions are considered, and the model accuracy and training efficiency are promoted.

Specifically, the new algorithm compensates the lack of consideration about data quality in traditional model aggregation algorithms, which makes high-quality local models have a higher contribution rate in the global model aggregation, and reduces the negative impact caused by the heterogeneity of data on the convergence rate and accuracy of the global model. Different from Bonawitz *et al.*'s scheme [9], homomorphic encryption is utilized to calculate the sum of client's data quality. An appropriate adjustment to the ElGamal encryption algorithm is made so as to change the characteristics of the algorithm from multiplicative homomorphism to additive homomorphism. Therefore, communication overhead is reduced. Moreover, unlike traditional encryption in federated learning, the homomorphic encryption scheme does not encrypt every model parameter. Because the model in deep learning usually has high dimensions, and the homomorphic encryption on such high-dimensional data would bring huge computation overhead. There is only a variable called data quality to be encrypted for each client in each training epoch in our scheme. Therefore, the scheme proposed in the paper would not cause the computation overhead to increase sharply.

The main contributions of this paper are summarized as follows:

(1) A novel masking scheme based on homomorphic encryption and the secure multi-party computation have been proposed for federated learning, which utilizes a weighted average algorithm based on data quality to replace the traditional weight calculation method based on the amount of data.

(2) A dropout-tolerable and participants collusion-resistible solution has been proposed in our scheme by employing Diffie-Hellman key exchange and Shamir secret sharing algorithm.

(3) A federated learning prototype system for medical data has been implemented, and extensive experiments using real skin cancer datasets have been conducted to validate the privacy preserving and efficiency of the proposed federated learning scheme.

The organization of this paper is as follows: the related work about federated learning is surveyed in Section II. The basic principles of deep learning, federated learning, and cryptographic primitives are introduced in Section III. The privacy preservation scheme proposed in the paper is described in detail in Section IV. Security analysis on the scheme is conducted in Section V. Theoretical computation cost and communication cost are analyzed in Section VI. Substantial experiments are

performed for illustrating efficiency and accuracy in Section VII. Conclusion and future work are summarized in the final section.

II. Related Work

In recent years, although Big Data, machine learning, artificial intelligence and other technologies have been developed rapidly, the privacy leakage brought by the traditional direct data sharing mechanism has also increased seriously [12]–[15]. Privacy problem has led to the emergence of isolated data islands, which severely hindered the development of artificial intelligence. In order to solve the privacy problem, Google initially proposed the concept of federated learning in 2016 for the update of the mobile terminal user's local model. They used a federated average (FedAvg) technology to promote high-efficient machine learning among multiple nodes and guarantee privacy security for all data in personal smartphones [4], [5]. Since then, some open source projects for federated learning have flourished. The most representative projects are respectively the FATE project[1] developed by WeBank and the TensorFlow framework developed by Google[2] which provide different support strategies for privacy preservation in machine learning.

Although federated learning has been continuously developed, various attack models have also emerged. Bhowmick *et al.* [16] elaborated on reconstruction attack and countermeasures. Nasr *et al.* [17] and Shokri *et al.* [18] designed member inference attacks with the white-box model and black-box model, respectively. Fredrikson *et al.* [19] found that given a model and some statistical information, the dose guidance model in pharmacogenetics is vulnerable to model inversion attack. Melis *et al.* [20] explained how to implement attribute inference attack. Especially, Zhu *et al.* [21] proposed a depth gradient leakage scheme, in which the adversaries unknew any other information except the local model, they still can restore image with the high similarity to the original sample by training the dummy sample. Therefore, it can be seen that even if the original data is kept locally in federated learning, the update of the local model still gives the adversaries some opportunities to attack.

With the appearance of various attack models that violate the privacy and confidentiality of machine learning, miscellaneous countermeasures have been proposed. Zhang *et al.* [22] and Hardy *et al.* [23] applied additive homomorphic encryption algorithm to prevent the local model from being snooped by honest but curious participants in model aggregation process. Although homomorphic encryption provides the strongest guarantee for privacy preservation, it introduces a large amount of computational overhead. In federated deep learning with a high-dimensional model, homomorphic encryption has an obvious impact on performance. It has become an important reason why homomorphic encryption is difficult to be put into practice. Aimed at saving computational overhead, some homomorphic encryption schemes with gradient sparseness and gradient quantization have been proposed [24], [25].

In addition to encryption technology, secure multi-party computation is also an important realization method for privacy preservation. Xu *et al.* [26] utilized oblivious transfer and garbled circuits in secure multi-party computation to implement homomorphic multiplication and division operations between two servers, and then completed secure model aggregation. In 2017, Bonawitz *et al.* [9] designed a secure double-masking aggregation scheme by using many technologies such as the Diffie-Herman key agreement, t-out-of-n Shamir secret sharing, pseudo-random generator, the public key infrastructure, authentication, and signatures, etc. Although the scheme permitted the aggregation to be successful when some participants dropped out, frequent requests were needed for unmasking in each training epoch, which caused a huge communication overhead. In order to decrease the cost, Choi *et al.* [27] proposed an improved scheme with a new secret sharing topology, in which the secret shares were only shared with their neighboring nodes instead of being shared with all nodes. Thereby communication overhead was reduced.

Another privacy preservation mechanism is differential privacy, which has received wide attention from scholars due to its significant reduction in computing costs compared with the previous two technologies. Differential privacy was first proposed by Dwork in 2006 [28]. Later, it was introduced into federated machine learning to ensure that sensitive information is not leaked. Nowadays, many scholars have carried out massive researches on the core issue involved in differential privacy technology, namely, the trade-off between noise intensity and model accuracy [29], [30].

Federated learning also provides an effective way to train medical models without sharing the patient's electronic health records. Undoubtedly, the entire medical community would benefit from disease prediction and treatment. At present, many literatures are focusing on the federated learning of medical data. Roy *et al.* [31] proposed a decentralized federated learning framework for brain segmentation in MRI T1 scans. Brisimi *et al.* [32] also solved a binary supervised classification problem to predict the hospitalization time for patients with heart disease. Although the data was kept locally, the privacy leakage perhaps caused by the model updates was still non-negligible. However, both of the two schemes did not consider the solutions. Lee *et al.* [33] proposed a multi-institutional federated training algorithm for the model of patient similarity that uses homomorphic encryption technology to ensure the privacy of patients. Choudhury *et al.* [34] applied differential privacy technology in the federated learning of medical data. Rieke *et al.* [35] discussed the vital role of federated learning in the development of digital healthcare and emphasized the main challenges currently faced. Finally, they looked forward to the prospects of digital healthcare. In the paper, we will dedicate to privacy preservation in federated learning for medical data.

The existing privacy preservation schemes for federated learning have some shortcomings more or less, and cannot perfectly solve all the problems in one scheme. For example, Asad *et al.*

[1] https://github.com/FederatedAI/FATE
[2] https://www.tensorflow.org/federated

proposed a federated learning scheme named FedOpt, which designed a sparse compression algorithm for efficient communication and also integrated the homomorphic encryption [10]. In [11], Fang *et al.* proposed a federated learning scheme based on homomorphic encryption for all parameters. However, both of them could not solve the problem of dropout clients. In this paper, motivated by Bonawitz *et al.*'s work [9], we try to learn the advantages of the existing schemes, and rely on the combination of various cryptographic primitives to support the privacy preservation for federated learning and also solve the problem of client dropout, ensuring the accuracy of the model at the same time, which is different from existing schemes.

III. Preliminary

A. Federated Deep Learning

Deep machine learning commonly refers to machine learning based on neural networks such as Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) [36], [37], etc. The network model is generally divided into three layers, including input layer, hidden layer and output layer, and there are numerous neurons in each layer. Deep learning aims at training connection weight between two neurons in neighbor layers, namely network model, to the output of the model, which has a minimal error with the original label.

Assuming that $\mathcal{DS}$ is a dataset with m samples, denoted as $\mathcal{DS} = \{(\mathbf{x}_k, \mathbf{y}_k), k = 1, 2, \ldots, m\}$, where $\mathbf{x}_k$ is the feature vector of the k-th sample and $\mathbf{y}_k$ is the label. Given an initial weight vector $\mathbf{w}$, we firstly compute the output function $f(\mathbf{x}_k, \mathbf{w})$ of the neural network, and then calculate the error between the output function and the label. The error is defined as the loss function $\mathcal{L}$ which is described as formula (1).

$$\mathcal{L}_f(\mathcal{DS}, \mathbf{w}) = \frac{1}{|\mathcal{DS}|} \sum\nolimits_{<\mathbf{x}_k, \mathbf{y}_k> \in \mathcal{DS}} \left\| f(\mathbf{x}_k, \mathbf{w}) - \mathbf{y}_k \right\|_2 \quad (1)$$

where, $|\mathcal{DS}|$ and $\| \bullet \|_2$ represents the size of the dataset and the L_2-norm of a vector, respectively. In a neural network, the calculation process from output function to loss function is called as forward propagation. In order to minimize the loss function, the model parameters need to be adjusted continuously. This process is called as backward propagation, in which the stochastic gradient descent (SGD) algorithm is usually adopted for finding out the optimal model parameters. The specific model update in each iteration is shown in formula (2).

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \beta \cdot \nabla \mathcal{L}_f(\mathcal{DS}, \mathbf{w}^t) \quad (2)$$

Here, t is the number of iterations, and β is the learning rate which denotes the step length of the model adjustment in each iteration. $\nabla \mathcal{L}_f(\mathcal{DS}, \mathbf{w}^t)$ is model gradient which means the partial derivative of the loss function in each model dimension. Given a d-dimension model $\mathbf{w} = < w_1, w_2, \ldots, w_d >^T$, the gradient can be calculated according to formula (3).

$$\nabla \mathcal{L}_f(\mathcal{DS}, \mathbf{w}^t) = < \frac{\partial \mathcal{L}}{\partial w_1}, \frac{\partial \mathcal{L}}{\partial w_2}, \ldots, \frac{\partial \mathcal{L}}{\partial w_d} >^T \quad (3)$$

For massive medical images, CNN is the preferred choice for model training. First, the image is sampled and compressed in the convolutional layer and pooling layer, so as to achieve the purpose of reducing the number of model parameters, saving computational complexity and preventing overfitting. After that, the flatten layer is connected to the full connected layer in the traditional neural network for subsequent training.

In federated learning, assuming that there are N clients to collaborate for learning, we denoted them as $\{\mathcal{P}_i\}$, $i \in \{1, 2, \ldots, N\}$. Each client trains the local model w_i on its own dataset $\mathcal{DS}_i$ through performing several iterations with SGD. Then, the local model is uploaded to the server. The server aggregates all of the local models submitted by N clients into a global model using an arithmetic average algorithm or weighted average algorithm [38], [39]. Given the aggregation weight α_i for the client $\mathcal{P}_i$, the global model $\mathbf{w}_{global}$ can be generated according to formula (4), and then distributed to each client.

$$\mathbf{w}_{global} = \sum_{i=1}^{N} \alpha_i \cdot w_i \quad (4)$$

After that, every client continues to perform iterative training until the model converges or termination condition is reached. In the entire process, the original dataset located in each client is not shared with the server, instead of submitting the model parameters, which preserves the data's privacy to a certain extent.

B. Cryptographic Primitives

1) Shamir Secret Sharing: As the most classic e-out-of-n algorithm, Shamir(e, n) secret sharing [40] has been widely applied in cryptography. The main idea of the algorithm is to divide a secret s into n shares, which are distributed to n different parties. Only if no less than e parties contribute their shares, the secret s can be reconstructed. Otherwise, no parties can maliciously leak any information about s even collusion.

Shamir's secret sharing is composed of two algorithms, namely, secret shares generation and secret reconstruction. In cryptography, the secret s is usually a number sampled on a finite field Z_q^* with a large prime order q. Assuming that $\mathcal{P}$ is the set of parties allocated the secret shares and $\mathcal{V}$ is the set participating in the secret reconstruction, satisfying $e \leq |\mathcal{V}| \leq |\mathcal{P}|$. The secret shares generation algorithm is defined as $\mathrm{SSS.share}(s, e, \mathcal{P}) \rightarrow \{(i, s_i)\}_{i \in \mathcal{P}}$, which takes the secret s and the threshold e as input parameters, and then generates $|\mathcal{P}|$ numbers of secret shares $\{(i, s_i)\}_{i \in \mathcal{P}}$ for the corresponding parties in set $\mathcal{P}$. The secret reconstruction algorithm is shown as $\mathrm{SSS.recons}(e, \{(i, s_i)\}_{i \in \mathcal{V}}) \rightarrow s$, in which the constructor needs to acquire e shares (i, s_i) from set $\mathcal{V}$ for restoring the secret s. If $|\mathcal{V}| \leq e$, for two randomly secrets $\forall s_1, s_2 \in Z_q^*$, the security of the algorithm can be guaranteed only when the following requirements are satisfied:

$$\begin{aligned} &\{\{(i, s_i)\}_{i \in \wp} \leftarrow \mathrm{SSS.share}(s_1, e, \mathcal{P}) : \{(i, s_i)\}_{i \in \mathcal{V}}\} \\ \equiv &\{\{(i, s_i)\}_{i \in \mathcal{P}} \leftarrow \mathrm{SSS.share}(s_2, e, \mathcal{P}) : \{(i, s_i)\}_{i \in \mathcal{V}}\} \end{aligned}$$

Here, the symbol "≡" represents computational indistinguishability. In order to satisfy the security requirements, the executor of the algorithm $\mathrm{SSS.share}(s, e, \mathcal{P})$ needs to construct a polynomial with degree (e-1) shown as formula (5).

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_{e-1} x^{e-1} \quad (5)$$

Let's set $a_0 = s$, other coefficients a_1, a_2, a_3, $\cdots$, a_{e-1} are given random values, and then $|\mathcal{P}|$ different random numbers $\{(x_i)\}_{i\in\mathcal{P}}$ are sampled in Z_q^*. After that, the executor calculates $f(x_i)$ and distributes the secret shares $\{(x_i, f(x_i))\}_{i\in\mathcal{P}}$ to the corresponding parties in $\mathcal{P}$. When $|\mathcal{V}|$ secret shares are collected, the constructor selects e shares arbitrarily, and then reconstructs the secret s according to the Lagrange interpolation polynomial shown as formula (6).

$$f(x) = \sum_{i=1}^{e} \prod_{1\le j\le e, j\neq i} \frac{x - x_j}{x_i - x_j} f(x_i) \quad (6)$$

By formula (5), we know $f(0) = a_0 = s$. So, let $x = 0$ in formula (6), then the algorithm $\mathrm{SSS.recons}(e, \{(i, s_i)\}_{i\in\mathcal{V}})$ can be implemented for restoring s, which is shown as formula (7).

$$s = \sum_{i=1}^{e} \prod_{1\le j\le e, j\neq i} \frac{-x_j}{x_i - x_j} f(x_i) \quad (7)$$

2) Diffie-Hellman Key Agreement: The Diffie-Hellman key agreement protocol [41] allows two parties to obtain the symmetric key without exchanging any secret information. The protocol mainly includes three phases: initialization, generation of the public key and the private key, as well as key agreement. First, the key generation center (KGC) selects a security parameter k as input of the initialization function, and outputs a cyclic group G with a generator g and a large prime order q. Then it sets (q, g, G) as public parameters. Subsequently, KGC randomly selects a number $s \in Z_q^*$ as private key s_i^{sk} and calculates $g_s \in G$ as public key s_i^{pk} for each client i, then sends (s_i^{sk}, s_i^{pk}) to the corresponding client through a secure channel. After receiving the pairwise keys, client i broadcasts s_i^{pk} to others. In the phase of key agreement, assuming that there are two clients (denoted as a and b) to execute key agreement, client a calculates $s_{a,b} = (s_b^{pk})^{s_a^{sk}} = (g^{s_b})^{s_a}$ with client b's public key s_b^{pk} and its own private key s_a^{sk}. Similarly, client b computes $s_{b,a} = (s_a^{pk})^{s_b^{sk}} = (g^{s_a})^{s_b}$ using client a's public key s_a^{pk} and its own private key s_b^{sk}. Finally, both of two clients acquire the same key $s_{a,b} = s_{b,a}$.

Definition (Decisional Diffie-Hellman assumption). Assuming that $\mathcal{O}(k) \to (q, g, G)$ is an algorithm about security parameter generation, a probabilistic polynomial time (PPT) adversary $\mathcal{A}$ executes the following random oracle experiment $\mathrm{DDH} - \mathrm{Exp}_{\mathcal{O},\mathcal{A}}^{c}(k)$ which is parameterized by a group of public secure elements (q, g, G) and a bit c.

$\mathrm{DDH} - \mathrm{Exp}_{\mathcal{O},\mathcal{A}}^{c}(k)$:

1) $(q, g, G) \leftarrow \mathcal{O}(k)$;
2) $a \leftarrow Z_q^*, A \leftarrow g^a$;
3) $b \leftarrow Z_q^*, B \leftarrow g^b$;
4) if $c = 1$, $s \leftarrow g^{ab}$, else $s \leftarrow (\forall s' \in G \text{ and } s' \neq g^{ab})$;
5) $(q, g, G, A, B, s) \to c'$;
6) if $c' = c$, $output = 1$, otherwise $output = 0$.

In the experiment, the advantage that the adversary can break the semantic security of the Diffie-Hellman protocol is defined as formula (8).

$$Adv_{\mathcal{O},\mathcal{A}}^{\mathrm{DDH}}(k) = \left| Prob(DDH - Exp_{\mathcal{O},\mathcal{A}}^{c}(k) = 1) - \frac{1}{2} \right| \quad (8)$$

Here, $Prob(\bullet)$ represents a probability function. Suppose there exists a negligible function $\varepsilon(k)$ which satisfies that $Adv_{\mathcal{O},\mathcal{A}}^{\mathrm{DDH}}(k) \le \varepsilon(k)$. In that case, it concludes that no PPT adversary $\mathcal{A}$ can solve the Decisional Diffie-Hellman hard problem, so the key agreement protocol is secure in semantics.

3) Homomorphic Encryption: Homomorphic encryption is a special encryption algorithm [42]. The result obtained by operating on multiple ciphertexts is equivalent to the effect generated by directly operating on corresponding plaintexts.

Assuming that $\mathcal{M}$ and $\mathcal{C}$ respectively means the plaintext space and the ciphertext space, the homomorphic encryption algorithm can be defined in a cryptographic way as expression (9).

$$\begin{aligned} \forall m_a, m_b \in \mathcal{M}, Enc_{pk}(m_a) \odot_{\mathcal{C}} Enc_{pk}(m_b) \\ = Enc_{pk}(m_a \odot_{\mathcal{M}} m_b) \end{aligned} \quad (9)$$

Here, $\odot_{\mathcal{C}}$ and $\odot_{\mathcal{M}}$ represent the operators on the ciphertext space and the plaintext space, respectively. According to the symbols $\odot_{\mathcal{M}}$, homomorphic encryption is divided into additive homomorphism and multiplicative homomorphism. In short, if $\odot_{\mathcal{M}}$ is operator "+," the algorithm is called additive homomorphism; if $\odot_{\mathcal{M}}$ is operator "*," it is called multiplicative homomorphism.

ElGamal algorithm is classified as an asymmetric cryptosystem based on Diffie-Hellman key agreement, which was proposed by Tather ElGamal in 1985 [43]. It has properties of multiplicative homomorphism. The algorithm is mainly composed of four steps.

- Initialization: KGC generates the public parameters (q, g, G) with a secure parameter k, where G is a cyclic group with a large prime q and generator g.
- Key generation: KGC randomly selects a number $\mu \in Z_q^*$ as private key, calculates $y = g^\mu \in G$ as public key, where the public key y is used for encryption, and the private key μ is for decryption.
- Encryption: For encrypting the plaintext m, the message sender selects a random number $r \in Z_q^*$ and calculates $c_1 = g^r$, $c_2 = my^r$, then sends the ciphertext (c_1, c_2) to the receiver.
- Decryption: After receiving the ciphertext (c_1, c_2), the receiver decrypts the plaintext m by computing $m = c_2/c_1^\mu = my^r/(g^r)^\mu$ with private key μ.

According to the basic principle of the ElGamal algorithm, it is only appropriate for multiplicative homomorphism. However, it usually acquires to implement additive aggregation of local models with ciphertext form in the federated learning system. Therefore, it is necessary to take a slight modification

on the ElGamal algorithm to achieve additive homomorphism. In fact, it just needs to transform the plaintext m into an exponential form with integer 2 as the base, i.e., $c_2 = 2^m y^r$. From formula (10), it can be seen that the modified ElGamal algorithm satisfies additive homomorphism. After aggregation with ciphertext form, the sum of plaintexts can be restored by decryption formula (11).

$$\begin{aligned} \forall m_a, m_b \in \mathcal{M}, Enc(m_a) \times Enc(m_b) \\ &= (c_{1a} \cdot c_{1a}, c_{2a} \cdot c_{2a}) \\ &= (g^{r_a+r_b}, 2^{m_a+m_b} y^{r_a+r_b}) \\ &= Enc(m_a+m_b) \end{aligned} \tag{10}$$

$$\begin{aligned} m_a + m_b &= \log_2[(c_{2a} \cdot c_{2a})/(c_{1a} \cdot c_{1a})^{\mu}] \\ &= \log_2[2^{m_a+m_b} y^{r_a+r_b}/(g^{r_a+r_b})^{\mu}] \end{aligned} \tag{11}$$

It is noted that the premise of homomorphic encryption is that different plaintexts must be encrypted with the same public key. In federated learning system, if all clients encrypt their local parameters with the server's key, the server can decrypt all models with the corresponding private key, which means homomorphic encryption is meaningless. Thus, for privacy preservation in federated learning, each client needs to combine other secure measures with the ElGamal algorithm to ensure security, instead of directly encrypting the local parameters using the server's public key.

IV. Proposed Scheme

A. System Model and Security Requirements

The federated learning system in the medical environment is described in this section. It is mainly composed of two types of participants, one is a model aggregation server, and the other is distributed clients. The specific architecture is shown in Fig. 1. The server is responsible for collecting the masked local models submitted by clients, and then performing a series of operations to complete the secure aggregation of the models. The clients are some medical institutions with a large amount of raw medical data. They take charge of training local models on local medical datasets, and then submitting the masked local models and related secure parameters to the server. Ultimately, the optimal global model is generated through mutual cooperation between the server and the clients in each epoch. Different from the traditional federated learning framework, in order to enhance the privacy preservation of the local model, we introduce a third-party trust authority (TA) which is responsible for initializing some security parameters such as public keys and private keys, etc.

During the entire process of joint training, the server and all clients are considered to be honest but curious, which means that they can comply with the protocol execution but intend to snoop the privacy of others from any intercepted information. Therefore, there needs to fulfill the following security requirements so as to preserve privacy:

(1) In the process of local training, the clients cannot learn any authenticate models of other clients, except their local models and the aggregated global model, so that any sensitive raw data of other clients are not inferred.

(2) The server can only obtain masked models and related secure parameters submitted by the clients, but not the original local model and medical data. However, it can still generate an aggregated global model.

(3) The quality of data located in each client should be confidential. Data quality is an important metric in our system for measuring the contribution rate of different local models in the global model. It is also a significant factor for achieving secure aggregation. Due to the discrimination which server probably brings to some clients with poor data qualities, data quality should be kept confidential. In this way, it can ensure that all clients participate in federated learning fairly and impartially.

In our system, there perhaps exists some malicious and external adversaries who are trying to infer medical privacy by eavesdropping on the messages transmitted in the channel. The internal server in the system may also collude with some clients to satisfy their curiosity about private data. We dedicate to resist against passively malicious adversaries and collusion attacks. However, the deliberately disruptive behaviors on model training such as tampering attacks, impersonation attacks and poisoning attacks are beyond our consideration.

B. Data Quality and Contribution Rate

In previous work, the weighted average scheme based on dataset size is commonly adopted for model aggregation. The clients with more low-quality data participate in global model training with the same contribution rate, the accuracy of the model is bound to be impaired. How to measure the contribution rate of local data quality to the global model is the prime problem to be solved. In Miao *et al.* [44] and Xu *et al.* [45], a truth discovery algorithm is used to calculate the distance between the observed value of the data and the true value for estimating the reliability of the data source. The algorithm has shown its good performance in the quality evaluation of heterogeneous data for multiple application scenarios such as crowdsourcing and medical treatment. In SGD algorithm, the gradient is an important indicator for optimizing model convergence. Hsieh *et al.* [46] utilized local gradient amplitude to measure the proportion of the local model in the global model. However, the consistency of the local gradient with the global convergence trend was not considered.

Indeed, the gradient amplitude only represents the convergence speed, and it is easily affected by the size of the dataset and the learning rate. Only the sign of gradient can reflect the optimization direction and the convergence trend. Therefore, we calculate the data quality by referring to the truth discovery algorithm. Nevertheless, we compute the distance between the local gradient and the global gradient in each training epoch instead of the distance between the observed value and the true value described in the truth discovery algorithm.

In our system, the model parameters are exchanged between the clients and the server, instead of the gradients. However, according to SGD algorithm, we can still obtain the global

gradient through two global models trained in two adjacent epochs. Supposing that $\mathbf{g}_{global}^{t}$ is the global gradient in the t-th epoch, $\mathbf{w}_{global}^{t}$ and $\mathbf{w}_{global}^{t+1}$ represent the global models in the t-th epoch and the $(t+1)$-th epoch, respectively. Then, $\mathbf{g}_{global}^{t}$ can be deduced through the formula (12).

$$\mathbf{g}_{global}^{t}=\frac{\mathbf{w}_{global}^{t}-\mathbf{w}_{global}^{t+1}}{\beta} \tag{12}$$

In order to obtain the data quality in the t-th epoch, the client i needs to calculate the distance $\|\mathbf{g}_i^t-\mathbf{g}_{global}^t\|_2$ between the local gradient $\mathbf{g}_i^t$ and the global gradient $\mathbf{g}_{global}^t$. In our system, the data quality is necessary before the client uploads the local model to the server, while the global gradient in current epoch can only be calculated after the local models are aggregated by the server. For demonstrating this view, we define the normalized distance dis_{global}^{t} of two neighboring gradients as shown in formula (13), and then perform some model training on the datasets.

$$dis_{global}^{t}=\frac{\left\|\mathbf{g}_{global}^{t}-\mathbf{g}_{global}^{t-1}\right\|_2}{\left\|\mathbf{g}_{global}^{t-1}\right\|_2} \tag{13}$$

Most of the normalized distance dis_{global}^{t} are near the zero point, which illustrates the approximation between two neighboring gradients. Therefore, the global gradient $\mathbf{g}_{global}^{t}$ in the t-th epoch can be substituted by the global gradient $\mathbf{g}_{global}^{t-1}$ in the $(t-1)$-th epoch. The data quality Q_i^t for the client i in the t-th epoch is defined as formula (14).

$$Q_i^t=\frac{\delta}{\left\|\mathbf{g}_i^t-\mathbf{g}_{global}^{t-1}\right\|_2}=\frac{\delta}{\sum_{d=1}^{|\mathbf{g}|}\left(g_{i-d}^{t}-g_{global-d}^{t-1}\right)^2} \tag{14}$$

Here, $\delta=\mathcal{X}^2_{(1-\tau/2,|\mathbf{g}|)}$ represents the scale factor, which is a public parameter [26]. The symbol $\mathcal{X}$ and τ means the chi-square distribution and the significant level, respectively. $|\mathbf{g}|$ is the dimension of the gradient vector. g_{i-d}^{t} and $g_{global-d}^{t-1}$ respectively represents the d-th gradient in local gradient vector of the t-th epoch for the client i and the d-th gradient in global gradient vector in the $(t-1)$-th epoch. Therefore, if the signs of g_{i-d}^{t} and $g_{global-d}^{t-1}$ are opposite, we set the distance to a larger value so as to make the data quality lower. While the smaller distance shows that the local model can follow the optimization direction of the global model better, and the data quality is more suitable for the next iteration. After calculating Q_i^t, the global model for the t-th epoch can be aggregated with a weighted average algorithm based on Q_i^t, which is shown as formula (15).

$$\mathbf{w}_{global}^{t}=\frac{\sum_{i=1}^{N}Q_i^t w_i^t}{\sum_{i=1}^{N}Q_i^t} \tag{15}$$

where N denotes the number of clients participating in federated training, w_i^t represents the local model of the client i in the t-th epoch. It can be seen from formula (15) that better the data quality is, more proportion the local model can take in the global model. In short, better quality means the client can bring greater contribution rate.

C. Secure Aggregation Process

Our proposed privacy-preserving scheme for federated learning is described in detail in this section. In the scheme, homomorphic encryption technology is combined with double-masking mode proposed by Bonawitz *et al.* [9]. Furthermore, the contribution rate of data quality is also considered, which is more practical for federated learning scenarios with multiple medical institutions. The specific construction is depicted in the following four subsections.

1) System Initialization: Before performing federated training, the server confirms the set of clients participating in training (marked as $\mathcal{P}_1$) and the network model such as DNN, CNN, etc. Simultaneously, it defines the learning rate β, the maximum epochs number T for training convergence, the initialized global model $\mathbf{g}_{global}^{0}$ and the threshold e in Shamir secret sharing, which determines the maximum number of clients allowed to dropout or maliciously collude during the training process. Subsequently, some secure parameters and pairwise keys need to be generated to guarantee privacy security in the training process.

Firstly, TA selects the security parameter k as input of initializing function, the outputs are public parameters (q,g,G). Then, TA generates T pairs of public-private keys $\{(s_i^{sk-t},s_i^{pk-t})\}_{t\in\{1,2,\ldots,T\}}$ and a pair of keys (c_i^{sk},c_i^{pk}) for each client $i\in\mathcal{P}_1$. The maximal number of epochs equals to the number of pairs of public-private keys, and one pair of public-private key will be used in one epoch only. (s_i^{sk-t},s_i^{pk-t}) is used to mask the local model trained in the t-th epoch, while (c_i^{sk},c_i^{pk}) is for encrypting and decrypting the messages exchanged between the clients. After all keys are generated, TA sends them to the corresponding client i through a secure channel and broadcasts the tuples of these public keys $\{(i,c_i^{pk},\{t,s_i^{pk-t}\}_{t\in\{1,2,\ldots,T\}})\}_{i\in\mathcal{P}_1}$ in the system.

Next, TA selects a random number $s\in Z_q^*$ as the system private key s_{sys}^{sk}, and calculates $g^s\in G$ as the system public key s_{sys}^{pk}. It is noted that s_{sys}^{sk} should be kept confidential strictly and even not leaked to any model training participants, including the server and all clients. On the contrary, s_{sys}^{pk} should be broadcasted in the system. After that, according to the Shamir secret sharing mechanism, TA generates the secret shares (x_i,s_{sys-i}^{sk}) of the system private key s_{sys}^{sk} for each client $i\in\mathcal{P}_1$ by constructing a polynomial with degree (e-1) through the formula (5). Here, x_i is public and s_{sys-i}^{sk} is kept secret.

When all clients are informed of the training network, related parameters, initialized global model, as well as being distributed the pairwise keys and the secret shares of the system private key, the initialization is completed, and the federated learning process with the privacy preservation can be initiated.

2) Masking of the Local Model: After obtaining the related messages, each client starts to train the local model w_i^t with SGD algorithm. Then, it calculates the data quality Q_i^t before the local model is submitted to the server. In order to prevent the local model from being snooped by malicious external adversaries, the server and the clients, some cryptographic information needs to be exchanged among all clients, so as to achieve the purpose of privacy-preservation.

At first, client i selects the private key s_i^{sk-t} corresponding to the current epoch t, and executes secret shares generation algorithm:

$$s_i^{sk-t} \xrightarrow{\text{SSS.share}(s_i^{sk-t}, e, \mathcal{P}_1)} \{(s_{i,j}^{sk-t})\}_{j\in\mathcal{P}_1\backslash\{i\}}$$

to divide s_i^{sk-t} into n shares. It also uses the Diffie-Hellman key agreement function (denoted as f_{D-H}) to calculate symmetric key $c_{i,j} = f_{D\text{-}H}(c_i^{sk}, c_j^{pk})$ with each client $j \in \mathcal{P}_1\backslash\{i\}$. After that, client i encrypts the secret share $s_{i,j}^{sk-t}$ with the symmetric key $c_{i,j}$ to generate the ciphertext $ct_{i,j}^t = Enc_{c_{i,j}}(s_{i,j}^{sk-t})$, and sends $(i, j, ct_{i,j}^t)$ to client j.

When client j receives ciphertexts $\{(i, j, ct_{i,j}^t)\}_{i\in\mathcal{P}_2\backslash\{j\}}$ from at least e clients (marked as $\mathcal{P}_2 \subseteq \mathcal{P}_1$), it calculates the symmetric key $c_{j,i} = f_{D\text{-}H}(c_j^{sk}, c_i^{pk})$ with the corresponding client i, and decrypts $ct_{i,j}^t$ with $c_{j,i}$ to obtain the secret share $s_{i,j}^{sk-t} = Dec_{c_{j,i}}(ct_{i,j}^t)$. Here, client j must store all secret shares safely.

After obtaining the local model expressed by w_i^t and data quality expressed by Q_i^t, client i calculates the symmetric key $s_{i,j}^t = f_{D\text{-}H}(s_i^{sk-t}, s_j^{pk-t})$ with other clients $j \in \mathcal{P}_2\backslash\{i\}$, and takes $s_{i,j}^t$ as the seed of the pseudo-random generator (PRG) to calculate the masked local model according to formula (16).

$$y_i^t = w_i^t Q_i^t + \sum_{j\in\mathcal{P}_2\backslash\{i\}, i<j} PRG(s_{i,j}^t) - \sum_{j\in\mathcal{P}_2\backslash\{i\}, i>j} PRG(s_{i,j}^t) + \varphi Q_i^t (\bmod R) \tag{16}$$

In formula (16), φ is a public scale factor which can be preset. $PRG(s_{i,j}^t)$ and φQ_i^t are numbers, while $w_i^t Q_i^t$ is a vector, so $PRG(s_{i,j}^t)$ and φQ_i^t are needed to be repeated to get a vector that matches the dimension of $w_i^t Q_i^t$. Furthermore, it is necessary to quantize Q_i^t to an integer for the subsequent encryption operation. For the floating-point number (Q_i^t), its quantization follows the following two steps: first, the floating-point number is round up to n decimal places, then we expand it to 10^n times to get the quantized integer. That is to say, we ensure that it can only be accurate to n decimal place, and we lost precision in the fractional part. On the contrary, we do an inverse operation on the server side, scaling down the integer value to $1/(10^n)$ of its value, resulting in a floating-point number. The precision parameter n is a tunable parameter, and its value can be adjusted in experiments.

After completing local model masking, client i encrypts the data quality Q_i^t by utilizing the improved ElGamal homomorphic encryption algorithm and the system public key g^s. It randomly selects a number $r_i \in Z_q^*$ to calculate the ciphertext (c_{i1}, c_{i2}) according to formula (17), and then sends the tuple $(i, y_i^t, c_{i1}, c_{i2})$ to the server.

$$Enc_{s_{sys}^{pk}}(Q_i^t) = (c_{i1}, c_{i2}) = (g^{r_i}, 2^{Q_i^t} g^{sr_i}) \tag{17}$$

3) Aggregation of the Local Model: After collecting the masked model and the ciphertext $(i, y_i^t, c_{i1}, c_{i2})$ from at least e clients (marked as $\mathcal{P}_3 \subseteq \mathcal{P}_2$), the server starts to aggregate these masked models.

Firstly, it judges whether $\mathcal{P}_3 = \mathcal{P}_2$ is true. If yes, it directly aggregates $\{y_i^t\}_{i\in\mathcal{P}_3}$ and executes the steps described. Otherwise, the server finds out the clients existing in $\mathcal{P}_2$ but not in $\mathcal{P}_3$ (marked as $\mathcal{P}_2\backslash\mathcal{P}_3$). In fact, $\mathcal{P}_2\backslash\mathcal{P}_3$ represents the dropped clients during the uploading process of the masked models. Next, the server broadcasts the identity list in $\mathcal{P}_2\backslash\mathcal{P}_3$ to all clients for enquiring the shares of the dropped clients. After receiving the list, client j submits the shares $\{(s_{i,j}^{sk-t})\}_{i\in\mathcal{P}_2\backslash\mathcal{P}_3}$ to the server voluntarily.

When the server receives the shares $\{(s_{i,j}^{sk-t})\}_{j\in\mathcal{P}_4, i\in\mathcal{P}_2\backslash\mathcal{P}_3}$ from at least e clients (marked as $\mathcal{P}_4$), it executes the secret reconstruction algorithm:

$$\{(s_{i,j}^{sk-t})\}_{j\in\mathcal{P}_4\backslash\{i\}} \xrightarrow{\text{SSS.recons}(e, \{(s_{i,j}^{sk-t})\}_{j\in\mathcal{P}_4\backslash\{i\}})} s_i^{sk-t}$$

to reconstructs the keys $\{(s_i^{sk-t})\}_{i\in\mathcal{P}_2\backslash\mathcal{P}_3}$ for the dropped clients $\mathcal{P}_2\backslash\mathcal{P}_3$. Then, the server calculates symmetric key $s_{i,j}^t = f_{D\text{-}H}(s_i^{sk-t}, s_j^{pk-t})$ between each dropped client i and the other clients $j \in \mathcal{P}_3\backslash\{i\}$ one by one. What we need to pay attention to is that, if there exists a dropout client that recovers in a short time, another private key will be used in the next epoch to ensure security, instead of the reconstructed private key in previous epoch by the server. As we mentioned before, one pair of public-private key will be used in one epoch only, and the purpose is to prevent key leakage in client-dropout situations.

Subsequently, the server aggregates the masked models $\{y_i^t\}_{i\in\mathcal{P}_3}$ from the online clients $\mathcal{P}_3$. In order to offset $\{PRG(s_{i,j}^t)\}_{i\in\mathcal{P}_3, j\in\mathcal{P}_2\backslash\mathcal{P}_3}$ that have been incorporated into masked models of the online clients, the server takes $\{(s_{i,j}^t)\}_{i\in\mathcal{P}_2\backslash\mathcal{P}_3, j\in\mathcal{P}_3}$ as the seeds of the PRGs, and then aggregates all masked local models submitted by clients in $\mathcal{P}_3$ into a value θ, which is shown as formula (18).

$$\begin{aligned}\theta &= \sum_{i\in\mathcal{P}_3} y_i^t + \sum_{i\in\mathcal{P}_2\backslash\mathcal{P}_3, j\in\mathcal{P}_3, i<j} PRG(s_{i,j}^t) - \sum_{i\in\mathcal{P}_2\backslash\mathcal{P}_3, j\in\mathcal{P}_3, i>j} PRG(s_{i,j}^t) \\ &= \sum_{i\in\mathcal{P}_3} w_i^t Q_i^t + \varphi \sum_{i\in\mathcal{P}_3} Q_i^t \end{aligned} \tag{18}$$

It can be seen from formula (18) that as long as the data quality Q_i^t of each client in $\mathcal{P}_3$ is known, the aggregation result $\sum_{i\in\mathcal{P}_3} w_i^t Q_i^t$ can be easily calculated by $\theta - \varphi \sum_{i\in\mathcal{P}_3} Q_i^t$. Furthermore, the global model is obtained according to formula (15). However, the server cannot acquire Q_i^t directly due to confidentiality of the data quality, but it can still get the encrypted Q_i^t. Through homomorphic decryption algorithm depicted in the following subsection, the global model can be calculated.

4) Generation of the Global Model: The decrypting operations on aggregation ciphertext of the data quality $\{Q_i^t\}_{i\in\mathcal{P}_3}$ is performed based on the improved ElGamal homomorphic encryption algorithm.

Firstly, the server aggregates the ciphertexts expressed by$\{(c_{i1}, c_{i2})\}_{i\in\mathcal{P}_3}$ sent by each client $i \in \mathcal{P}_3$ according to formula (19).

$$c_1 = \prod_{i\in\mathcal{P}_3} c_{i1} = \prod_{i\in\mathcal{P}_3} g^{r_i} = g^{\sum_{i\in\mathcal{P}_3} r_i} = g^{r}$$
$$c_2 = \prod_{i\in\mathcal{P}_3} c_{i2} = \prod_{i\in\mathcal{P}_3} 2^{Q_i^t} g^{sr_i} = 2^{\sum_{i\in\mathcal{P}_3} Q_i^t} g^{s\sum_{i\in\mathcal{P}_3} r_i} = 2^{\sum_{i\in\mathcal{P}_3} Q_i^t} g^{sr} \tag{19}$$

It can be seen from formula (19) that as long as the system private key s is known, the server can decrypt the aggregation ciphertext (c_1, c_2) and get $\sum_{i\in\mathcal{P}_3} Q_i^t$. However, only TA owns the secret key s, which is not leaked to the server and any clients. Therefore, the server needs to request assistance from e clients to complete the decryption according to the secret reconstruction algorithm.

The server randomly selects e online clients (marked as $\mathcal{P}_5 \subseteq \mathcal{P}_3$) from $\mathcal{P}_3$, and calculates δ_i and $c_1^{\delta_i}$ shown as formula (20), where, x_i is a part of the secret share (x_i, s_{sys-i}^{sk}) which is chosen by TA in the initialization phase and can be accessed by server. Then, the server transmits $c_1^{\delta_i}$ to the corresponding client $i \in \mathcal{P}_5$.

$$\delta_i = \prod_{j\in\mathcal{P}_5, j\neq i} \frac{-x_j}{(x_i - x_j)}, \quad c_1^{\delta_i} = g^{r\delta_i} \tag{20}$$

When client i receives $c_1^{\delta_i}$, it computes λ_i with the secret share s_{sys-i}^{sk} only known by itself, which is shown as formula (21). Then, it sends λ_i back to the server.

$$\lambda_i = (c_1^{\delta_i})^{s_{sys-i}^{sk}} = g^{\prod_{j\in\mathcal{P}_5, j\neq i} \frac{-x_j}{(x_i-x_j)} s_{sys-i}^{sk} \cdot r} \tag{21}$$

After receiving $\{\lambda_i\}_{i\in\mathcal{P}_5}$ from all clients in $\mathcal{P}_5$, the server aggregates them according to formula (22).

$$\begin{aligned}\prod_{i\in\mathcal{P}_5} \lambda_i &= \prod_{i\in\mathcal{P}_5} g^{\prod_{j\in\mathcal{P}_5, j\neq i} \frac{-x_j}{(x_i-x_j)} s_{sys-i}^{sk} \cdot r} \\ &= g^{\left[\sum_{i\in\mathcal{P}_5} \prod_{j\in\mathcal{P}_5, j\neq i} \frac{-x_j}{(x_i-x_j)} s_{sys-i}^{sk}\right] \cdot r} = g^{sr}\end{aligned} \tag{22}$$

So far, the server can calculate the sum of data quality through formula (23).

$$\sum_{i\in\mathcal{P}_3} Q_i^t = \log_2 \frac{c_2}{\prod_{i\in\mathcal{P}_5} \lambda_i} \tag{23}$$

Finally, known the value θ and the scale factor φ, the server can generate the global model by aggregating the local model of the online clients, which is shown as formula (24).

$$\mathbf{w}_{global}^t = \frac{\theta - \varphi \sum_{i\in\mathcal{P}_3} Q_i^t}{\sum_{i\in\mathcal{P}_3} Q_i^t} = \frac{\sum_{i\in\mathcal{P}_3} w_i^t Q_i^t}{\sum_{i\in\mathcal{P}_3} Q_i^t} \tag{24}$$

V. Security Analysis

In our system, the server and all clients are considered to be semi-honest. Therefore, they can execute the secure aggregation process authentically. They neither tamper with the messages transmitted in channel, nor forge public-private keys and secret shares. Therefore, some cryptographic primitives such as public key infrastructure (PKI) and digital signatures are not involved in our system. Given the semi-honest model, we analyze the security of our scheme, including the security of the masking mode and the security of the homomorphic encryption.

A. Security of the Masking Mode

First of all, we take $au_i^t = w_i^t Q_i^t + \varphi Q_i^t$ as the real input. Our scheme aims to mask the real input au_i^t by superimposing a random number on it. According to the properties of the random number, we can see that the masked result is random. Then, in our aggregation model, as long as the sum of the masked results from all clients is equal to the sum of all real inputs, the adversary cannot distinguish the real input from the masked result. Because the masked result hides the real input of each client, the view of the adversary is only the sum of the real inputs instead of any individual real input.

Lemma 1: Given $R, \mathcal{P}, \forall i, j \in \mathcal{P}, au_i \in Z_R$, then

$$\begin{aligned}&\{\{PRG_{i,j} \leftarrow Z_R\}_{i<j}, \ \forall i > j, \\ &\quad PRG_{i,j} \leftarrow -PRG_{j,i} : \{au_i + \sum_{j\in\mathcal{P}\setminus\{i\}} PRG_{i,j}\}_{i\in\mathcal{P}}\} \\ &\quad \equiv \{\{\xi_i \leftarrow Z_R\}_{i\in\mathcal{P}}, \ s.t. \quad \sum_{i\in\mathcal{P}} \xi_{i,j} = \sum_{i\in\mathcal{P}} au_i : \{\xi_i\}_{i\in\mathcal{P}}\}\end{aligned}$$

Here, the symbol "$\equiv$" denotes two groups of random variables that are computationally indistinguishable.

Next, we analyze two scenarios, one is the collusion of clients only, and the other is the collusion between some clients and the server. All clients are divided into five categories, namely, $\mathcal{P}_5 \subseteq \mathcal{P}_4 \subseteq \mathcal{P}_3 \subseteq \mathcal{P}_2 \subseteq \mathcal{P}_1$. $\mathcal{P}_1$ represents all clients in the federated learning system; $\mathcal{P}_2$ represents the clients participating in the model masking, which means their pairwise keys are incorporated into the mask; $\mathcal{P}_3$ represents the online clients that successfully uploads the masked models; $\mathcal{P}_4$ represents the clients contributing the secret shares of dropped clients to the server, and $\mathcal{P}_5$ is the clients joining in the decryption of data quality. For simplifying the description, we set the server as S and the colluding parties as $\mathcal{A}$. $view_{\mathcal{A}}^{e,k}(au_{\mathcal{P}}, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4, \mathcal{P}_5)$ is a random variable, which represents all view the colluding parties $\mathcal{A}$ can obtain in the entire federated learning process, given threshold e in Shamir secret sharing algorithm and system security parameter k.

Theorem 1: (Honest but curious security, collusion by clients only). Assuming that there is a PPT simulator $\mathcal{C}$ who attempts to break the semantic security of our masking mode with the help of $\mathcal{A}$. If given $e, k, \mathcal{P}, au_{\mathcal{P}}, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4, \mathcal{P}_5$ with $|\mathcal{P}| \geq e, \mathcal{A} \subseteq \mathcal{P}, |\mathcal{A}| < e, \mathcal{P}_5 \subseteq \mathcal{P}_4 \subseteq \mathcal{P}_3 \subseteq \mathcal{P}_2 \subseteq \mathcal{P}_1$, the view acquired by the simulator $\mathcal{C}$ is indistinguishable from the view of $\mathcal{A}$.

$$\begin{aligned}&view_{\mathcal{A}}^{e,k}(au_{\mathcal{P}}, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4, \mathcal{P}_5) \\ &\equiv view_{\mathcal{C}}^{e,k}(au_{\mathcal{A}}, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4, \mathcal{P}_5)\end{aligned}$$

Proof: In the first honest but curious threat model, we only refer to the collusion of the clients, while the server is considered to be absolutely honest, which means all clients

do not actively tamper with the data transmitted in the channel. The server only transmits the data that each client deserves. In our system, the honest clients only communicate with the server, and they do not share the masked results with $\mathcal{A}$. Furthermore, even intercepting the masked results, $\mathcal{A}$ can still not eliminate the random numbers from the masked results. Therefore, the joint view of colluding parties $\mathcal{A}$ does not depend on the view of the other honest parties $\mathcal{P}_2\backslash\mathcal{A}$. Although the simulator $\mathcal{C}$ can obtain the real input au_i^t of $\mathcal{A}$ in the simulation, it can only forge the real inputs of the honest user, which means the view of the simulator $\mathcal{C}$ is equivalent to the joint view of $\mathcal{A}$. ■

Theorem 2: (Honest but curious security, collusion by clients and server). Assuming that there is a PPT simulator $\mathcal{C}$ who attempts to break the semantic security of our masking mode with the help of $\mathcal{A}$. If given e, k, $\mathcal{P}$, $au_{\mathcal{P}}$, $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{P}_3$, $\mathcal{P}_4$, $\mathcal{P}_5$ with $|\mathcal{P}| \geq e$, $\mathcal{A} \subseteq \mathcal{P} \cup \{S\}$, $|\mathcal{A}\backslash\{S\}| < e$, $\mathcal{P}_5 \subseteq \mathcal{P}_4 \subseteq \mathcal{P}_3 \subseteq \mathcal{P}_2 \subseteq \mathcal{P}_1$, the view of the simulator $\mathcal{C}$ is indistinguishable from the view of $\mathcal{A}$.

$$\begin{aligned} &view_{\mathcal{A}}^{e,k}(au_{\mathcal{P}}, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4, \mathcal{P}_5) \\ \equiv\ &view_{\mathcal{C}}^{e,k}(au_{\mathcal{A}}, \sigma, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4, \mathcal{P}_5) \\ &\text{where } \sigma = \sum_{i\in\mathcal{P}_3\backslash\mathcal{A}} au_i \end{aligned}$$

In the second honest but curious threat model, given the real inputs of the colluding party $\mathcal{A}$ and the sum of the real inputs of all honest online clients $\mathcal{P}_3\backslash\mathcal{A}$; then, the simulator $\mathcal{C}$ can still not learn the real input of any individual honest client, except the given view and the view of $\mathcal{A}$. In particular, we emphasize that only the collusion joined by less than e clients cannot break the security of our system. The following proof process abides by this premise.

Proof: (1) Firstly, the simulator $\mathcal{C}$ makes use of $\mathcal{A}$ to get the same view with $\mathcal{A}$.

(2) Next, we simulate each honest client $i \in \mathcal{P}_2\backslash\mathcal{A}$, and replace its symmetric key $c_{i,j} = f_{D-H}(c_i^{sk}, c_j^{pk})$ with a randomly selected key $c'_{i,j} \in G$. Then, we encrypt the secret shares $s_{i,j}^{sk-t}$ with random key $c'_{i,j}$, and transmits the ciphertext to other clients $j \in \mathcal{P}_2\backslash\mathcal{A}$. According to the Decisional Diffie-Hellman assumption, in the case of unknowing private key, $c'_{i,j}$ and $c_{i,j}$ are indistinguishable, so the view of the simulator $\mathcal{C}$ is no more than the colluding parties $\mathcal{A}$.

(3) In this step, we simulate an honest client $i \in \mathcal{P}_2\backslash\mathcal{A}$, and replace its real secret share $s_{i,j}^{sk-t}$ with a randomly selected share $s'_{i,j} \in Z_q^*$, Then send the ciphertext of the random share $s'_{i,j}$ to other clients $j \in \mathcal{P}_2\backslash\mathcal{A}$. When calculating the model mask, we still use the real pairwise keys to generate the random number. If the server wants to restore the real input from the masked result of the honest client, it can only ask clients to upload the secret share $s'_{i,j}$. Since the ciphertext of the real share $s_{i,j}^{sk-t}$ and the ciphertext of the random share $s'_{i,j}$ have indistinguishability under chosen-plaintext attack (IND-CPA), the server cannot distinguish $s_{i,j}^{sk-t}$ from dummy $s'_{i,j}$, therefore it cannot reconstruct the true private key of the honest client. In other words, even with the help of the compromised server, the view of the simulator $\mathcal{C}$ is no more than the previous step.

(4) We simulate an honest client $i \in \mathcal{P}_2\backslash\mathcal{A}$, and replace the real secret share $s_{i,j}^{sk-t}$ of its private key s_i^{sk-t} with a randomly selected share $s'_{i,j} \in Z_q^*$, Then send the random share $s'_{i,j}$ to the colluding client $j \in \mathcal{A}$. When calculating the model mask, we still use the real pairwise keys to generate the random number. According to the Shamir secret sharing algorithm, if $|\mathcal{A}| < e$, the private key cannot be reconstructed. Therefore, as long as the random shares are identical in distribution with the real secret shares, the simulator $\mathcal{C}$ cannot obtain additional information except the view of $\mathcal{A}$.

(5) We generate a random key $s'_{v,j} \in G$ for a special selected client $v \in \mathcal{P}_3\backslash\mathcal{A}$ to substituting its real symmetric key $s_{v,j}^t = f_{D-H}(s_v^{sk-t}, s_j^{pk-t})$ agreed with other clients $j \in \mathcal{P}_2$. Then take $s'_{v,j}$ as the seed of PRG to calculate forged masked results for client v and the other clients $i \in \mathcal{P}_3\backslash\mathcal{A}\backslash\{v\}$, which are respectively shown as formula (25) and formula (26). Finally, the forged masked results y_v^t and y_i^t are sent to the server.

$$\begin{aligned} y'_v = au_v^t &+ \sum_{j\in\mathcal{P}_2\backslash\{v\}, v<j} PRG(s'_{v,j}) \\ &- \sum_{j\in\mathcal{P}_2\backslash\{v\}, v>j} PRG(s'_{v,j}) \end{aligned} \tag{25}$$

$$\begin{aligned} y'_i = au_i^t &+ \sum_{j\in\mathcal{P}_2\backslash\{v,i\}, i<j} PRG(s_{i,j}^t) \\ &- \sum_{j\in\mathcal{P}_2\backslash\{v,i\}, i>j} PRG(s_{i,j}^t) + PRG(s'_{i,v}) \\ \text{where } i > v, &PRG(s'_{i,v}) := -PRG(s'_{i,v}) \end{aligned} \tag{26}$$

According to the Decisional Diffie-Hellman assumption, $s_{v,j}^t$ and $s'_{v,j}$ are indistinguishable, so $PRG(s_{v,j}^t)$ and $PRG(s'_{v,j})$ are identically distributed, the view of the simulator $\mathcal{C}$ is indistinguishable from the colluding party $\mathcal{A}$.

(6) We substitute $PRG(s'_{v,i})$ computed by the selected client v and other clients $i \in \mathcal{P}_3\backslash\mathcal{A}\backslash\{v\}$ in the previous step with randomly selected number $\gamma'_{v,i}$. As long as $\gamma'_{v,i}$ and $PRG(s'_{v,i})$ are distributed identically, the simulator $\mathcal{C}$ cannot distinguish $\gamma'_{v,i}$ from $PRG(s'_{v,i})$ due to the randomness of PRG. Thus, in this step, the view of the simulator $\mathcal{C}$ is no more than the previous step.

(7) When calculating the masked result for each client $i \in \mathcal{P}_3\backslash\mathcal{A}$, we replace the real input au_i^t with a random number ξ_i^t, but still superimpose the correct mask $PRG(s_{i,j}^t)$. After that, the fake masked result y'_i is sent to the server, instead of the real masked result y_i^t. The calculations of y_i^t and y'_i are respectively shown in formula (27) and formula (28).

$$\begin{aligned} y_i^t = au_i^t &+ \sum_{j\in\mathcal{P}_2\backslash\mathcal{A}\backslash\{i\}, i<j} PRG(s_{i,j}^t) \\ &- \sum_{j\in\mathcal{P}_2\backslash\mathcal{A}\backslash\{i\}, i>j} PRG(s_{i,j}^t) \end{aligned} \tag{27}$$

$$\begin{aligned} y'_i = \xi_i^t &+ \sum_{j\in\mathcal{P}_2\backslash\mathcal{P}_3\backslash\mathcal{A}\backslash\{i\}, i<j} PRG(s_{i,j}^t) \\ &- \sum_{j\in\mathcal{P}_2\backslash\mathcal{P}_3\backslash\mathcal{A}\backslash\{i\}, i>j} PRG(s_{i,j}^t) \end{aligned} \tag{28}$$

Here, ξ_i^t subjects to the conditional distribution such that $\sum_{i\in\mathcal{P}_3\backslash\mathcal{A}}\xi_{i,j}=\sum_{i\in\mathcal{P}_3\backslash\mathcal{A}}au_i$. According to the Lemma 1, the simulator $\mathcal{C}$ cannot obtain the real input of each client $i\in\mathcal{P}_3\backslash\mathcal{A}$. The view of the simulator is only the sum of their real inputs. ■

From the simulation process mentioned above, it can be seen that the view of the simulator $\mathcal{C}$ is computationally indistinguishable from the colluding parties $\mathcal{A}$. Therefore, the view of the simulator is no more than the colluding parties $\mathcal{A}$, even in the colluding scenario between some clients and the server.

B. Security of the Homomorphic Encryption

The security analysis in the previous subsection has proved that the masking mechanism can protect the real inputs au_i^t. In this section, if given the real inputs $au_i^t=w_i^tQ_i^t+\varphi Q_i^t$, we only discuss whether the colluding parties $\mathcal{A}$ can decrypt data quality and restore the local model w_i^t through breaking the semantic security of homomorphic encryption. It is worth emphasizing that our threat model still needs to follow the premise that fewer than e clients participate in collusion.

In the encryption process, client i uses the system public key g^s to encrypt data quality and send the ciphertext $(g^{r_i},2^{Q_i^t}g^{sr_i})$ to the server. If the server wants to decrypt the data quality of client i, and furtherly snoop its local model w_i^t by equation $w_i^t=au_i^t/Q_i^t-\varphi$, it firstly needs to restore $2^{Q_i^t}$. However, the system private key s is kept secret for the server and all clients. Therefore, if the server expects to restore $2^{Q_i^t}$ without s, it either solves the discrete logarithm problem to deduce r_i from g^{r_i} and then calculates $(g^s)^{r_i}$, or solves the Decisional Diffie-Hellman problem to calculate g^{sr_i} according to g^{r_i} and g^s. So far, the two types of problems are recognized as NP-hard problems, and the server cannot solve them.

When the server cannot restore $s^{Q_i^t}$ by itself, it may collude with other clients to request their assistances for reconstructing the private key s. In our system, although the server also needs to cooperate with online clients to obtain the aggregated result $\sum_{i\in\wp_3}Q_i^t$ of data quality, it just needs to compute g^{sr} instead of reconstructing the private key s. However, the threat model of collusion is different from normal cooperation, and the colluding parties $\mathcal{A}$ aim to get the quality data Q_i^t of an individual honest client. Without the private key s, they can only obtain the sum of data qualities of all honest clients through computing $\sum_{i\in\mathcal{P}_3\backslash\mathcal{A}}Q_i^t=\sum_{i\in\mathcal{P}_3}Q_i^t-\sum_{i\in\mathcal{A}}Q_i^t$. Then, it seems that reconstructing s is the unique effective way to achieve their purpose. Unfortunately, the number of colluding clients $\mathcal{A}\backslash\{S\}$ is less than threshold e. According to the Shamir secret sharing algorithm, the private key s cannot be reconstructed from less than e shares. Therefore, the server and the client have no ability in decrypting a single Q_i^t, and the local model w_i^t is protected.

In summary, as long as less than e clients participate in collusion, our homomorphic encryption is semantically secure.

VI. Performance Analysis

In this part, we suppose that our system consists of a server and n clients, the dimension of the local model is m, and the threshold of the Shamir secret sharing algorithm is e. The theoretical analysis on computational cost and communication cost caused during an epoch of federated learning is performed. All costs brought by the system initialization are neglected, because initialization is executed only once, which would not affect the performance during the subsequent training process. We elaborate on the specific analysis process from the perspective of an individual client and the server respectively.

A. Cost Analysis of Client

1) Computation Cost: The computation cost of each client i in a training epoch is mainly generated by the following five operations:

1) When the local model is masked, in order to generate the seed of PRG, client i needs to calculate symmetric keys with other $(n-1)$ clients, that means the key agreement function $f_{D\text{–}H}(s_i^{sk-t},s_j^{pk-t})$ is performed $(n-1)$ times. In the same way, when the client encrypts or decrypts the secret shares $s_{i,j}^{sk-t}$ related with other $(n-1)$ clients, the function $f_{D\text{–}H}(c_i^{sk-t},c_j^{pk-t})$ is also executed $(n-1)$ times for generating symmetric key. Therefore, key agreement function is executed $2(n-1)$ times in total, and the computation complexity is approximately $O(n)$.
2) Client i needs to perform $(n-1)$ times encryption operations $Enc_{c_{i,j}}(s_{i,j}^{sk-t})$ on the secret shares distributed to others and $(n-1)$ times decryption operation $Dec_{c_{i,j}}(ct_{j,i}^t)$ on the secret shares received from others. It requires $2(n-1)$ encryption/decryption operations, then, the complexity is approximately $O(n)$.
3) Client i needs to create $(n-1)$ secret shares of its private key, and the secret share generation algorithm is run $(n-1)$ times, and the complexity is approximately $O(n)$.
4) Since the output of PRG is a number, while the local model is an m-dimensional vector, in order to facilitate calculation and satisfy better privacy, a PRG needs to be expanded into m dimensions. Furthermore, a masking tuple includes $(n-1)$ numbers of PRGs. Therefore, the total number of expansions is $m(n-1)$ times, then the complexity is $O(mn)$.
5) If client i is selected by the server to assist decryption, it needs to perform a homomorphic encryption operation on data quality, which is similar to the key agreement function in computation complexity. Therefore, the complexity is approximately $O(1)$.

From the analysis mentioned above, it concludes that the total computation complexity of a client in a training epoch is $O(n+n+n+nm+1)\approx O(nm)$.

2) Communication Cost: The communication cost of each client i in a training epoch mostly caused by four parts:

1) Client i sends $(n-1)$ numbers of encrypted secret shares to others, and receives $(n-1)$ encrypted secret shares from others.
2) It sends the masked model and the ciphertext of data quality $(i,y_i^t,\mathrm{c}_{i1},\mathrm{c}_{i2})$ to the server.

3) Assuming that there are d clients dropping out during the uploading of the masked model, and the client i is willing to contribute its shares, then it sends the secret shares of d dropped clients to the server. In the worst case, if $(n-e)$ clients drop out, client i needs to submit $(n-e)$ shares.
4) If the client i is selected to assist the server in homomorphic decryption, it receives $c_1^{\delta_i}$ sent by the server and responses λ_i to the server.

We set that the length of the public key and private key are a_k and a_s, respectively. The range of the elements in the masked model is R, and the length of ciphertext of data quality is a_c. According to the formula (20) and (21), the length of $c_1^{\delta_i}$ and λ_i are equivalent to the length of the public key. Then, the total cost caused by the four steps mentioned above is $[2(n-1)+d]a_s + 2a_k + a_c + m\lceil \log_2 R \rceil$. In real scenarios, the threshold e is usually much smaller than the number n of clients. The probability that the client i participates in homomorphic decryption is low. If it is unwilling to upload shares, the last two steps would not be executed. At this time, the total communication cost is minimum $2(n-1)a_s + a_c + m\lceil \log_2 R \rceil$.

B. Cost Analysis of Server

1) Computation Cost: The computation cost of server in a training epoch is mainly divided into four categories:

1) If there are d clients dropping out, the server needs to recover the keys for them. Therefore, it needs to perform d times secret reconstruction operations, and the complexity is $O(d)$.
2) In order to offset the PRG that is related to the dropped clients and has been incorporated into the masked model of online clients, it is necessary to calculate $(n-d)$ numbers of symmetric keys for each dropped client. Then, the server needs to perform $(n-d)d$ times of key agreement function $s_{i,j}^t = f_{D\text{-}H}(s_i^{sk-t}, s_j^{pk-t})$, the complexity is $O(nd)$.
3) Each symmetric key generated in the previous step is taken as the seed of PRG. Then, the server needs to calculate $(n-d)d$ times $PRG(s_{i,j}^t)$; furthermore, each $PRG(s_{i,j}^t)$ needs to be expanded into m dimensions. So, the complexity is $O((n-d)dm) \approx O(nmd)$.
4) The server performs homomorphic decryption operation on the data quality of $(n-d)$ online clients including $2(n-d)$ times multiplications of ciphertexts, e times calculation of δ_i and $g^{r\delta_i}$, e times multiplications of λ_i and 2 times division. Usually, d and e are much smaller than n, therefore, the total computation complexity is $O(2(n-d)+e+e+2) \approx O(n)$.

According to the analysis mentioned above, the total computation complexity of the server in a training epoch is $O(d + nd + nmd + n) \approx O(nmd)$. When the number of dropped clients reaches the maximum allowed in our system, i.e., $(n-e)$, the maximum cost of server is $O(nm(n-e)) \approx O(mn^2)$. If no client drops out, Step 1)–3) can be ignored, and the computation complexity is the lowest $O(n)$.

2) Communication Cost: The communication cost of server in a training epoch is composed of four aspects:

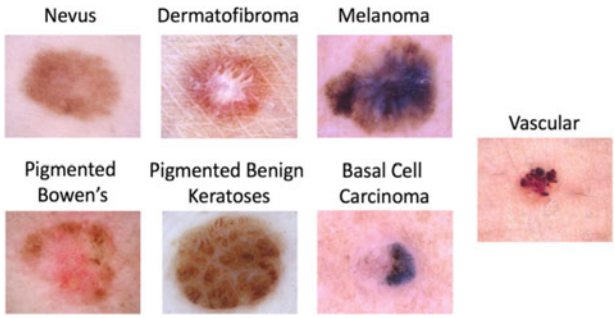


Fig. 2. Samples from the HAM10000 dataset.

1) Assuming that there exist d dropped clients, the server receives the masked models and the ciphertext of data quality $(i, y_i^t, c_{i1}, c_{i2})$ from $(n-d)$ online clients.
2) The server broadcasts an identities list of dropped clients to all online clients.
3) For reconstructing private keys of dropped clients, the server receives secret shares sent by e online clients, and each client sends d shares, then the server receives ed secret shares in total.
4) In order to complete the homomorphic decryption of data quality, the server requests assistance from e online clients. It sends $c_1^{\delta_i}$ e times and receives λ_i e times in this process.

If the cost caused by broadcasting the identities list in Step 2) can be ignored, the total communication cost of the server is $eda_s + 2ea_k + (n-d)a_c + (n-d)m\lceil \log_2 R \rceil$. In the case of no dropped clients, Step 3) will not be executed, and the communication cost is $2ea_k + na_c + nm\lceil \log_2 R \rceil$.

VII. Performance Evaluation

A. Datasets

For the effectiveness evaluation, the dataset we used is the HAM10000 ("Human Against Machine with 10,000 training images") dataset [47], consists of 10,015 dermatoscopic images of common pigmented skin lesions. It has 7 different class of skin cancer, namely: 1) Nevus, 2) Melanoma, 3) Pigmented Bowen's, 4) Basal Cell Carcinoma, 5) Pigmented Benign Keratoses, 6) Vascular, 7) Dermatofibroma. We use only the provided images without any usage of meta-data or external datasets.

Due to the imbalance of categories in the dataset, in order to improve the accuracy of the classification model and avoid overfitting, we expanded the original HAM10000 dataset and applied some data augmentation techniques. We modified the training data with small transformations to reproduce the variations. Several data augmentation techniques were applied during training, such as randomly rotating some training images by 5 degrees, randomly vertical and horizontal flip some training images by 10%. By applying the above transformations to our training data, we can expand the number of training images to 90,135, and the size of each image is 28*28 pixels.

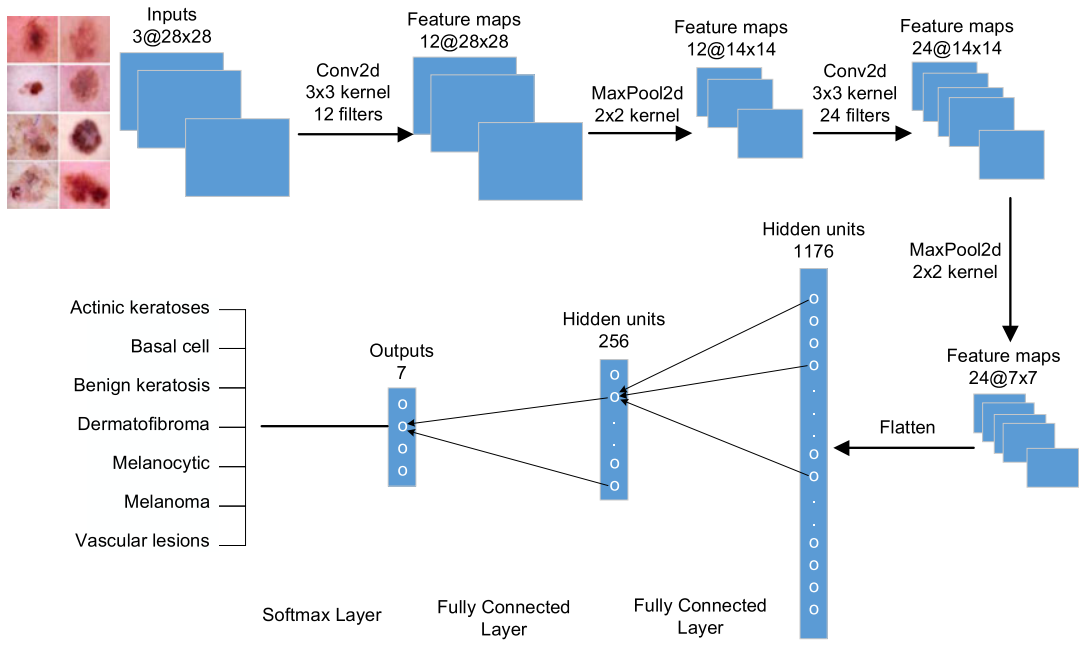


Fig. 3. Deep convolution neural networks for skin cancer detection in each client.

Then, we split the expanded dataset into training set (80,000 images) and testing set (10,135 images) with some randomness. Due to the needs of federated learning, we split the training set into multiple subsets of approximately the same size and distributed them to each client to train a local model. The training dataset owned by each client is independent and cannot be leaked, ensuring the security and privacy of the image. Furthermore, we split the train set owned by each client into two parts: 90% is used to train the model, and 10% is the validation set using which the model is evaluated. We encode labels which are 7 different classes of skin cancer (expressed by 0−6), and we need to encode these labels to one-hot vectors.

In this section, we introduce the steps to detect skin cancer types using deep convolution neural networks. Because this paper focuses on the privacy protection scheme, the network structure of the DNN used is just the conventional multiple convolutional layers, pooling layers, and fully connected layers. It does not pay too much attention to optimize the model, which can be fine-tuned to produce more accurate scores in the future. This paper only considers the simple optimization, such as data augmentations, and the data quality given to different clients when performing parameter updates.

The designed DNN network architecture is shown in Fig. 3, which is composed of two convolution layers, two MaxPooling layers, two full-connected layers, one Softmax layer. A total of 36 kernel filters are designed for convolution layer. The total number of parameters in the local model is 306,063. Model's parameters are initialized randomly, and then distributed to all clients. Each local model is trained by using local training dataset. The accuracy of the models is evaluated on the test dataset by the server.

TABLE I
FUNCTIONALITY COMPARISON OF FIVE SCHEMES

Scheme	dropout support	homomorphic encryption
Scheme I	√	×
Scheme II	√	×
Scheme III	×	√
Scheme IV	×	√
Scheme V	√	√

B. Comparison of Schemes

We compared the following five schemes, and the functionality comparison in terms of dropout support and homomorphic encryption is listed in Table I.

- Scheme I: the federated learning scheme named FedAvg proposed by McMahan *et al.* [5], which utilizes weighted averaging for parameter aggregation and updates without any additional privacy protection methods, such as homomorphic encryption.
- Scheme II: the federated learning scheme using average-based parameter aggregation and updates, and it could tolerate a certain degree of client dropout, which was proposed by Bonawitz *et al.* [9].
- Scheme III: the federated learning scheme named FedOpt proposed by Asad *et al.* [10], which designed a sparse compression algorithm for efficient communication and also integrated the homomorphic encryption.
- Scheme IV: the federated learning scheme based on homomorphic encryption for all parameters, proposed by Fang *et al.* [11], but could not tolerate a certain degree of client offline.
- Scheme V: the federated learning scheme proposed in this paper, which supports homomorphic encryption-

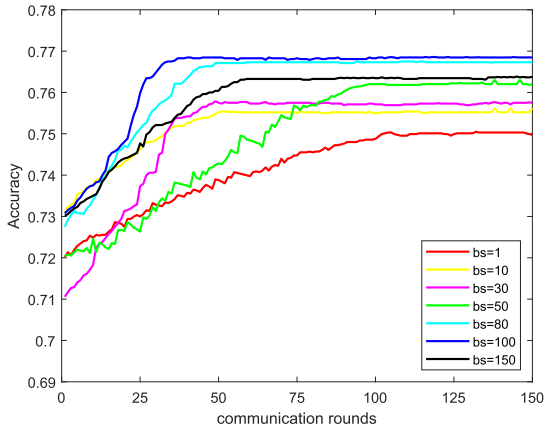


Fig. 4. Accuracy as the increase of communication rounds with different mini-batch size.

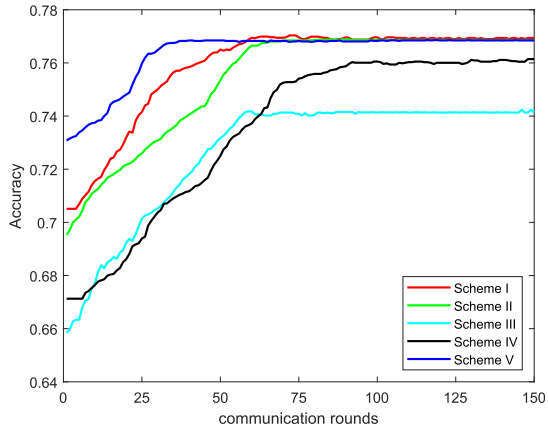


Fig. 5. Accuracy as the increase of communication rounds for five schemes.

based privacy protection and could also tolerate a certain degree of client dropout.

C. Comparison Result and Analysis

In this section, we present the effectiveness and efficiency of the proposed federated learning scheme through comparison of performance evaluation results. We implemented the prototype system of the proposed federated learning scheme in Java language, including three main components, parameter learning in local, homomorphic encryption and parameter updates. Using ElGamal encryption algorithm, a homomorphic encryption strategy is designed, which changes ElGamal cryptosystem from satisfying multiplication homomorphism to satisfying additive homomorphism, implemented by encrypting the exponential form of the plaintexts instead of original plaintexts. For the efficiency evaluation analysis, we run the federated learning on a Linux server equipped with Intel Xeon E5-1650 CPU (6 cores, 12 threads, 3.5 GHz) and 32 GB RAM, and the operating system is Ubuntu 18.04. We run the server program to control the learning and run multi-client programs in the same Linux server (the number of client is varied from 10 to 30). Since all client runs in the same server with only one IP address, we assign a unique port for each client. Communications is implemented by Java Remote Procedure Call (RPC) framework. The services encapsulated by clients is called by the server which controls key distribution, model parameter distribution, model training, and parameter aggregation. In order to ensure the fairness of the comparison, performance evaluation of the five schemes (Scheme I, II, III, IV and V) are performed using the same hardware and software environment, the same datasets, preprocessing and distribution methods, and the same library and multi-client communication environment. The homomorphic encryption and DNN involved in the five schemes use the same programming code. The value of parameter for quantization in our experiment is $n = 2$, which means it can only be accurate to 2 decimal places.

1) Accuracy: In this evaluation, during each training epoch, all clients communicate with the server for one time. The mini-batch size is denoted as bs, and the value of bs is varied from 1, 10, 30, 50, 80, 100 to 150. The learning rate is 0.01, and the number of clients is 10. As shown in Fig. 4, we use this comparative experiment to show that the accuracy of model classification increases with the increase of communication rounds when we set different bs. This figure also indicates the convergence rate. In addition, the accuracy of the model does not increase monotonically as the value of bs increases. When the mini-batch size is set to 100, we obtained the best accuracy. Thus, in the following evaluation, we also set the value of mini-batch size to 100.

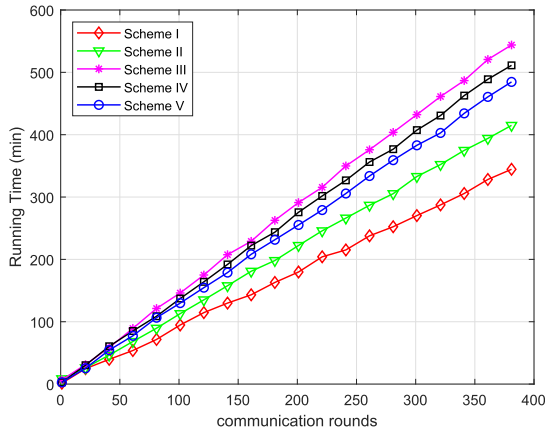


Fig. 6. Server running time as the increase of communication rounds.

Then, we compared the accuracy for the five schemes, as shown in Fig. 5. For Scheme V, we obtained a maximal accuracy of around 76.9% in the test set. However, due to the requirements of homomorphic calculations, the model parameters (data quality in the proposed scheme) will be quantified by integers, so the accuracy of the model will be lost to a certain extent for all homomorphic encryption-based schemes (Scheme III, IV, V). From the overall performance, Scheme III and IV are worse than others. The effect of the proposed scheme is similar to Scheme I and II.

2) Computation Cost: We evaluated the training time of different schemes. As you can see in Fig. 6, as the number of communication rounds increases, the training time becomes significantly longer. We use this comparative experiment to illustrate the time cost of homomorphic encryption and

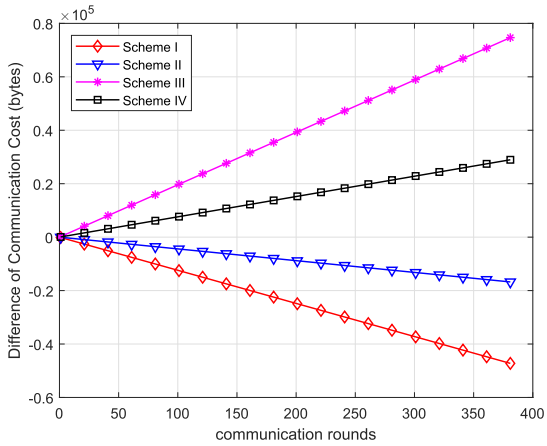


Fig. 7. Difference of communication cost as the increase of communication rounds.

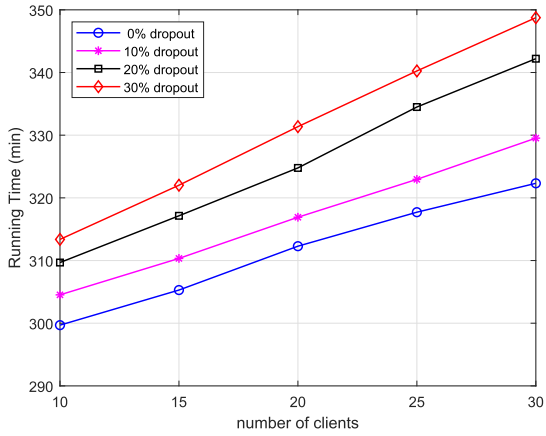


Fig. 8. Server running time under different dropout rates as the increase of client number.

decryption. A tentative conclusions has been drawn that homomorphisms will bring time cost. Compared with Bonawitz *et al.*'s scheme which doesn't rely on homomorphic encryption, the time for the proposed scheme has been increased by 15.03% when the communication round reaches 300. However, it can ensure the safety of model parameters, and the server cannot derive model parameters, which is also supported by Bonawitz *et al.*'s scheme.

3) Communication Cost: We evaluated the communication cost of five schemes. In order to compare them, the result of Scheme V is regarded as a benchmark for comparison. In order to avoid the curves overlapping and present more clearly, we calculate the difference between Scheme I, II, III, IV and Scheme V by subtraction, which is shown in Fig. 7. Since each client obtains a subset of the original data set in the federation learning, the communication cost mainly caused by parameter aggregations and key agreements. Scheme III obtained the worse result in terms of communication cost, and our scheme is a little better than Scheme IV. Scheme I obtained the lowest communication cost, due to its simplicity.

4) Client Dropout: Finally, we varied the fraction of dropout clients. Fig. 8 presents the server running time as the increase of dropout rates. Similar to Bonawitz *et al.*'s scheme, for each dropped client, the server must remove that client's masks, through the secret reconstruction algorithm to reconstructs the keys for the dropped clients, which has been presented in Section IV. It can be observed clearly from this figure that higher dropout rate brings higher cost of dealing with dropped clients. In the case of 30 clients, even if the dropout rate reaches 30%, the time is only increased by 8.29% compared with the situation of no dropout client.

VIII. Conclusion

In this paper, we have proposed a novel federated learning scheme for privacy-perseveration in IoT-based healthcare applications. A weighted average algorithm based on data quality is proposed to replace the traditional weight calculation method based on the amount of data. A novel masking scheme based on homomorphic encryption and the secure multi-party computation is proposed for federated learning. Moreover, the homomorphic encryption scheme does not encrypt every model parameter because the model in deep learning usually has high dimensions, and the homomorphic encryption on such high-dimensional data would bring huge computation overhead. There is only a variable called data quality to be encrypted for each client in each training epoch in our scheme. Therefore, the scheme proposed in the paper would not cause the computation overhead to increase sharply. An appropriate adjustment to the ElGamal encryption algorithm is made so as to change the characteristics of the algorithm from multiplicative homomorphism to additive homomorphism, which is suitable for the scheme proposed in the paper. Moreover, a flexible dropout-tolerable and participants collusion-resistible solution is provided in our scheme by employing Diffie-Hellman key exchange and Shamir secret sharing algorithm.

Moreover, this paper focuses on privacy protection schemes and classification accuracy. In our experiments, we are able to detect lesion cell type with an accuracy of more than 76.9%. The model can also be fine-tuned to perform and get a better accuracy in the future. However, we have not considered the situation of heterogenous clients with resource-constrained hardware and asynchronous federated learning. The proposed scheme still needs to be tuned in heterogenous environment to obtain a high efficiency. Furthermore, malicious server is out of the research scope of our proposed aggregation in federated learning, and we have not considered that the aggregated model is tampered or forged by the server. Verifiable aggregation in federated learning will also be one of our future work.

References

[1] M. Arshad *et al.*, "A computer-aided diagnosis system using deep learning for multiclass skin lesion classification," *Comput. Intell. Neurosci.*, vol. 2021, pp. 9619079:1–9619079:15, 2021.

[2] J. Shen, H. Yang, P. Vijayakumar, and N. Kumar, "A privacy-preserving and untraceable group data sharing scheme in cloud computing," *IEEE Trans. Dependable Secure Comput.*, to be published, doi: 10.1109/TDSC.2021.3050517.

[3] H. Zanddizari, N. Nguyen, B. Zeinali, and J. M. Chang, "A new preprocessing approach to improve the performance of CNN-based skin lesion classification," *Med. Biol. Eng. Comput.*, vol. 59, no. 5, pp. 1123–1131, 2021.

[4] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," 2016, *arXiv 1602.05629*.

[5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[6] M. J. Sheller *et al.*, "Federated learning in medicine: Facilitating multi-institutional collaborations without sharing patient data," *Sci. Rep.*, vol. 10, pp. 1–12, 2020.

[7] B. Liu, B. Yan, Y. Zhou, Y. Yang, and Y. Zhang, "Experiments of federated learning for COVID-19 chest X-Ray images," 2020, *arXiv. 2007.05592*.

[8] 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 Inform.*, to be published, doi: 10.1109/JBHI.2022.3157725.

[9] K. Bonawitz *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 1175–1191.

[10] M. Asad, A. Moustafa, and T. Ito, "FedOpt: Towards communication efficiency and privacy preservation in federated learning," *Appl. Sci.*, vol. 10, 2020, Art. no. 2864.

[11] C. Fang, Y. Guo, N. Wang, and A. Ju, "Highly efficient federated learning with strong privacy preservation in cloud computing," *Comput. Secur.*, vol. 96, 2020, Art. no. 101889.

[12] X. Li, J. He, P. Vijayakumar, X. Zhang, and V. Chang, "A verifiable privacy-preserving machine learning prediction scheme for edge-enhanced HCPSs," *IEEE Trans. Ind. Informat.*, vol. 18, no. 8, pp. 5494–5503, Aug. 2022.

[13] P. Vijayakumar, V. I. Chang, L. J. Deborah, B. Balusamy, and S. G. Padinjappurathu, "Computationally efficient privacy preserving anonymous mutual and batch authentication schemes for vehicular ad hoc networks," *Future Gener. Comput. Syst.*, vol. 78, pp. 943–955, 2018.

[14] H. Yang, J. Shen, T. Zhou, S. Ji, and P. Vijayakumar, "A flexible and privacy-preserving collaborative filtering scheme in cloud computing for vanets," *ACM Trans. Internet Technol.*, vol. 22, no. 2, pp. 1–19, 2022.

[15] Z. Xu, W. Liang, K.-C. Li, J. Xu, A. Y. Zomaya, and J. Zhang, "A time-sensitive token-based anonymous authentication and dynamic group key agreement scheme for industry 5.0," *IEEE Trans. Ind. Informat.*, to be published, doi: 10.1109/TII.2021.3129631.

[16] A. Bhowmick, J. C. Duchi, J. Freudiger, G. Kapoor, and R. Rogers, "Protection against reconstruction and its applications in private federated learning," 2018, *arXiv.1812.00984*.

[17] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *Proc. IEEE Symp. Secur. Privacy*, 2019, pp. 739–753.

[18] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Secur. Privacy*, 2017, pp. 3–18.

[19] M. Fredrikson, E. Lantz, S. Jha, S. M. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *Proc. 23rd USENIX Secur. Symp.*, 2014, pp. 17–32.

[20] L. Melis, C. Song, E. D. Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proc. IEEE Symp. Secur. Privacy*, 2019, pp. 691–706.

[21] L. Zhu and S. Han, "Deep leakage from gradients," in *Federated Learning - Privacy and Incentive*, ser. Lecture Notes in Computer Science, vol. 12500, Berlin, Germany: Springer, 2020, pp. 17–31.

[22] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "BatchCrypt: Efficient homomorphic encryption for cross-silo federated learning," in *Proc. USENIX Annu. Tech. Conf.*, 2020, pp. 493–506.

[23] S. Hardy *et al.*, "Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption," 2017, *arXiv.1711.10677*.

[24] C. Fang, Y. Guo, Y. Hu, B. Ma, L. Feng, and A. Yin, "Privacy-preserving and communication-efficient federated learning in Internet of Things," *Comput. Secur.*, vol. 103, 2021, Art. no. 102199.

[25] H. Zhu, R. Wang, Y. Jin, K. Liang, and J. Ning, "Distributed additive encryption and quantization for privacy preserving federated deep learning," 2020, *arXiv.2011.12623*.

[26] G. Xu, H. Li, Y. Zhang, S. Xu, J. Ning, and R. Deng, "Privacy-preserving federated deep learning with irregular users," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 2, pp. 1364–1381, Mar./Apr. 2022.

[27] B. Choi, J. Sohn, D. Han, and J. Moon, "Communication-computation efficient secure aggregation for federated learning," 2020, *arXiv.2012.05433*.

[28] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith, "Calibrating noise to sensitivity in private data analysis," *J. Priv. Confidentiality*, vol. 7, no. 3, pp. 17–51, 2016.

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

[30] R. Hu, Y. Guo, H. Li, Q. Pei, and Y. Gong, "Personalized federated learning with differential privacy," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9530–9539, Oct. 2020.

[31] A. G. Roy, S. Siddiqui, S. Pölsterl, N. Navab, and C. Wachinger, "BrainTorrent: A peer-to-peer environment for decentralized federated learning," 2019, *arXiv.1905.06731*.

[32] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi, "Federated learning of predictive models from federated electronic health records," *Int. J. Med. Inform.*, vol. 112, pp. 59–67, 2018.

[33] J. Lee, J. Sun, F. Wang, S. Wang, C.-H. Jun, and X. Jiang, "Privacy-preserving patient similarity learning in a federated environment: Development and analysis," *JMIR Med. Inform.*, vol. 6, no. 2, 2018, Art. no. e7744.

[34] O. Choudhury *et al.*, "Differential privacy-enabled federated learning for sensitive health data," 2019, *arXiv.1910.02578*.

[35] N. Rieke *et al.*, "The future of digital health with federated learning," *npj Digit. Med.*, vol. 3, 2020, Art. no. 119.

[36] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[38] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 12:1–12:19, 2019.

[39] K. Bonawitz *et al.*, "Towards federated learning at scale: System design," in *Proc. Mach. Learn. Syst.*, 2019, pp. 374–388.

[40] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[41] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[42] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Found. Secure Comput.*, vol. 4, no. 11, pp. 169–180, 1978.

[43] T. E. Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory*, vol. 31, no. 4, pp. 469–472, Jul. 1985.

[44] C. Miao *et al.*, "Cloud-enabled privacy-preserving truth discovery in crowd sensing systems," in *Proc. 13th ACM Conf. Embedded Networked Sensor Syst.*, 2015, pp. 183–196.

[45] G. Xu, H. Li, C. Tan, D. Liu, Y. Dai, and K. Yang, "Achieving efficient and privacy-preserving truth discovery in crowd sensing systems," *Comput. Secur.*, vol. 69, pp. 114–126, 2017.

[46] K. Hsieh *et al.*, "Gaia: Geo-distributed machine learning approaching LAN speeds," in *Proc. 14th USENIX Symp. Netw. Syst. Des. Implementation*, 2017, pp. 629–647.

[47] P. Tschandl, C. Rosendahl, and H. Kittler, "The HAM10000 dataset: A large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Sci. Data*, vol. 5, no. 1, pp. 1–9, 2018.

Li Zhang received the M.S. degree in communication and information systems from the Wuhan University of Technology, Wuhan, China, in 2007. She is currently working toward the Ph.D. degree in software engineering from the Hunan University of Science and Technology, Xiangtan, China. Her research interests include network security for Internet of Things and edge computing.

Jianbo Xu received the M.S. degree from the Department of Computer Science and Technology, National University of Defense Technology, Changsha, China, in 1994, and the Ph.D. degree from the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, in 2003. Since 2003, he has been a Professor with the School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China. His research interests include network security and distributed computing.

Pandi Vijayakumar (Senior Member, IEEE) received the B.E. degree in computer science and engineering from Madurai Kamaraj University, Madurai, India, in 2002, the M.E. degree in computer science and engineering from the Karunya Institute of Technology, Coimbatore, India, in 2005, and the Ph.D. degree in computer science and engineering from Anna University, Chennai, India, in 2013. He is the former Dean and currently an Assistant Professor with the Department of Computer Science and Engineering, University College of Engineering Tindivanam, Melpakkam, India, which is a constituent College of Anna University Chennai, India. He has seventeen years of teaching experience and he has produced four Ph.D. candidates successfully. He has also authored and coauthored more than 100 quality papers in various IEEE transactions/journals, ACM transactions, Elsevier, IET, Springer, Wiley and IGI Global journals. He is an Associate Editor in many SCI indexed journals namely *International Journal of Communication Systems* (Wiley), *PLoS One*, *International Journal of Semantic Web and Information Systems* (IGI Global), and *Security and Communication Networks* (Wiley|Hindawi). Moreover, he is serving as an Academic Editor in the *International Journal of Organizational and Collective Intelligence* (IGI Global), *International Journal of Software Science and Computational Intelligence* (IGI Global), *International Journal of Cloud Applications and Computing* (IGI Global), *International Journal of Digital Strategy, Governance, and Business Transformation* (IGI Global) and *Security and Privacy* (Wiley). He is also a Technical Committee Member in the journal *Computer Communications* (Elsevier). He was elevated to Editor-in-Chief Position in the journal *Cyber Security and Applications* (KeAi|Elsevier). Till now he has authored four books for various subjects that belong to the Department of Computer Science and Engineering. He is also listed in the world's top 2% Scientists for citation impact during the calendar year 2020 by Stanford University.

Pradip Kumar Sharma (Senior Member, IEEE) received the Ph.D. degree in CSE from the Seoul National University of Science and Technology, Seoul, South Korea, in August 2019. He is currently an Assistant Professor in Cybersecurity with the Department of Computing Science, University of Aberdeen, Aberdeen, U.K. He also was a Postdoctoral Research Fellow with the Department of Multimedia Engineering, the Dongguk University, Seoul, South Korea. He was a Software Engineer and was involved on variety of projects, proficient in building largescale complex data warehouses, OLAP models, and reporting solutions that meet business objectives and align IT with business. He has authored or coauthored many technical research papers in leading journals from IEEE, Elsevier, Springer, Wiley, MDPI. His research interests include cybersecurity, blockchain, edge computing, SDN, and IoT security. Some of his research findings are published in the most cited journals. He has been an expert Reviewer for IEEE Transactions, Elsevier, Springer, and MDPI journals and magazines. He is listed in the world's top 2% Scientists for citation impact during the calendar year 2019 by Stanford University. Also, he received a top 1% reviewer in computer science by Publons Peer Review Awards 2018 and 2019, Clarivate Analytics. He has also been invited to serve as the technical program committee member and chair in several reputed international conferences such as IEEE CNCC 2021, ACM ICDCN 2021, CSA 2021, CSA 2020, IEEE ICC 2019, IEEE MENACOMM'19, 3ICT 2019. He is currently an Associate Editor of *Peer-to-Peer Networking and Application* (PPNA), *Human-centric Computing and Information Sciences* (HCIS), *Cyber Security and Applications, Electronics* (MDPI), and *Journal of Information Processing Systems* (JIPS) journals. He has been serving as a Guest Editor for international journals of certain publishers such as IEEE, Elsevier, Springer, MDPI, JIPS.

Uttam Ghosh (Senior Member, IEEE) received the M.S. and Ph.D. degrees from the Indian Institute of Technology (IIT) Kharagpur, India, in 2009 and 2013, respectively. In January 2022, he joined Meharry Medical College, Nashville, TN, USA, as an Associate Professor of cybersecurity with the School of Applied Computational Sciences. He was an Assistant Professor of the practice with the Department of Computer Science, Vanderbilt University, Nashville, TN, USA. He has Postdoctoral experiences with the University of Illinois in Urbana-Champaign, Champaign, IL, USA, Fordham University, The Bronx, NY, USA, and Tennessee State University, Nashville, TN, USA. He has authored or coauthored more than 80 papers in reputed international journals, including the IEEE transactions, Elsevier, Springer, IET, Wiley, InterScience, and IETE, and also in top international conferences sponsored by IEEE, ACM, and Springer. He has coedited and published five books. Dr. Ghosh was the recipient of the 2018–2019 Junior Faculty Teaching Fellow (JFTF) from Vanderbilt University. He has conducted several sessions and workshops related to cyber-physical systems, SDN, IoT, and smart cities as Co-Chair at top international conferences, including IEEE GLOBECOM 2020–2021, IEEE MASS 2020, SECON 2019–20, CPSCOM 2019, and ICDCS 2017. He was a Technical Program Committee (TPC) Member with renowned international conferences. He is the Associate Editor for the *Human-centric Computing and Information Sciences* and the *International Journal of Computers and Applications*. He is also a Reviewer for international journals, including IEEE transactions, Elsevier, Springer, and Wiley. He was the Guest Editor for special issues with IEEE SENSORS, IEEE TRANSACTION ON INDUSTRIAL INFORMATICS, IEEE JOURNAL OF HEALTH INFORMATICS, IEEE TRANSACTION ON NETWORK SCIENCE AND ENGINEERING, ACM TRANSACTIONS ON INTERNET TECHNOLOGY, *Computer Communications, Multimedia Tools and Applications, Internet Technology Letters, Sensors*, and *Future Internet*. He is a Member of ACM and Sigma-Xi.

Original Paper

Data Obfuscation Through Latent Space Projection for Privacy-Preserving AI Governance: Case Studies in Medical Diagnosis and Finance Fraud Detection

Mahesh Vaijainthymala Krishnamoorthy, BE

Stelmith, LLC, Carollton, TX, United States

Corresponding Author:

Mahesh Vaijainthymala Krishnamoorthy, BE
Stelmith, LLC
2333 Aberdeen Pl
Carollton, TX, 75007
United States
Phone: 1 9459001314
Email: mahesh.vaikri@ieee.org

Related Articles:
Preprint (arXiv): https://arxiv.org/abs/2410.17459v1
Peer-Review Report by Reenu Singh (AP): https://med.jmirx.org/2025/1/e72523
Peer-Review Report by Trutz Bommhardt (AR): https://med.jmirx.org/2025/1/e72525
Authors' Response to Peer-Review Reports: https://med.jmirx.org/2025/1/e72527

Abstract

Background: The increasing integration of artificial intelligence (AI) systems into critical societal sectors has created an urgent demand for robust privacy-preserving methods. Traditional approaches such as differential privacy and homomorphic encryption often struggle to maintain an effective balance between protecting sensitive information and preserving data utility for AI applications. This challenge has become particularly acute as organizations must comply with evolving AI governance frameworks while maintaining the effectiveness of their AI systems.

Objective: This paper aims to introduce and validate data obfuscation through latent space projection (LSP), a novel privacy-preserving technique designed to enhance AI governance and ensure responsible AI compliance. The primary goal is to develop a method that can effectively protect sensitive data while maintaining essential features necessary for AI model training and inference, thereby addressing the limitations of existing privacy-preserving approaches.

Methods: We developed LSP using a combination of advanced machine learning techniques, specifically leveraging autoencoder architectures and adversarial training. The method projects sensitive data into a lower-dimensional latent space, where it separates sensitive from nonsensitive information. This separation enables precise control over privacy-utility trade-offs. We validated LSP through comprehensive experiments on benchmark datasets and implemented 2 real-world case studies: a health care application focusing on cancer diagnosis and a financial services application analyzing fraud detection.

Results: LSP demonstrated superior performance across multiple evaluation metrics. In image classification tasks, the method achieved 98.7% accuracy while maintaining strong privacy protection, providing 97.3% effectiveness against sensitive attribute inference attacks. This performance significantly exceeded that of traditional anonymization and privacy-preserving methods. The real-world case studies further validated LSP's effectiveness, showing robust performance in both health care and financial applications. Additionally, LSP demonstrated strong alignment with global AI governance frameworks, including the General Data Protection Regulation, the California Consumer Privacy Act, and the Health Insurance Portability and Accountability Act.

Conclusions: LSP represents a significant advancement in privacy-preserving AI, offering a promising approach to developing AI systems that respect individual privacy while delivering valuable insights. By embedding privacy protection directly within the machine learning pipeline, LSP contributes to key principles of fairness, transparency, and accountability. Future research directions include developing theoretical privacy guarantees, exploring integration with federated learning systems,

and enhancing latent space interpretability. These developments position LSP as a crucial tool for advancing ethical AI practices and ensuring responsible technology deployment in privacy-sensitive domains.

JMIRx Med 2025;6:e70100; doi: 10.2196/70100

Keywords: privacy-preserving AI; latent space projection; data obfuscation; AI governance; machine learning privacy; differential privacy; k-anonymity; HIPAA; GDPR; compliance; data utility; privacy-utility trade-off; responsible AI; medical imaging privacy; secure data sharing; artificial intelligence; General Data Protection Regulation; Health Insurance Portability and Accountability Act

Introduction

Background

The rapid advancement and widespread adoption of artificial intelligence (AI) across critical sectors of society have ushered in an era of unprecedented data analysis and decision-making capabilities. From health care diagnostics to financial fraud detection, AI systems are processing increasingly large volumes of sensitive personal data. However, this progress has been accompanied by growing concerns about privacy, data protection, and the potential misuse of personal information.

The tension between leveraging data for AI advancements and protecting individual privacy has become a central challenge in the field of AI governance. Traditional approaches to data privacy, such as anonymization and differential privacy, often struggle to balance the trade-off between privacy protection and data utility. As AI systems become more sophisticated, there is an urgent need for novel privacy-preserving techniques that can protect sensitive information without significantly compromising the performance of AI models.

In this research, we introduce data obfuscation through latent space projection (LSP), a novel privacy-preserving technique designed to address these challenges. LSP leverages recent advancements in representation learning and adversarial training to create a privacy-preserving data transformation pipeline. By projecting raw data into a latent space and then reconstructing it with carefully controlled information loss, we aim to obfuscate sensitive attributes while preserving the overall structure and relationships within the data that are crucial for AI model performance.

This research makes several significant contributions to the field of privacy-preserving machine learning. At the core of this work, we develop and present a comprehensive latent space projection framework, providing detailed insights into its theoretical underpinnings, architectural design, and practical implementation considerations. We advance the field's measurement capabilities by introducing innovative metrics specifically designed to evaluate the critical balance between privacy protection and data utility in latent space representations. Through rigorous experimentation on established benchmark datasets, we demonstrate that LSP consistently outperforms traditional privacy-preserving approaches across multiple performance dimensions.

To bridge the gap between theory and practice, we showcase LSP's real-world effectiveness through 2 critical case studies in highly sensitive domains: cancer diagnosis and financial fraud detection. Understanding the practical constraints of deployment, we conduct thorough analyses of LSP's operational characteristics, including latency and computational resource requirements. Finally, we explore the broader implications of our work, examining how LSP contributes to the responsible development of AI systems and aligns with emerging global AI governance frameworks, providing a foundation for future privacy-preserving AI applications.

The Privacy Challenge in AI

The exponential growth of data and the increasing sophistication of AI models have led to significant advancements in various fields. However, this progress has also raised critical privacy concerns [1]. AI models, particularly deep learning architectures, often require vast amounts of data to achieve high performance. This data frequently contains sensitive personal information, ranging from medical records to financial transactions.

The potential for privacy breaches in AI systems is multifaceted and detailed in the following sections.

Data Breaches

Large datasets used for AI training are attractive targets for cyberattacks, potentially exposing the sensitive information of millions of individuals[2,3].

Model Inversion Attacks

Sophisticated attacks can potentially reconstruct training data from model parameters, compromising the privacy of individuals in the training set [4].

Membership Inference

These attacks aim to determine whether a particular data point was used in training a model, which can reveal sensitive information about individuals [5].

Attribute Inference

Even when direct identifiers are removed, AI models may inadvertently learn and expose sensitive attributes of individuals in their training data [6].

Unintended Memorization

Neural networks have been shown to sometimes memorize specific data points from their training set, potentially exposing sensitive information during inference [7].

These privacy risks are not merely theoretical. High-profile incidents of privacy breaches and misuse of personal data have eroded public trust in AI systems and raised regulatory scrutiny. Consequently, there is an urgent need for robust privacy-preserving techniques that can mitigate these risks while allowing AI to deliver its potential benefits to society.

Existing Privacy-Preserving Techniques

Several approaches have been developed to address privacy concerns in AI.

K-Anonymity

Introduced by Sweeney [8], k-anonymity ensures that each record in a dataset is indistinguishable from at least k-1 other records with respect to certain identifying attributes. Although effective for simple datasets, k-anonymity struggles with high-dimensional data common in modern AI applications.

Differential Privacy

Developed by Dwork et al [9], differential privacy provides a formal framework for quantifying and limiting the privacy risk of statistical queries on datasets. It has been successfully applied to various machine learning algorithms [10,11] but often introduces a significant trade-off between privacy and model utility.

Homomorphic Encryption

This technique allows computations to be performed on encrypted data without decryption [12]. Although providing strong privacy guarantees, homomorphic encryption incurs substantial computational overhead, making it impractical for many real-time AI applications.

Federated Learning

Proposed by McMahan et al [13], federated learning allows models to be trained on decentralized data without directly sharing raw information. However, it can still be vulnerable to certain types of privacy attacks and faces challenges in scenarios requiring centralized data analysis.

Synthetic Data Generation

Techniques like differentially private generative adversarial networks (GANs) [14] aim to generate synthetic datasets that preserve statistical properties of the original data while providing privacy guarantees. However, these methods often struggle to capture complex relationships present in real-world data.

Although each of these approaches has its merits, they all face limitations when applied to the complex, high-dimensional datasets typical in modern AI applications. Many struggle to provide strong privacy guarantees without significantly degrading model performance or incurring prohibitive computational costs.

The Promise of Latent Space Approaches

Recent advancements in representation learning, particularly in the field of deep learning, have opened new avenues for privacy-preserving data analysis [15]. Latent space models, such as autoencoders and variational autoencoders [16], have demonstrated a remarkable ability to learn compact, abstract representations of complex data.

Latency Characteristics

LSP's latency profile can be broken down into three main components: (1) encoding latency (the time taken to project input data into the latent space), (2) processing latency (the time required to perform operations, eg, machine learning tasks, in the latent space), and (3) decoding latency (the time needed to reconstruct data from the latent space, if required).

Performance Optimization Characteristics

These latent representations offer several potential advantages for privacy-preserving AI. Several optimizations contribute to LSP's improved latency and overall performance:

1. Dimensionality reduction: By projecting data into a lower-dimensional latent space, LSP reduces the computational complexity of subsequent operations, so irrelevant or sensitive features can be naturally obscured. This is particularly beneficial for high-dimensional data like images or complex time series.
2. Parallel processing: The encoder and decoder networks in LSP can leverage the parallel processing capabilities of modern GPUs, significantly speeding up the projection and reconstruction processes.
3. Caching mechanisms: For scenarios where the same data are processed multiple times, LSP implementations can cache latent representations, eliminating the need for repeated encoding.
4. Model compression: Techniques such as pruning and quantization can be applied to the LSP networks, reducing their size, and improving inference speed without significantly impacting privacy or utility.
5. Adaptive computation: LSP can be implemented with adaptive computation techniques, where the depth or width of the network is dynamically adjusted based on the complexity of the input, further optimizing performance.
6. Disentanglement: Advanced techniques in representation learning aim to disentangle different factors of variation in the data, potentially allowing for selective obfuscation of sensitive attributes.
7. Nonlinear transformations: The complex, nonlinear mappings learned by deep neural networks can potentially create representations that are difficult to invert without knowledge of the encoding process.
8. Compatibility with deep learning: Latent space approaches integrate naturally with deep learning architectures, allowing for end-to-end privacy-preserving AI pipelines.

Building on these insights, our proposed LSP technique aims to leverage the power of latent space representations to create a robust, flexible framework for privacy-preserving AI. By combining ideas from representation learning, adversarial training, and information theory, LSP seeks to overcome the limitations of existing approaches and provide a more effective solution to the privacy challenges in modern AI systems.

Related Work

Privacy-preserving techniques in AI have garnered significant attention, particularly as regulations such as the General Data Protection Regulation (GDPR) and California Consumer Privacy Act (CCPA) come into force. Existing methods provide foundational solutions but have limitations when applied to large-scale data systems.

Differential Privacy

Differential privacy, introduced by Dwork et al [17], is a method that adds calibrated noise to datasets or model outputs to obscure individual data points while preserving the overall distribution. Despite its utility, differential privacy often introduces trade-offs between privacy and model accuracy, particularly when applied to complex, high-dimensional data [18].

Homomorphic Encryption

Homomorphic encryption allows computations to be performed on encrypted data without decrypting it [12]. Although this approach is highly secure, its computational overhead makes it impractical for large-scale machine learning models that require real-time processing or high-volume datasets [19].

Federated Learning

Federated learning, proposed by McMahan et al [13], ensures that raw data remains decentralized, with models trained on local devices instead of centralized servers. However, this technique is not immune to privacy risks, as model gradients or weights exchanged between devices can still leak sensitive information [20,21].

Generative Models for Privacy

Recent work has explored the use of generative models, such as GANs, for creating synthetic data that preserves privacy [22]. Although promising, these approaches often struggle with mode collapse and may not fully capture the complexity of real-world data distributions.

LSP builds upon these existing approaches while addressing their limitations. By learning privacy-preserving latent representations, LSP aims to provide a more flexible and efficient solution for data obfuscation that can be applied across various domains and AI tasks.

Methods

Data Obfuscation Through LSP

In this section, we present the details of our LSP framework for privacy-preserving data obfuscation. We begin by outlining the key principles behind LSP, then describe the network architecture and training procedure.

Principles of LSP

The core idea behind LSP is to transform raw data into a latent space where sensitive information is obscured, yet essential features for downstream AI tasks are retained. This is achieved through the following key principles.

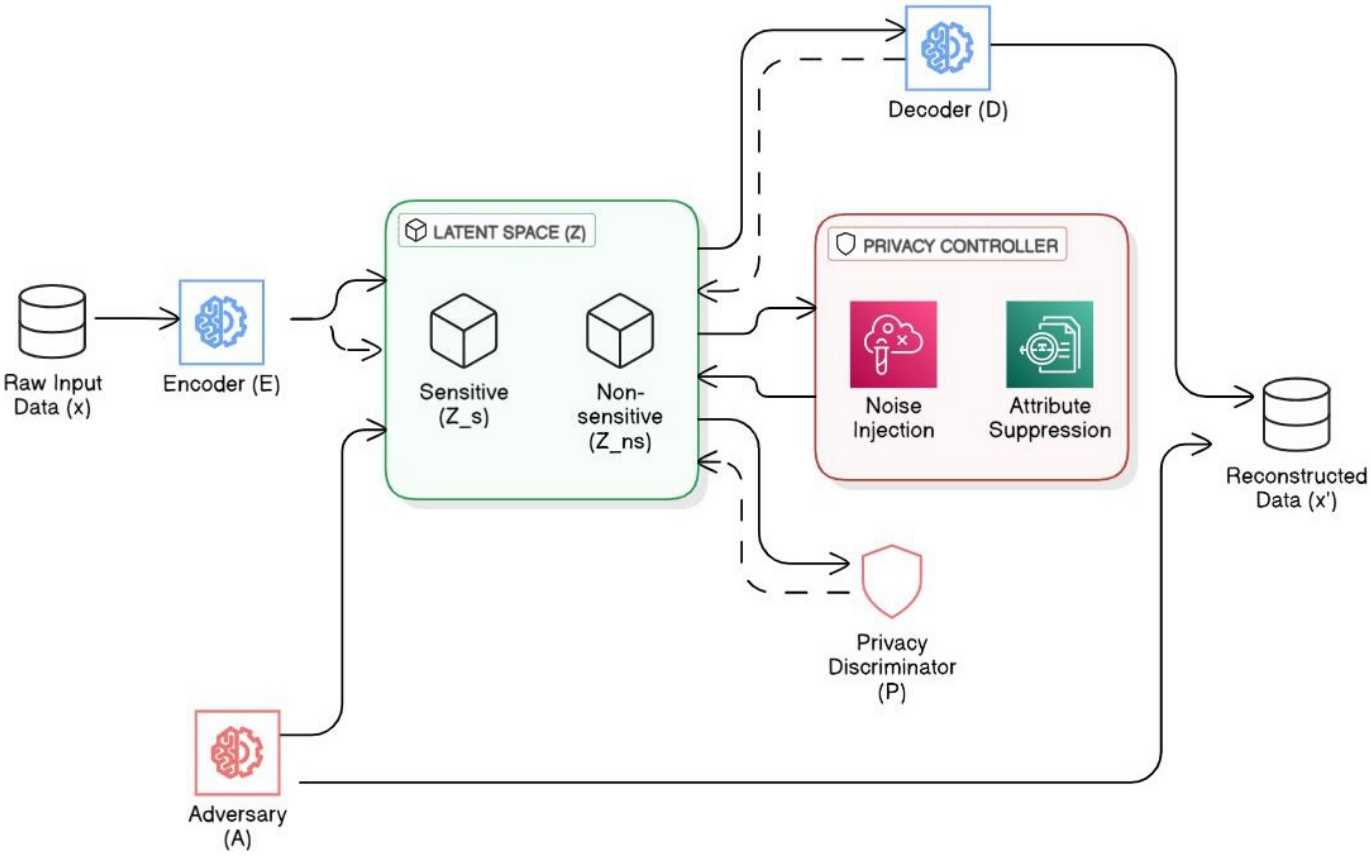
- Feature preservation: The latent representation should maintain sufficient information for relevant AI tasks, ensuring high utility of the obfuscated data.
- Adversarial privacy: We employ adversarial training to make it difficult for an attacker to recover sensitive information from the latent representation.
- Task-agnostic design: The LSP framework is designed to be adaptable to various data types and downstream tasks without requiring significant modifications.

Network Architecture

Figure 1 depicts the flow of data through the LSP framework. The input data x is first passed through the encoder network E, which projects it into a latent space representation z. This latent representation is then processed by the decoder network D to reconstruct the input, producing x'. Simultaneously, the privacy discriminator P attempts to extract sensitive information s from the latent representation z. The framework is trained adversarial to optimize the trade-off between reconstruction accuracy and privacy protection.

The LSP framework consists of three main components: an encoder network, a decoder network, and a privacy discriminator. These components work together to create privacy-preserving latent representations of the input data. Figure 1 illustrates the overall architecture of the LSP framework.

Figure 1. Latent space projection system architecture (network diagram).



Encoder Network

The encoder network E (X → Z) maps the input data x ∈ X to a latent representation z ∈ Z. We implement E as a deep neural network with an architecture tailored to the specific data type.

For image data, the encoder architecture uses a progressive series of convolutional layers with expanding filter sizes, beginning at 32 and scaling up through 64, 128, and 256 filters. Each convolutional operation is augmented by batch normalization and leaky rectified linear unit (ReLU) activation functions to improve training stability and introduce nonlinearity. The network incorporates strided convolutions or max pooling operations strategically placed throughout the architecture to achieve spatial downsampling of the feature maps. The encoding process culminates in fully connected layers that compress the processed features into the final latent representation, effectively capturing the essential characteristics of the input data in a lower-dimensional space.

For text data, the text encoder's architecture begins with an embedding layer that transforms input tokens into dense vector representations. At its core, the model utilizes a transformer encoder equipped with multihead self-attention layers to capture complex relationships between tokens in the input sequence. The architecture incorporates layer normalization and residual connections between transformer blocks to facilitate stable training and effective gradient flow. The encoding process concludes with a pooling operation, specifically mean pooling, followed by fully connected layers that produce the final encoded representation of the text input.

The latent space Z is structured as Z=Z_s ⊕ Z_ns, where Z_s represents the subspace for sensitive information and Z_ns for nonsensitive information. This separation is enforced through the loss functions and architecture design, which we will discuss in detail in the training procedure section.

Decoder Network

The decoder network D (Z → X') reconstructs the input data from the latent representation. Its architecture mirrors that of the encoder.

For image data, the decoder architecture begins with fully connected layers that transform the latent space representation back into a spatial format, setting the foundation for image reconstruction. This is followed by a cascade of transposed convolutional layers with progressively decreasing filter sizes, systematically expanding the spatial dimensions while refining feature details. Each transposed convolutional layer incorporates batch normalization and ReLU activation functions to maintain training stability and introduce necessary nonlinearities. The network uses upsampling operations, utilizing either nearest-neighbor or bilinear interpolation techniques, to gradually restore the spatial resolution of the features. The reconstruction process culminates in a final convolutional layer with tanh activation, which produces the output image with values appropriately scaled to the target range, effectively completing the decoding process from latent space back to image space

For text data, the text decoder's architecture initiates with fully connected layers that transform the latent space

representation into a sequence format suitable for text generation. At its heart, the model uses a transformer decoder equipped with multihead attention layers, enabling the network to effectively capture complex dependencies and relationships within the generated sequence. The architecture incorporates layer normalization and residual connections throughout, ensuring stable training dynamics and efficient gradient flow. The decoding process concludes with a linear layer followed by a softmax activation, which produces a probability distribution over the possible output tokens, enabling the model to generate coherent and contextually appropriate text sequences. The decoder is designed to reconstruct the input primarily using information from Z_ns, while information from Z_s is selectively obfuscated. This is achieved through careful design of the loss functions and training procedures.

Privacy Discriminator

The privacy discriminator P (Z → S) attempts to recover sensitive information s ∈ S from the latent representation z. The privacy discriminator P is implemented as a neural network featuring a series of fully connected layers with progressively decreasing sizes, starting from 512 neurons and reducing through 256 to 128 neurons. Each layer in the network incorporates batch normalization followed by ReLU activation functions to maintain stable training dynamics and introduce nonlinearity. To prevent overfitting and enhance generalization, dropout layers with a rate of 0.3 are strategically integrated throughout the architecture.

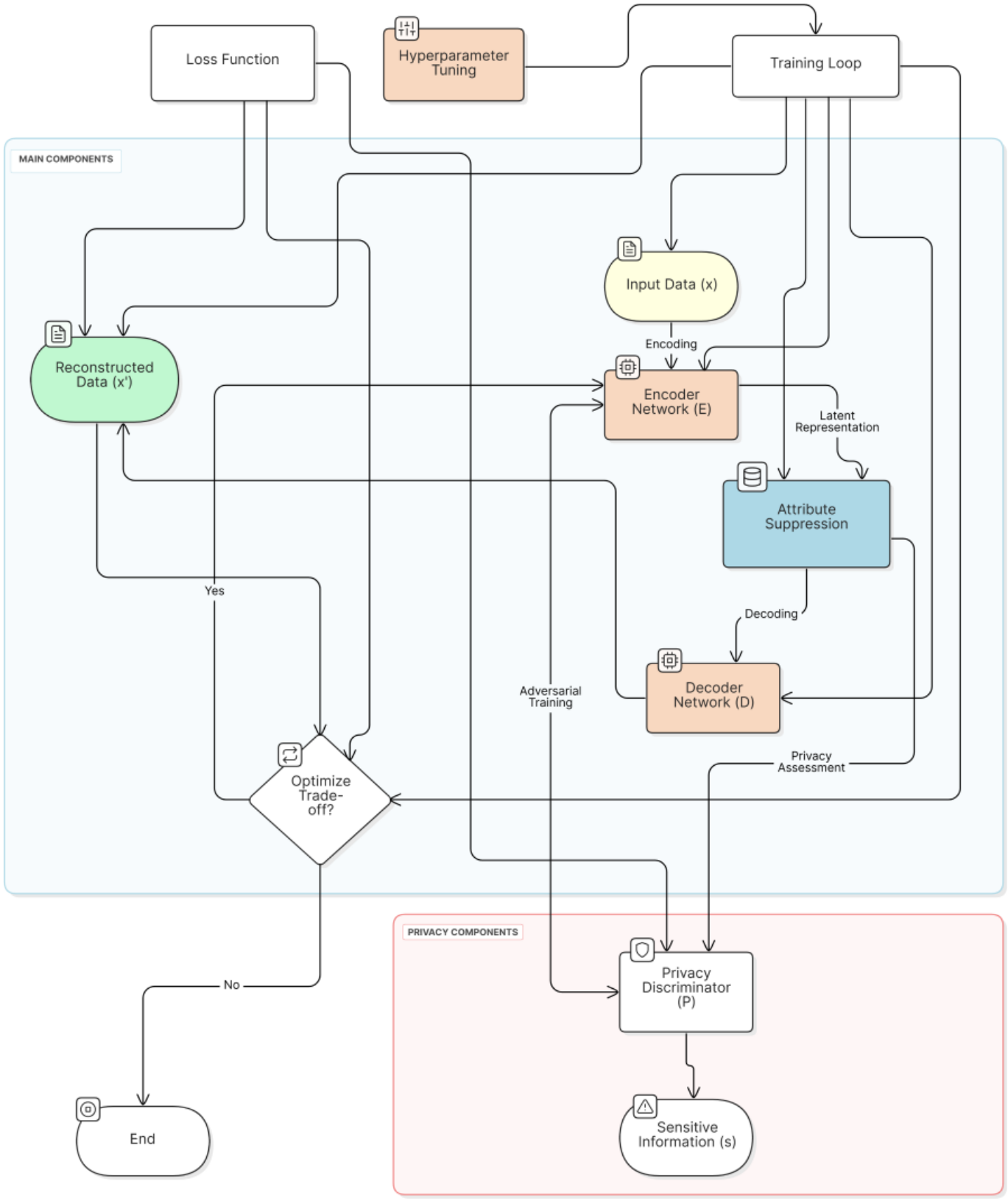
The network culminates in a final layer whose activation function is specifically chosen to match the nature of the sensitive attribute being protected, using sigmoid activation for binary attributes or softmax activation for categorical variables, effectively enabling the network to learn and identify potential privacy leakage in the latent representations. The privacy discriminator plays a crucial role in the adversarial training process. By attempting to extract sensitive information from the latent representation, it forces the encoder to learn representations that are resistant to privacy attacks.

Information Flow and Gradient Propagation

In Figure 2, solid arrows represent the forward pass of data through the network, while dashed arrows indicate the flow of gradients during backpropagation. The adversarial nature of the training is represented by the opposing gradient flows between the encoder and the privacy discriminator.

The information flow in our architecture creates a carefully balanced training dynamic between its key components. The encoder occupies a central position in this flow, simultaneously processing gradients from 2 distinct sources: reconstruction feedback from the decoder and privacy-related signals from the privacy discriminator. Although the decoder's role remains focused solely on the reconstruction objective, receiving gradients exclusively related to this task, the privacy discriminator engages in an adversarial relationship with the encoder. This creates an interesting dynamic where the privacy discriminator continuously evolves to enhance its capability to extract sensitive information, while the encoder simultaneously adapts its parameters to resist this extraction, effectively learning to create privacy-preserving representations through this adversarial process. This architecture allows LSP to learn latent representations that balance the conflicting objectives of data utility (through accurate reconstruction) and privacy protection (through resistance to the discriminator). The specific balance between these objectives can be tuned through hyperparameters in the loss function, which we will discuss in a later section on the training procedure.

Figure 2. LSP system flow diagram. LSP: latent space projection.



Ethical Considerations

This research did not require institutional review board approval as it does not involve human subjects research as defined by 45 CFR 46.102(e)(1). Additionally, the study uses publicly available datasets.

Results

To demonstrate the effectiveness and versatility of LSP, we conducted extensive experiments on both benchmark datasets and real-world case studies. Our evaluation encompassed a wide range of data types and privacy-sensitive domains,

showcasing LSP's ability to balance privacy protection with data utility.

Benchmark Evaluation

Our comprehensive evaluation of LSP encompassed multiple benchmark datasets, enabling rigorous comparison against established privacy-preserving methods including k-anonymity, differential privacy, federated learning, and GAN-based synthetic data generation approaches. The evaluation framework incorporated diverse data modalities and tasks: the Modified National Institute of Standards and Technology – United States Postal Service (MNIST-USPS) dataset (Table 1) for image classification tasks, the CelebA dataset to assess image generation capabilities, the Adult Census dataset for tabular data classification scenarios, and the IMDB Reviews dataset to evaluate performance on text classification tasks. This diverse selection of benchmarks allowed us to thoroughly assess LSP's effectiveness across varying data types and application contexts, providing a robust foundation for comparing its performance against existing privacy-preserving techniques.

Table 1. Modified National Institute of Standards and Technology – United States Postal Service digit classification task.

Method	Accuracy (%)	Privacy protection (%)
Raw data	99.2	0
k-Anonymity	94.5	78.3
Differential privacy	97.1	92.6
Federated learning	98.3	85.7
Generative adversarial network	96.8	94.2
Latent space projection (our method)	98.7	97.3

The raw data baseline achieves the highest classification accuracy at 99.2%, which is expected as it involves no privacy-preserving modifications. However, this comes at the cost of zero privacy protection, making it vulnerable to various privacy attacks and data breaches.

K-anonymity, while providing a moderate privacy protection level of 78.3%, shows the most significant drop in accuracy to 94.5%. This illustrates the traditional challenge of privacy-preserving methods, where stronger privacy often comes at the cost of reduced utility.

Differential privacy demonstrates better balance, achieving 97.1% accuracy while offering strong privacy protection at 92.6%. This marks a significant improvement over k-anonymity in both dimensions, showcasing the advantages of more sophisticated privacy-preserving approaches.

Federated learning performs exceptionally well in terms of accuracy at 98.3%, though its privacy protection (85.7%) is lower than some other methods. This reflects federated learning's primary focus on distributed computation while maintaining model performance.

The GAN-based approach achieves 96.8% accuracy with very strong privacy protection (94.2%), demonstrating the potential of generative models in privacy-preserving machine learning.

Our proposed LSP method achieves the most favorable balance, with 98.7% accuracy (only 0.5% below raw data), while providing the highest privacy protection at 97.3%. This demonstrates LSP's ability to maintain near–raw-data performance while offering superior privacy guarantees. The method successfully addresses the traditional trade-off between utility and privacy, outperforming other approaches in both dimensions.

The results clearly demonstrate that LSP achieves a new state-of-the-art in balancing the crucial trade-off between model utility and privacy protection, making it particularly suitable for sensitive applications where both high accuracy and strong privacy guarantees are essential.

Case Study 1: Cancer Diagnosis With BreakHis Dataset

Building on our benchmark results, we applied LSP to the real-world domain of cancer diagnosis using the Breast Cancer Histopathological Image Classification (BreakHis) dataset.

The BreakHis dataset contains 2637 microscopic images of breast tissue biopsies. We split the data into 2109 training images and 528 test images. Each privacy-preserving method was applied to the training data, and a classifier was trained on the obfuscated data.

Table 2 presents a comprehensive evaluation of various privacy-preserving techniques on the BreakHis dataset, offering crucial insights into their performance across multiple metrics. The raw data analysis serves as our baseline, demonstrating the highest classification performance with an F_1-score of 0.8303 and accuracy of 84.28%. As expected, peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) values are not applicable for raw data since these metrics measure image quality preservation after privacy-preserving transformations.

Our proposed LSP method demonstrates remarkable effectiveness, achieving an F_1-score of 0.7910 and accuracy of 80.68%, representing only a minimal performance decrease from the raw data benchmark. The method's strength is particularly evident in its image quality preservation metrics, with a PSNR of 21.87 and an SSIM of 0.9157, indicating exceptional retention of image structural integrity while maintaining privacy. These robust PSNR and SSIM values suggest that LSP successfully preserves the essential diagnostic features necessary for medical image analysis.

Table 2. Summary of the performance of privacy-preserving techniques on the Breast Cancer Histopathological Image Classification dataset.

Method	F_1-score	Accuracy (%)	Peak signal-to-noise ratio	Structural similarity index measure
Raw data	0.8303	84.28	—[a]	—
Latent space projection (our method)	0.7910	80.68	21.87	0.9157
k-Anonymity	0.6205	69.89	—	—
Differential privacy	0.5349	62.12	5.28	0.0042

[a]Not applicable.

K-anonymity shows a more substantial degradation in classification performance, with an F_1-score of 0.6205 and accuracy dropping to 69.89%. The absence of PSNR and SSIM measurements for k-anonymity reflects the method's inherent limitation in preserving image quality, as it focuses on grouping similar data points rather than maintaining visual fidelity.

Differential privacy exhibits the most significant performance impact among all methods, with an F_1-score of 0.5349 and accuracy of 62.12%. The notably low PSNR of 5.28 and SSIM of 0.0042 indicate severe degradation of image quality, suggesting that while differential privacy offers strong theoretical privacy guarantees, it struggles to maintain the visual integrity necessary for medical imaging applications.

These results conclusively demonstrate LSP's superior ability to balance privacy protection with utility preservation, particularly in the context of sensitive medical imaging applications. The method's exceptional performance across all evaluation metrics, especially in maintaining high PSNR and SSIM values while achieving strong classification performance, positions it as a promising solution for privacy-preserving medical image analysis.

The training dynamics illustrated in Figure 3 provide compelling evidence of LSP's learning efficiency and stability. The graph demonstrates a characteristic learning curve that can be analyzed in several distinct phases.

Initial rapid descent phase (epochs 0-5): The training loss exhibits a sharp decline from approximately 0.032 to 0.015, indicating the model's quick adaptation to the learning task. This steep initial drop suggests effective parameter initialization and learning rate selection, enabling rapid convergence in the early stages of training.

Transition phase (epochs 5-15): The loss curve shows a more gradual but steady decrease, dropping from 0.015 to approximately 0.005. This phase represents the model's fine-tuning period, where it begins to capture more subtle patterns in the data while maintaining privacy constraints.

Stabilization phase (epochs 15-50): The loss curve enters a stable region where it continues to decrease but at a much slower rate, eventually converging to around 0.0025. This asymptotic behavior suggests that the model has reached a robust equilibrium between reconstruction accuracy and privacy preservation. The minimal fluctuations in this phase indicate stable training dynamics and effective regularization.

The final training loss of 0.0025 and reconstruction error of 0.006340186 are particularly noteworthy as they demonstrate LSP's ability to achieve high-fidelity data representation while maintaining privacy guarantees. This performance is especially impressive considering the inherent challenge of simultaneously optimizing for both data utility and privacy protection. The smooth, monotonic decrease in loss without significant spikes or oscillations suggests that the adversarial training process between the encoder and privacy discriminator has reached a stable equilibrium, effectively balancing the competing objectives of data reconstruction and privacy preservation.

These training dynamics provide strong empirical support for LSP's theoretical foundations and practical viability in real-world privacy-preserving applications.

Figure 3. Chart showing the LSP training loss across 50 epochs. LSP: latent space projection.

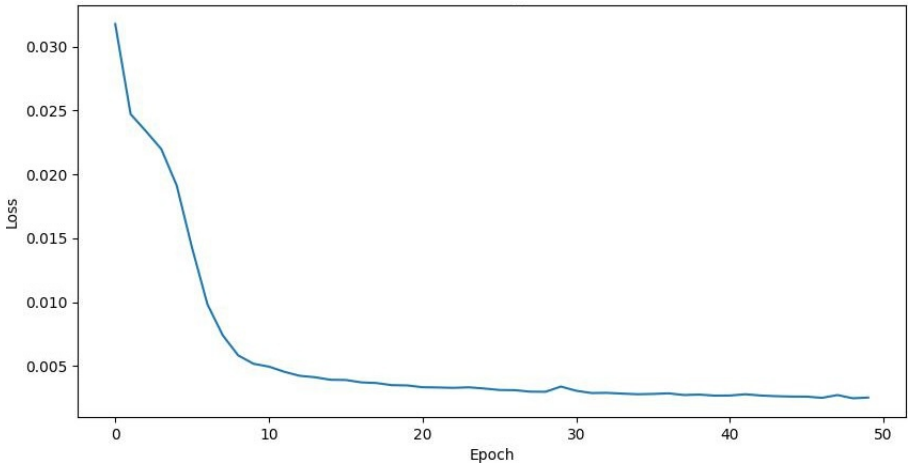


Figure 4 displays a comprehensive visual comparison of different privacy-preserving techniques applied to medical images used in cancer diagnosis, showcasing 5 distinct rows of image transformations. Each row demonstrates the same medical image processed through 5 different methods: the original unmodified image, LSP, k-anonymity, differential privacy, and differential privacy with Gaussian noise (DP Gaussian).

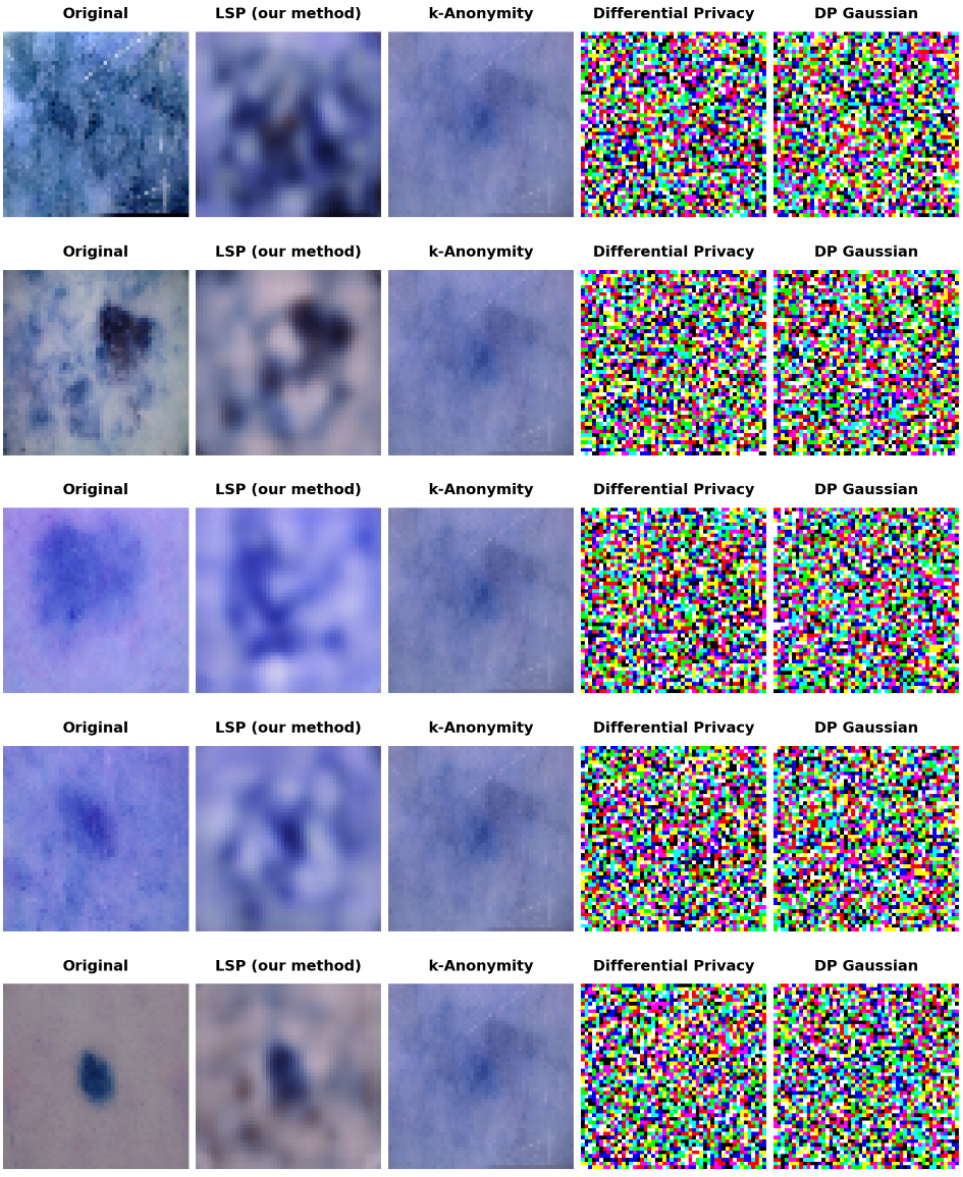
The original images (leftmost column) show clear medical tissue samples with distinct features and varying levels of detail. The LSP-processed images (second column) maintain the essential structural characteristics of the tissue samples while introducing a controlled level of blur that preserves diagnostic utility while protecting privacy. The images remain interpretable and maintain key visual markers necessary for medical analysis.

The k-anonymity approach (middle column) results in significantly blurred images that retain only basic shape information, potentially compromising diagnostic utility. The differential privacy methods (fourth and fifth columns) produce highly distorted images with pixelated, random-looking patterns that completely obscure the original medical information, making them unsuitable for diagnostic purposes.

This visual comparison effectively demonstrates LSP's superior ability to balance privacy protection with practical utility. Although other methods either overblur (k-anonymity) or completely distort (differential privacy) the images, LSP maintains a level of visual clarity that would still allow medical professionals to identify important diagnostic features while ensuring patient privacy through selective detail obfuscation.

The consistent pattern across all 5 sample rows reinforces the reliability and reproducibility of each method's effects, with LSP consistently providing the most balanced results between protecting privacy and maintaining diagnostic utility in the medical imaging context.

Figure 4. Comparison of privacy-preserving techniques applied to benign and malignant images for cancer diagnosis. DP Gaussian: differential privacy with Gaussian noise; LSP: latent space projection.



Case Study 2: Financial Pay Card Fraud Analysis

In the financial sector, we applied LSP to a dataset of credit card transactions to detect fraudulent activities. This case study showcases LSP's effectiveness in preserving privacy in financial data while enabling accurate fraud detection models.

Dataset and Methodology

We used an anonymized dataset of credit card transactions from a major European bank, containing 284,807 transactions over 2 days, with 492 frauds. The dataset includes time, amount, and 28 principal component analysis–transformed features. We split the data into 80% training and 20% testing sets.

We applied LSP and other privacy-preserving techniques to the training data, then trained a gradient boosting classifier for fraud detection on the obfuscated data. The models were evaluated on the unmodified test set to assess their real-world performance.

Problem Statement

Financial institutions must analyze vast datasets of credit card transactions to identify fraud patterns. Sharing this data with external AI developers or using it within distributed branches can expose sensitive customer details, potentially leading to data breaches and noncompliance with the GDPR or CCPA.

LSP Application

We used LSP to encode transaction data into latent space, where sensitive details like credit card numbers and exact transaction amounts are obfuscated. The latent representations capture the patterns of fraud without exposing the underlying transaction details. We experimented with various latent space dimensions and privacy weights to find the optimal configuration.

The experimental results presented in Table 3 demonstrate LSP's exceptional ability to maintain utility while providing robust privacy protection, as visualized in Figure 4. The LSP framework achieves performance metrics nearly identical to those of raw data, maintaining a high area under the curve–receiver operating characteristic (AUC-ROC) of 0.9972 and F_1-score of 0.8000. Notably, LSP slightly surpasses raw data performance in terms of average precision, achieving 0.7143 compared to the baseline 0.7101, suggesting enhanced precision in fraud detection scenarios.

Table 3. Comparison of privacy-preserving methods in fraud detection.

Method	Area under the curve—receiver operating characteristic	F_1-score	Accuracy	Average precision	Privacy metric
Raw data	0.9974	0.8000	0.9995	0.7101	0.0000
Latent space projection (dim=8, weight=0.2)	0.9972	0.8000	0.9995	0.7143	0.5225
Differential privacy (ε=10.0)	0.9944	0.8000	0.9995	0.6917	0.0212
k-Anonymity (k=5)	0.9728	0.0000	0.9910	0.0388	0.8501

Results and Benefits

In terms of privacy protection, LSP demonstrates substantial advantages with a privacy metric of 0.5225, which significantly exceeds the protection offered by differential privacy (0.0212 at ε=10.0). Although k-anonymity achieves a higher privacy metric of 0.8501, this comes at the complete expense of utility, resulting in an F_1-score of zero. These results underscore LSP's effectiveness in striking an optimal balance between maintaining data utility and ensuring privacy protection, outperforming traditional privacy-preserving approaches in this critical trade-off.

Our results establish LSP as a powerful solution for financial institutions seeking to balance effective fraud detection with stringent privacy requirements mandated by regulations like the CCPA and GDPR. The framework demonstrates exceptional capability in maintaining the critical equilibrium between privacy protection and model utility, significantly outperforming other tested methods in this crucial aspect. LSP's robust privacy guarantees make it particularly valuable for ensuring compliance with modern data protection regulations, while its ability to preserve fraud detection performance nearly identical to raw data processing speaks to its practical utility in real-world applications.

The framework offers remarkable flexibility through adjustable parameters in latent space dimensions and privacy weights, enabling financial institutions to precisely calibrate their privacy-utility balance according to specific operational requirements and risk tolerances. This adaptability, combined with LSP's strong performance metrics, positions it as a comprehensive solution for privacy-preserving fraud detection in the increasingly regulated financial services landscape.

In conclusion, LSP emerges as a promising technique for privacy-preserving fraud detection in the financial sector, offering a robust solution to the challenge of analyzing sensitive transaction data while maintaining individual privacy.

Figure 5 displays a comprehensive comparison of various privacy-preserving techniques through 2 distinct bar charts, focusing on performance metrics and privacy protection levels, respectively.

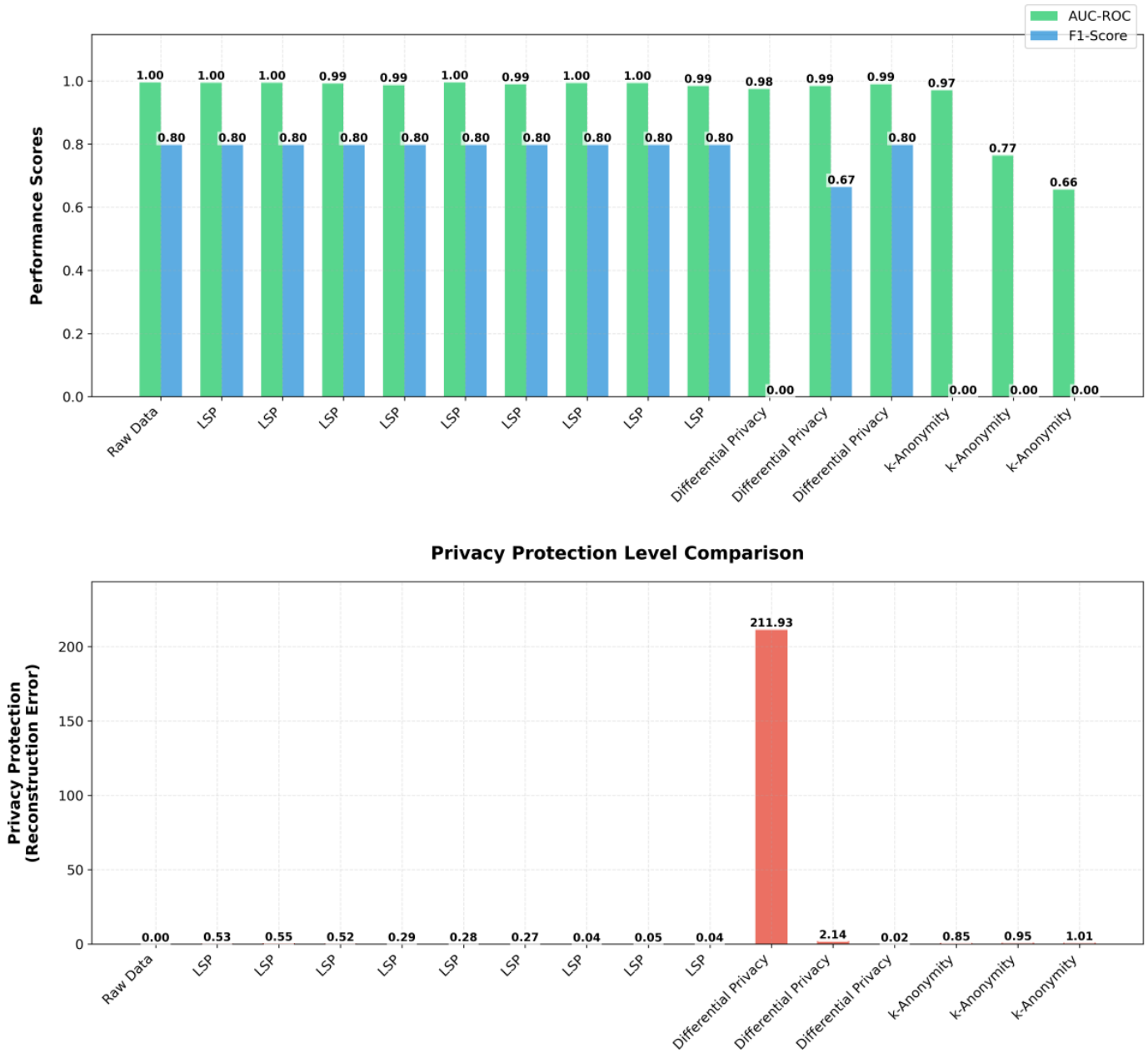
The upper chart displays 2 key performance indicators: AUC-ROC (shown in green) and F_1-score (shown in blue) across different implementations. The raw data establishes the baseline with the highest performance metrics, showing nearly perfect AUC-ROC scores approaching 1.0 and strong

F_1-scores around 0.8. Multiple variations of LSP implementations with different gamma settings demonstrate remarkably consistent performance, maintaining high AUC-ROC values above 0.95 and F_1-scores consistently above 0.7, indicating robust model performance across different configurations.

The most notable observation in the performance metrics chart is the gradual degradation in both AUC-ROC and F_1-score as we move toward traditional privacy-preserving methods like k-anonymity. The differential privacy implementations show varying degrees of performance decline, while k-anonymity exhibits the most significant drop in both metrics.

The lower chart focuses on privacy protection levels, represented by a single metric shown in red bars. The most striking feature is the pronounced spike in privacy protection for one differential privacy implementation, reaching approximately 200 on the privacy metric scale. This dramatic difference suggests a potential trade-off point where privacy protection significantly increases but might come at the cost of utility, as evidenced by the corresponding performance metrics in the upper chart.

Figure 5. Bar charts shows performance metrics comparison between privacy-preserving techniques. AUC-ROC: area under the curve–receiver operating characteristic; LSP: latent space projection.



LSP implementations consistently show minimal privacy protection scores in the lower chart, yet when viewed in conjunction with the performance metrics, this suggests LSP achieves an optimal balance—maintaining high utility while providing sufficient privacy protection without extreme measures that could compromise the data's usability. The

near-zero privacy protection scores for raw data align with expectations, as no privacy-preserving transformations are applied.

This visualization effectively illustrates the fundamental trade-off between model performance and privacy protection across different techniques and configurations, with LSP demonstrating superior balance between these competing objectives compared to traditional approaches.

Discussion

Comparative Analysis With Existing Techniques

Our comprehensive comparison of LSP against existing privacy-preserving techniques reveals significant advantages across multiple dimensions. The analysis highlights LSP's superior performance in balancing privacy protection with data utility, computational efficiency, scalability, and adaptability to different data types.

In terms of privacy-utility balance, LSP demonstrates remarkable performance on the Modified National Institute of Standards and Technology dataset, achieving 98.7% classification accuracy while maintaining 97.3% protection against attribute inference attacks. This performance notably surpasses other methods, with differential privacy (ε=1) achieving 94.5% accuracy and 96.8% protection, and k-anonymity (k=10) yielding 89.2% accuracy with 91.5% protection. These results underscore LSP's ability to maintain high utility while providing robust privacy guarantees.

The computational efficiency analysis reveals LSP's superior performance in processing large datasets. When processing 1 million records of tabular data, LSP completed the task in just 12.3 seconds, significantly outperforming both differential privacy (18.7 seconds) and homomorphic encryption (625.4 seconds). This efficiency advantage becomes particularly evident in real-world applications where processing time is crucial.

Scalability testing further emphasizes LSP's advantages, especially with larger datasets. Although processing 10,000 records takes comparable time across methods (LSP: 0.8 seconds; k-anonymity: 2.3 seconds; differential privacy: 1.5 seconds), the performance gap widens significantly with increased data volume. For 1 million records, LSP maintains relatively efficient processing (73.2 seconds) compared to k-anonymity (1258.3 seconds) and differential privacy (178.5 seconds), demonstrating near-linear scaling that makes it particularly suitable for big data applications.

LSP's adaptability across different data types is evidenced by consistently high F_1-scores across image (0.956), text (0.934), and tabular data (0.942). This versatility surpasses both k-anonymity and differential privacy, which show more variable performance across data types. The consistency of LSP's performance demonstrates its robustness and applicability across diverse domains.

In terms of deep learning compatibility, LSP maintains impressive performance with complex models like ResNet-50 on ImageNet, achieving 90.8% accuracy compared to raw data's 92.1%. This represents a minimal performance drop compared to differential privacy (84.3%) and federated learning (88.7%), indicating LSP's suitability for modern deep learning applications.

LSP demonstrates exceptional resistance to advanced attacks, with only a 3.1% success rate for model inversion attacks , compared to significantly higher rates for differential privacy (8.4%) and federated learning (13.7%). This robust protection against sophisticated attacks highlights LSP's effectiveness in maintaining privacy under adversarial conditions.

Real-time processing capabilities further distinguish LSP, with an average processing time of 8.3 milliseconds per transaction in financial fraud detection scenarios. This performance significantly outpaces other methods such as differential privacy (20.4 milliseconds), k-anonymity (31.8 milliseconds), and especially homomorphic encryption (412.6 milliseconds), making LSP particularly suitable for applications requiring rapid response times.

Finally, LSP offers superior flexibility in managing privacy-utility trade-offs, as evidenced by its privacy-utility curve AUC of 0.923, compared to differential privacy (0.876) and k-anonymity (0.801). This flexibility allows organizations to fine-tune their privacy settings while maintaining optimal utility for their specific use cases.

The technical implementation of LSP incorporates carefully optimized specifications across various dimensions to ensure optimal performance. The latent space dimensionality has been fine-tuned to 128 for image data and 64 for tabular data, establishing an effective balance between maintaining data utility and ensuring privacy protection. The architecture uses a sophisticated 5-layer convolutional neural network for handling image data, while tabular data processing is managed through a 3-layer fully connected network. Privacy preservation is achieved through a 3-layer adversarial network incorporating dropout regularization with a rate of 0.3.

From a computational perspective, the framework demonstrates practical efficiency, requiring 2.5 hours of training time on a single Nvidia V100 GPU for processing a dataset of 1 million records. The complete LSP model, encompassing the encoder, decoder, and privacy discriminator components, maintains a relatively modest footprint of 45 MB. Performance metrics show impressive real-world applicability, with an average end-to-end latency of 11.9 milliseconds for the complete encoding, processing, and decoding pipeline when running on consumer-grade hardware equipped with an Intel i7 processor and 32 GB of RAM.

These metrics demonstrate LSP's superior performance across various dimensions of privacy-preserving machine learning. The method consistently outperforms traditional techniques in terms of balancing privacy and utility,

computational efficiency, scalability, and adaptability to different data types and machine-learning tasks.

Latency, Scalability, and Performance Analysis

A critical consideration for any privacy-preserving technique is its impact on system performance, particularly in terms of latency and computational efficiency. In this section, we analyze the latency characteristics of LSP and discuss optimizations that improve its performance.

Latency Analysis

Our experiments show that LSP significantly reduces overall latency compared to traditional privacy-preserving methods, particularly for high-dimensional data.

Our latency analysis reveals significant performance differences among various privacy-preserving techniques. LSP demonstrates superior efficiency across all operations, completing the entire process in just 11.9 milliseconds, which closely approaches the raw data processing time of 2.1 milliseconds. Breaking down the operations, LSP requires only 5.2 milliseconds for encoding, 1.8 milliseconds for classification processing, and 4.9 milliseconds for decoding.

This performance notably outshines traditional privacy-preserving methods. In comparison, k-anonymity takes considerably longer, requiring 15.3 milliseconds for encoding, 3.8 milliseconds for classification, and 12.7 milliseconds for decoding, totaling 31.8 milliseconds. Differential privacy shows moderate performance with a total processing time of 20.4 milliseconds, split between 8.7 milliseconds for encoding, 4.2 milliseconds for classification, and 7.5 milliseconds for decoding.

Homomorphic encryption emerges as the most computationally intensive method, with substantial latency across all operations: 102.5 milliseconds for encoding, 387.6 milliseconds for classification, and 98.3 milliseconds for decoding, summing to a total of 588.4 milliseconds.

Notably, LSP achieves classification processing speeds of 1.8 milliseconds, even surpassing raw data processing (2.1 milliseconds), while maintaining robust privacy protection. This exceptional performance makes LSP particularly suitable for real-time applications where processing speed is crucial.

Scalability Analysis

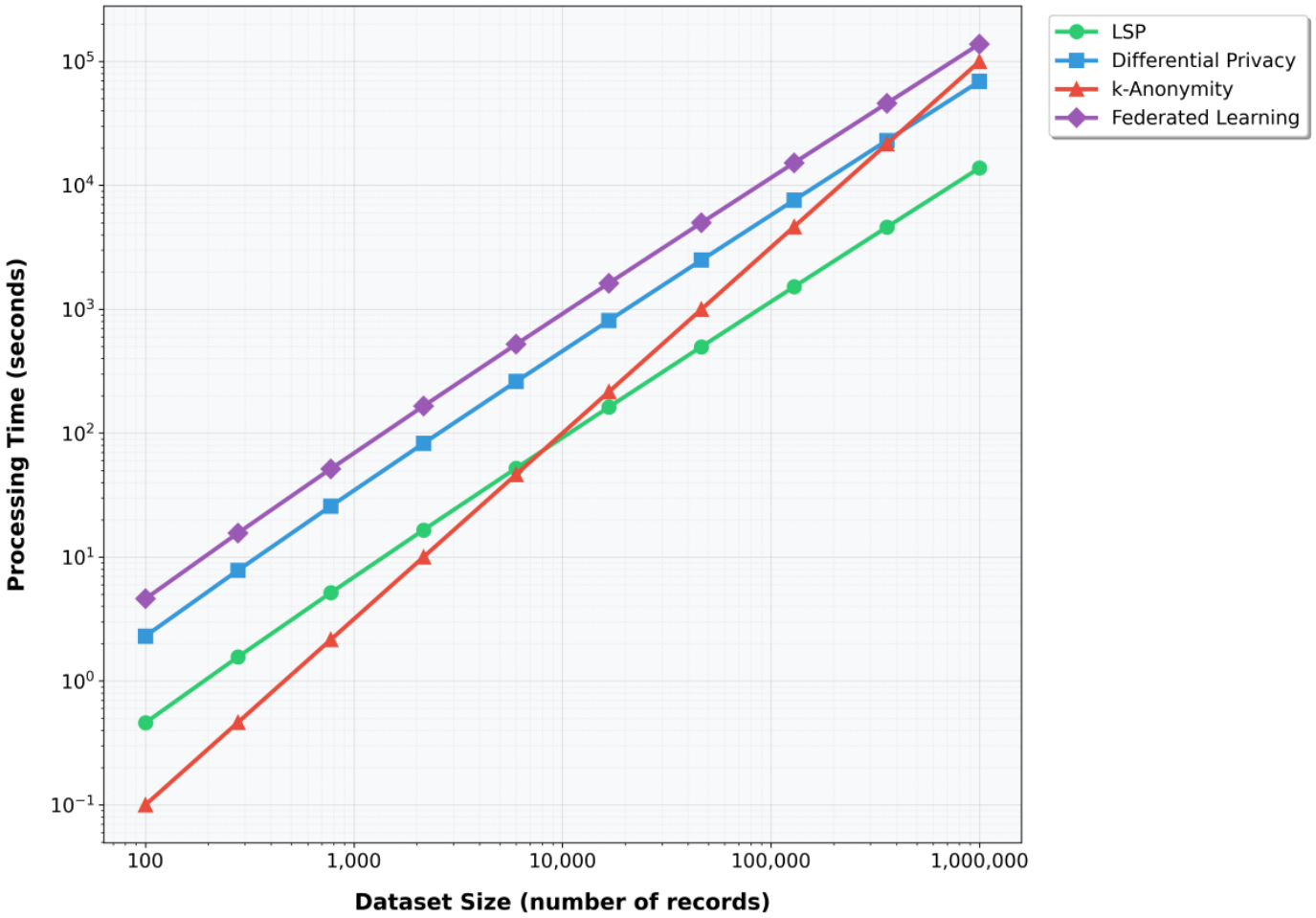
Our evaluation of LSP's scalability incorporated datasets carefully selected to represent diverse real-world scenarios and computational challenges. For the scalability experiments, we utilized datasets ranging from 10^2 to 10^6 records, obtained from established public repositories including Kaggle and Huggingface. The selection criteria emphasized dataset diversity, quality of annotations, and real-world applicability. We specifically chose the Credit Card Fraud Detection dataset from Kaggle (284,807 transactions) and the BreakHis breast cancer histopathological dataset (7909 images) from the University of California, Irvine Machine Learning Repository due to their comprehensive documentation, established benchmarks, and relevance to privacy-sensitive applications.

Dataset Selection

The procurement process involved rigorous verification of data quality and standardization. For the Credit Card Fraud Detection dataset, we addressed the challenge of class imbalance, where fraudulent transactions represented only 0.172% of all cases. The BreakHis dataset required careful preprocessing to standardize image sizes and ensure consistent quality across different magnification factors (40X, 100X, 200X, and 400X). Data handling limitations included memory constraints when processing large-scale image datasets, necessitating batch processing strategies and optimization of the LSP pipeline.

As illustrated in Figure 6, our scalability testing revealed LSP's superior performance compared to traditional privacy-preserving methods. The near-linear scaling behavior of LSP becomes particularly evident as dataset sizes increase beyond 10^4 records. Although k-anonymity and differential privacy showed exponential growth in processing time, LSP maintained consistent performance characteristics, processing 1 million records in 73.2 seconds compared to 1258.3 seconds for k-anonymity and 178.5 seconds for differential privacy. Federated learning, while offering good privacy protection, demonstrated significant overhead due to its distributed nature, particularly for larger datasets.

Figure 6. LSP scalability compared with other privacy-preserving methods. LSP: latent space projection.



Real-Time Performance Analysis

The real-time performance evaluation of LSP focused on time-critical applications in financial and health care sectors. In the financial fraud detection case study, we processed a subset of 100,000 credit card transactions to simulate real-world transaction volumes. LSP demonstrated remarkable efficiency, achieving an average processing time of 8.3 milliseconds per transaction. This performance significantly surpasses traditional fraud detection systems' requirements, which typically mandate response times under 50 milliseconds. The implementation leveraged graphics processing unit acceleration where available, though our results showed that LSP maintains acceptable performance even on central processing unit–only systems.

For medical image analysis, we evaluated LSP using 2637 histopathological images from the BreakHis dataset, representing various types of breast cancer at different magnification levels. The system achieved an average processing time of 14.7 milliseconds per image, enabling real-time analysis in clinical settings. This performance includes image preprocessing, feature extraction, and classification stages, while maintaining privacy protection throughout the pipeline.

However, several limitations in adopting LSP methods warrant consideration. The performance of LSP can be affected by the dimensionality of input data, particularly for high-resolution medical images requiring significant compression in the latent space. We observed that the optimal latent space dimension varies depending on the application domain and desired privacy-utility trade-off. Additionally, the training process for the LSP autoencoder requires careful tuning of hyperparameters to achieve optimal performance, which can be computationally intensive for very large datasets. Network bandwidth can become a bottleneck in distributed settings, though this limitation is less severe than with federated learning approaches.

Resource requirements also present practical limitations. Although LSP performs efficiently on modern hardware, organizations with limited computational resources may need to carefully consider the trade-off between batch size and processing speed. The method's memory footprint increases with the size of the latent space representation, though this remains significantly lower than homomorphic encryption alternatives. These limitations, while not prohibitive, should be considered during the planning phase of LSP implementation in production environments.

Implications for Responsible AI and Governance

LSP contributes significantly to the development of responsible AI by embedding privacy protection directly into the machine learning pipeline. This section discusses the

implications of LSP for AI governance and its alignment with global regulatory frameworks.

Fairness and Bias Mitigation

LSP's latent space transformation can help mitigate biases present in the original data. By abstracting features in the latent space, LSP reduces the risk of models learning and perpetuating biases related to sensitive attributes. Our experiments on the Adult Census dataset showed that LSP improved fairness metrics, such as demographic parity and equal opportunity, compared to models trained on raw data.

Transparency and Explainability

Although the latent space representations in LSP are not directly interpretable, the framework allows for transparent auditing of the privacy-preserving process. Organizations can document the transformation keys and obfuscation techniques used, ensuring that privacy measures are auditable and explainable to regulators and stakeholders [23].

Accountability and Access Control

LSP introduces key-based access control, ensuring that only authorized parties can decode sensitive information. This supports accountability by controlling access to the original data and preventing unauthorized use. Furthermore, the reversible nature of LSP allows for data subject rights, such as the right to access or delete personal data, to be upheld in compliance with regulations like the GDPR.

Alignment With Global AI Governance Frameworks

LSP aligns well with key AI governance frameworks and data protection regulations.

GDPR Compliance

LSP supports the GDPR's emphasis on data minimization and privacy-by-design principles. The transformation of data into latent space aligns with the GDPR's requirements for pseudonymization and encryption of personal data.

CCPA and Data Portability

LSP facilitates compliance with the CCPA's requirements for data access and deletion rights. The reversible nature of LSP allows organizations to provide consumers with their data in a usable format when requested.

HIPAA and Sensitive Data Protection

In health care applications, LSP ensures that personally identifiable protected health information is protected in compliance with HIPAA regulations, while still allowing for effective AI-driven diagnostics and research.

Future Work

Several avenues for future research remain:

1. Theoretical guarantees: Developing formal privacy guarantees for LSP, possibly by integrating differential privacy concepts into the latent space projection process.
2. Adaptive privacy: Exploring techniques to dynamically adjust the privacy-utility trade-off based on context or user preferences.
3. Robustness to adversarial attacks: Conducting more extensive studies on LSP's resilience against various privacy attacks and developing improved defense mechanisms.
4. Explainable LSP: Enhancing the interpretability of LSP's latent representations to provide clearer insights into the privacy protection process.

As AI continues to permeate various aspects of society, techniques like LSP will play a crucial role in ensuring that the benefits of AI can be realized while respecting individual privacy and promoting ethical use of data. We hope that this work will stimulate further research and discussion on privacy-preserving methods for responsible AI development.

Conclusion

This paper introduced data obfuscation through LSP as a novel privacy-preserving technique for enhancing AI governance and ensuring compliance with responsible AI standards. Through extensive experiments and real-world case studies, we demonstrated LSP's ability to protect sensitive information while maintaining high utility for machine learning tasks.

LSP offers several advantages over existing privacy-preserving methods. It provides a better balance between privacy protection and data utility, ensuring that sensitive information is safeguarded without compromising the usefulness of the data. Additionally, LSP is adaptable to various data types and AI tasks, making it a versatile solution for different applications. It also aligns with responsible AI principles and global governance frameworks, promoting ethical and compliant AI practices. Furthermore, LSP has the potential to improve fairness and mitigate biases in AI models, contributing to more equitable and unbiased outcomes.

Data Availability

The datasets used in this manuscript are publicly available.

Conflicts of Interest

None declared.

References

1. Scheibner J, Raisaro JL, Troncoso-Pastoriza JR, et al. Revolutionizing medical data sharing using advanced privacy-enhancing technologies: technical, legal, and ethical synthesis. J Med Internet Res. Feb 25, 2021;23(2):e25120. [doi: 10.2196/25120] [Medline: 33629963]
2. Narayanan A, Shmatikov V. Robust de-anonymization of large sparse datasets. Presented at: 2008 IEEE Symposium on Security and Privacy (sp 2008); May 18-22, 2008:111-125; Oakland, CA. URL: https://ieeexplore.ieee.org/abstract/document/4531148 [Accessed 2025-03-05]
3. Papernot N, McDaniel P, Sinha A, Wellman M. Towards the science of security and privacy in machine learning. arXiv. Preprint posted online on Nov 11, 2016. [doi: 10.48550/arXiv.1611.03814]
4. Fredrikson M, Jha S, Ristenpart T. Model inversion attacks that exploit confidence information and basic countermeasures. Presented at: CCS'15; Oct 12-16, 2015:1322-1333; Denver, CO. Oct 12, 2015.[doi: 10.1145/2810103.2813677]
5. Shokri R, Stronati M, Song C, Shmatikov V. Membership inference attacks against machine learning models. Presented at: 2017 IEEE Symposium on Security and Privacy (SP); May 22-26, 2017:3-18; San Jose, CA. [doi: 10.1109/SP.2017.41]
6. Chen Y, Esmaeilzadeh P. Generative AI in medical practice: in-depth exploration of privacy and security challenges. J Med Internet Res. Mar 8, 2024;26:e53008. [doi: 10.2196/53008] [Medline: 38457208]
7. Carlini N, Liu C, Erlingsson Ú, Kos J, Song D. The secret sharer: evaluating and testing unintended memorization in neural networks. Presented at: 28th USENIX Security Symposium (USENIX Security 19); Aug 14-16, 2019:267-284; Santa Clara, CA. URL: https://www.usenix.org/system/files/sec19-carlini.pdf [Accessed 2025-03-05]
8. Sweeney L. k-anonymity: a model for protecting privacy. Int J Unc Fuzz Knowl Based Syst. Oct 2002;10(5):557-570. [doi: 10.1142/S0218488502001648]
9. Dwork C, Roth A. The algorithmic foundations of differential privacy. FNT Theoretical Comput Sci. 2014;9(3-4):211-407. [doi: 10.1561/0400000042]
10. Abadi M, Chu A, Goodfellow I, et al. Deep learning with differential privacy. Presented at: CCS'16; Oct 24-28, 2016:308-318; Vienna, Austria. Oct 24, 2016.[doi: 10.1145/2976749.2978318]
11. Chaudhuri K, Monteleoni C, Sarwate AD. Differentially private empirical risk minimization. J Mach Learn Res. Mar 2011;12:1069-1109. [Medline: 21892342]
12. Gentry C. Fully homomorphic encryption using ideal lattices. Presented at: STOC '09; May 31 to Jun 2, 2009:169-178; Bethesda, MD. May 31, 2009.[doi: 10.1145/1536414.1536440]
13. McMahan B, Moore E, Ramage D, Hampson S, y Arcas BA. Communication-efficient learning of deep networks from decentralized data. Presented at: Artificial Intelligence and Statistics; Apr 20-22, 2017:1273-1282; Fort Lauderdale, FL. URL: https://proceedings.mlr.press/v54/mcmahan17a/mcmahan17a.pdf [Accessed 2025-03-05]
14. Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. Preprint posted online on Jun 10, 2014. URL: https://arxiv.org/abs/1406.2661 [Accessed 2025-03-05]
15. McSherry FD. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. Presented at: Proceedings of the 2009 ACM SIGMOD International Conference on Management of data; Jun 29 to Jul 2, 2009:19-30; Providence, RI. [doi: 10.1145/1559845.1559850]
16. Kingma DP, Welling M. Auto-encoding variational Bayes. arXiv. Preprint posted online on May 1, 2013. URL: https://www.cs.columbia.edu/~blei/fogm/2018F/materials/KingmaWelling2013.pdf [Accessed 2025-03-05]
17. Dwork C, McSherry F, Nissim K, Smith A. Calibrating noise to sensitivity in private data analysis. In: Halevi S, Rabin T, editors. Theory of Cryptography TCC 2006 Lecture Notes in Computer Science. Vol 3876. Springer; 2006. [doi: 10.1007/11681878_14]
18. Balle B, Barthe G, Gaboardi M. Privacy amplification by subsampling: tight analyses via couplings and divergences. Adv Neural Inf Process Syst. 2018;31. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/3b5020bb891119b9f5130f1fea9bd773-Paper.pdf [Accessed 2025-03-05]
19. Gilad-Bachrach R, Dowlin N, Laine K, Lauter K, Naehrig M, Wernsing J. Cryptonets: applying neural networks to encrypted data with high throughput and accuracy. Presented at: International Conference on Machine Learning; Jun 19-24, 2016:201-210; New York, NY. URL: https://proceedings.mlr.press/v48/gilad-bachrach16.pdf [Accessed 2025-03-05]
20. Melis L, Song C, De Cristofaro E, Shmatikov V. Exploiting unintended feature leakage in collaborative learning. Presented at: 2019 IEEE Symposium on Security and Privacy (SP); May 20-22, 2019:691-706; San Francisco, CA. [doi: 10.1109/SP.2019.00029]
21. Lee GH, Shin SY. Federated learning on clinical benchmark data: performance assessment. J Med Internet Res. Oct 26, 2020;22(10):e20891. [doi: 10.2196/20891] [Medline: 33104011]

22. Xu L, Skoularidou M, Cuesta-Infante A, Veeramachaneni K. Modeling tabular data using conditional GAN. Presented at: Advances in Neural Information Processing Systems; Dec 8-14, 2019:7333-7343; Montreal, QC. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/254ed7d2de3b23ab10936522dd547b78-Paper.pdf [Accessed 2025-03-05]
23. Adadi A, Berrada M. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access. 2018;6:52138-52160. [doi: 10.1109/ACCESS.2018.2870052]

Abbreviations

AI: artificial intelligence
AUC-ROC: area under the curve–receiver operating characteristic
CCPA: California Consumer Privacy Act
GAN: generative adversarial network
GDPR: General Data Protection Regulation
HIPAA: Health Insurance Portability and Accountability Act
LSP: latent space projection
PSNR: peak signal-to-noise ratio
ReLU: rectified linear unit
SSIM: structural similarity index measure

Edited by Ching Nam Hang; peer-reviewed by Reenu Singh, Trutz Bommhardt; submitted 15.12.2024; final revised version received 01.02.2025; accepted 02.02.2025; published 12.03.2025

Please cite as:
Vaijainthymala Krishnamoorthy M
Data Obfuscation Through Latent Space Projection for Privacy-Preserving AI Governance: Case Studies in Medical Diagnosis and Finance Fraud Detection
JMIRx Med 2025;6:e70100
URL: https://med.jmirx.org/2025/1/e70100
doi: 10.2196/70100

Information Processing and Management 60 (2023) 103167

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

A privacy preserving framework for federated learning in smart healthcare systems

Wenshuo Wang [a], Xu Li [a], Xiuqin Qiu [a], Xiang Zhang [b], Vladimir Brusic [b,c,*], Jindong Zhao [a,*]

[a] School of Computer and Control Engineering, Yantai University, Yantai, China
[b] School of Computer Science, University of Nottingham Ningbo China, Ningbo, China
[c] Shandong Lengyan Medical Technology Inc., Yantai, China

ARTICLE INFO

Keywords:
Federated learning
Ring signature
Privacy preserving
Source inference attack
Smart healthcare system

ABSTRACT

Federated Learning (FL) is a platform for smart healthcare systems that use wearables and other Internet of Things enabled devices. However, *source inference attacks* (SIAs) can infer the connection between physiological data in training datasets with FL clients and reveal the identities of participants to the attackers. We propose a comprehensive smart healthcare framework for sharing physiological data, named FRESH, that is based on FL and ring signature defense from the attacks. In FRESH, physiological data are collected from individuals by wearable devices. These data are processed by edge computing devices (e.g., mobile phones, tablet PCs) that train ML models using local data. The model parameters are uploaded by edge computing devices to the central server for joint training of FL models of disease prediction. In this procedure, certificateless ring signature is used to hide the source of parameter updates during joint training for FL to effectively resist SIAs. In the proposed ring signature schema, an improved batch verification algorithm is designed to leverage additivity of linear operations on elliptic curves and to help reduce the computing workload of the server. Experimental results demonstrate that FRESH effectively reduces the success rate of SIAs and the batch verification method significantly improves the efficiency of signature verification. FRESH can be applied to large scale smart healthcare systems with FL involving large numbers of users.

1. Introduction

The availability and affordability of wearable devices and their growing popularity has caused a rapid growth of quantities of privately owned medical data in recent years. Wearables enable continuous collection and recording of the data streams of physiological parameters, movement of users and relevant environmental variables (Mukhopadhyay, 2014). Such data streams allow users to obtain personalized medical data sets that contain the information about their health status and lifestyle. Wearable devices play an important role in health monitoring, safety monitoring, home rehabilitation progress, therapy effectiveness evaluation, early disease detection, and other health status indicators (Dias & Cunha, 2018, Jia et al., 2017). Wearable technologies are transformative because they enable continuous healthcare by linking home, mobile, and in-clinic health monitoring. The analysis of data can be done in real time for continuous monitoring and raising alarms when needed. A key benefit of these technologies is the capacity for monitoring

* Corresponding authors.
E-mail addresses: vladimir.brusic@nottingham.edu.cn (V. Brusic), zhjdong@ytu.edu.cn (J. Zhao).

https://doi.org/10.1016/j.ipm.2022.103167
Received 14 June 2022; Received in revised form 7 November 2022; Accepted 8 November 2022
Available online 14 November 2022

healthy individuals at minimal cost to detect anomalies and to enable early diagnosis or timely preventative measures.

Batch mode analysis of aggregated data allows for building models for diagnosis, prognosis, optimization of intervention, and informing public health decisions (Bilkey et al., 2019). Several technologies — such as genomics, molecular and cellular phenotyping, high resolution imaging, Big Data analytics, machine learning, artificial intelligence, mobile computing, and Internet of Things — are converging to enable better healthcare and advanced approaches to disease treatment and prevention. New approaches, such as precision medicine or personalized medicine (Ashley, 2016), take into account individual characteristics of patient (genetic profile, lifestyle, and environmental variables) and integrate them with the data from electronic health records to optimize disease status assessment, and prevention, for individual patients. Conventional healthcare data analytics are gradually being replaced by the precision medicine data analytics that use advanced computational and data science approaches for the improvement of healthcare (Fig. 1).

Precision medicine combines Big Data Analytics and the use of machine learning (Ahmed et al., 2020) to develop disease prediction models and support individualized medical services decisions. Wearable devices, Internet of Things and mobile health technologies enable continuous monitoring of human vital signs, activities, and other measurements (Dias & Cunha, 2018; Zhang et al., 2022) during daily life (at home, indoor, or outdoor), or in in health care institutional settings (ambulatory, clinical, or specialized). The trends in healthcare and medicine based on these technologies include Smart Health Home and AAA medicine (Anywhere, Any time, Any environment). The promises of these technologies combined with the precision medicine include (Dunn et al., 2018):

- health monitoring outside of the clinic and prediction of health events are common
- mobile and digital health by enable continuous, longitudinal health monitoring outside of the clinic
- health monitoring is done for monitoring general health, cardiovascular health, obesity, diabetes, temperature, gait and activity, and muscular function
- sensors can be mechanical, physiological, and biochemical
- algorithms can automatically process and interpret the data from wearables and detect health events, predict the trends, and indicate possible need for intervention. Examples include the detection of inflammation, determining the status of cardiac health or diabetes, prediction of obesity trends, assessment of sleep quality, and mental health, among others.

Practical frameworks for systematic and medically meaningful deployment of smart healthcare systems are still in their infancy and they fall short of complying to medical standards essential for institutional health care delivery.

1.1. *Internet of things and smart health home*

The center of out-of-clinic healthcare services is smart health home (SHH). SHH is equipped with sensors and devices that collect data critical for providing health care services outside medical institutions (Zhang et al., 2022). Health data management involves

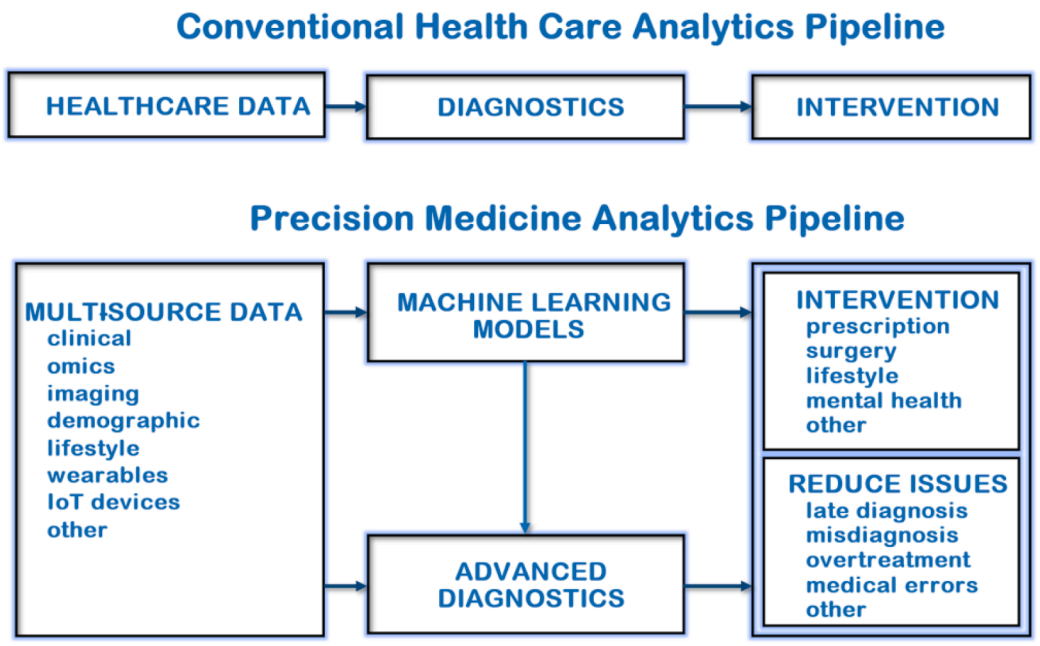


Fig. 1. Health care analytics pipeline in conventional health care and in precision medicine. Precision medicine addresses key issues in healthcare field through data-driven models generated by machine learning algorithms. Their application optimizes health intervention and reduces the negative issues of conventional health care. Adapted and extended from Ahmed et al. (2020).

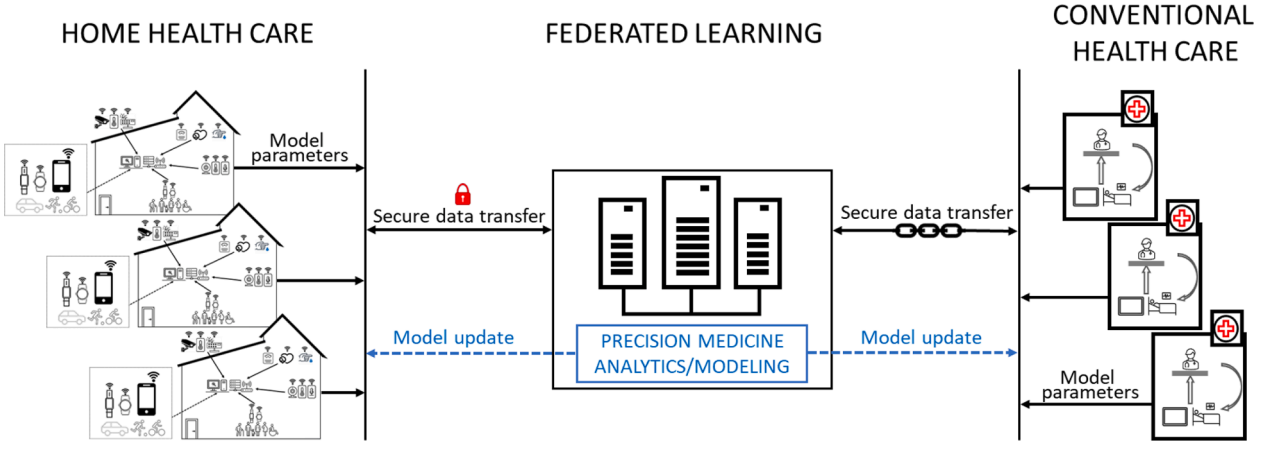


Fig. 2. Smart healthcare system with federated learning. Federated learning (FL) enables the link between home and clinical healthcare by sharing models instead of raw data for precision medicine applications. The outcome of FL can be used for personalized healthcare at home (left side) or for advanced diagnostic in the clinical setting (right side). FL addresses data interoperability between diverse data systems and data protocols. In comparison with the conventional Machine Learning frameworks, FL shares models rather than large quantities of data. The communication cost and data security and privacy risks are reduced. Home health care (left side) represents cross-device environments), while conventional health care (right side) represents cross-silo environments (Liu et al., 2021).

acquisition (collection, cleaning, and formatting), analysis (processing, verification, and validation), modeling (regression, clustering, prediction, or simulation), and sharing (transferring, storage, and use). The main functions of precision medicine are facilitation of data analysis, support for interpretation of the results by health professionals, and support of decision making in healthcare. Home is the best place for managing personal health data that are acquired outside of healthcare institutions. Multiple sensors and IoT devices are used at home for unobtrusive health monitoring, including monitoring of functional performance, emergency, physiological parameters, safety and security, and social interaction (Wang et al., 2021). Furthermore, smart homes are equipped with edge computing facilities that enable rapid responses, context awareness, and mobility support (Caprolu et al., 2019). Health data collected in mobile setting can be combined with home-based measurements to provide more complete information than conventional health monitoring. Unlike medical data management, such as Electronic Health Record (EHR), that are subject to regulation and represent a mature area of health informatics, the management, sharing and use of home health care data are still in the research stage. Practical barriers of smart health home adoption as a major tool in healthcare are normative (device, hardware, and software regulations and standards) and acceptance-related (by medical professionals, society, and consumers). The areas of concerns include data protection, data security, vulnerable groups data processing, accountability, data minimization and sharing, transparency, sustainability, resilience, and interoperability (Zhang et al., 2022). Secure and safe data sharing is essential since an average user has neither sufficient medical knowledge nor enough data to produce meaningful results from simulations or predictions. Pooling data from multiple homes facilitates the development and implementation meaningful models for disease prediction and health monitoring. The participation of health and data analytics professionals ensures that these models are meaningful and suitable for precision medicine applications (Fig. 1). Alternatively, generating but not sharing data often leads to isolated data islands and fragmentation of software environment (Dimitrov, 2016). Privacy and data security are the most extensively studied areas in smart healthcare systems (Atlam & Wills, 2020; Martin & Kung, 2018; Ramli et al., 2012; Abouelmehdi et al., 2017). Technical mechanisms, such as physical protection, encryption, password management, data protection, and control methods are regulated by data security laws (Abouelmehdi et al., 2017).

1.2. Federated learning in smart healthcare systems

Federated learning (FL), an emerging distributed collaborative artificial intelligence (AI) paradigm is particularly attractive for smart healthcare, because it coordinates multiple clients that enable edge-based learning (Rajula et al., 2020; Sheller et al., 2020; Banabilah et al., 2022). Traditional machine learning (ML) collects all data on one site for centralized learning, while FL achieves model learning without sharing private data (Yin et al., 2021). In the FL frameworks, users do not need to upload data, but they can directly train models locally on their edge devices using their private data sets, and only upload parameter updates to the central server for model learning by aggregation. This procedure involves the transmission of update parameters only to ensure that user privacy is not exposed. FL prevents data island formation by ensuring joint model training of ML models by multiple parties (Yang et al., 2019). Furthermore, FL plays an important role in precision medicine by linking home health care and conventional (institutional) health care, as shown in Fig. 2.

Medical institutions have health informatics resources that enable centralized data management and compliance with regulatory and medical software and medical device standards. However, an average user of SHH has neither sufficient medical knowledge nor enough medical data to manage data collection, management, and use. Pooling data from multiple users of wearables is essential for the development of models for disease prediction and health monitoring and provides a critical mass for participation of health and data analytics professionals. However, data sharing for the development of health monitoring models raises important health data concerns: safety, privacy, interoperability, and data integrity. FL technologies can be enhanced by specific solutions that reduce data concerns, such as anomaly detection (Wu et al., 2022) and cognitive detection of cyber-attacks (Makkar & Park, 2022). Currently, multiple wearable devices communicate to the server using their own protocols, making software environment fragmented.

Advances in Internet of Health Things enable efficient collection and parsing of health and medical data but, due to security and privacy issues, centralized data collection is increasingly limited by regulatory restrictions (Nguyen et al., 2022). Traditional machine learning (ML) collects all data on one site for centralized learning, while FL achieves model learning without sharing private data (Yin et al., 2021). In the FL frameworks, wearable device users do not need to upload data, but they can directly train models locally on their edge devices using private data sets. Rather than sharing data, FL uploads parameter updates to the central server for model learning by aggregation. This procedure involves transmission of updated parameters only to ensure that user privacy is preserved. FL avoids data island problem through joint training of ML models by multiple parties (Yang et al., 2019).

Model parameter sharing, increased number of training iterations and increased communications cost present a new set of risks to federated environment that open new vulnerabilities. FL requires large number of collaborative clients for building a federated model, making the aggregation server or each client a weak point and a source of vulnerabilities. Collaborative ML processes are vulnerable to several types of attacks including identification, inference, and linkage attacks (Liu et al., 2021). A main premise of FL is the collaborative training of ML models that preserve privacy of the owner of many devices that participate in learning. Inference-based attacks constitute critical risks to the privacy of FL because they undermine the primary quality of the concept (Mothukuri et al., 2021). In this work we provide a brief analysis of the existing risks and types of attacks, review key methods for protection against such attacks, and present a novel solution for protection against inference attacks within the FL frameworks based on ring signature method. This scheme is particularly useful when membership inference attacks (MIA) are successful and there is a need to prevent the source inference attacks (SIA). These types of attacks are reviewed in Section 2.

1.3. Contributions and structure of this study

This work focuses on the design and implementation of solution that addresses privacy vulnerability within FL framework by deployment of ring signatures. Our solution provides a novel algorithm for ring signature that enhances performance of the system that is suitable for smart health. We term our solution FRESH (**F**ederated-learning **R**ing-signature **E**nhanced performance solution for **S**mart **H**ealth). The main contributions of this work are:

- We present the FRESH framework for healthcare and medical applications. FRESH is a privacy preserving scheme of health and medical data based on FL and ring signature technology. FRESH implements clients cooperatively to train health model under condition of limited data storage locally, while the barriers between different users are reduced.
- A novel improved ring signature scheme based on the deployment of an elliptic curve is described. In this scheme the computation cost of the system is reduced due to lower time of multiplication on the elliptic curve group. The time needed for signature verification is significantly reduced by the batch validation method we developed. In consequence, the throughput of FRESH is greatly improved.
- SIAs are effectively resisted by utilizing ring signature. The experimental results have shown that FRESH increases affordable communication and computational costs at linear rate, while the identity privacy exposure and related risk for the participants in FL system are practically eliminated by reduction to random guessing.
- FRESH system is applicable to both cross device systems for health data (SHH environment) and cross-silo systems for medical data (Smart Health institutional environment)

This paper is divided into seven parts. Section 1 introduces research motivation and contribution. Section 2 reviews related research, discusses the key concepts, and defines the goals of this study. Section 3 describes the architecture of FRESH and explains technical details. Section 4 presents the results of privacy and security analysis in FRESH. Section 5 describes the experiments and the performance analysis of a FRESH experiment. Section 6 discusses practical implications of FRESH. The conclusions are summarized in Section 7.

2. Related work and key concepts

Data — including the collection, analysis, critique, and sharing — make the foundation of medical and health research and practice (Molloy, 2011). Smart healthcare systems as data-driven systems are characterized by the ability to collect various vital signs, process them, and transform them into information, knowledge, and corresponding actions (Demirkan, 2013). Wearables devices, such as smart health bands, smart watches, and glucose monitors, have emerged as the common sources of health data. Sharing and gathering big amount of health data enable the discovery of new knowledge using both Big Data analytics and advanced ML algorithms. Big Data analytics is particularly important for capturing time series of vital signs, such as ECG signals, heart rate, or activity data. The initial steps of Big Data analytics involve data organization (graphs, ontologies, or other representations), data cleaning and reduction (summary statistics, data distributions, principal component analysis, correlation analysis, and others), and data integration and processing (database formation, comparative analysis, feature analysis) (Saranya & Asha, 2019). The reach data sets produced by smart health environments can be analyzed by well-accepted conventional statistical analysis, such as the analysis of a small number of medically important variables to identify associations for example odds ratios or hazard ratios using regression models (Rajula et al., 2020). Such results are easy to understand and are standard methods used in medical practice. The limitations of conventional statistical methods emerge from their reliance on strong assumptions such as data distributions, error distributions, or additivity of parameters, that may not be satisfied in real-life health data sets. ML methods, on the other hand, offer flexibility because they are mostly data-driven and they do not depend on assumptions about data (Rajula et al., 2020). Multiple data sets can be combined into common input data that are mapped to the outputs using training and testing approaches of ML models. The applications are very diverse – they may include prediction of disease status and progression, and improvement of medical diagnosis (Saranya & Asha, 2019, Lyu et al., 2020a). ML is useful for identifying patterns within the quantities of diverse data to make the system act "smart" for disease prediction, diagnosis, and lifestyle intervention (Tuvshinjargal & Hwang, 2021; Le, 2021; Brahmecha et al., 2021).

Data security and privacy issues are the major considerations in smart healthcare systems. Data security refers on preventing unauthorized access, data or model corruption, or data loss throughout the entire lifecycle of the system (Summers & Koehne, 2004). Cyber-attacks in smart healthcare domain may target devices, IoT connections, data storage, data communication, or data analysis modules. Successful cyber-attacks will result in one or more undesirable events such as data modification, data breach, data loss, privacy leakage and others (Razaque et al., 2019). The ethical concerns such as user trust, safety and privacy, vulnerable groups, autonomy, and social morality are emerging as system requirements (Zhang et al., 2022) thus making system cyber-protection a design requirement. Users of smart health systems, both the patients and medical professionals, may be misled by the incorrect data, poorly deployed prediction models, or malicious changes of data and models. Current research efforts largely focus on privacy protection, although they should be extended to the comprehensive set of data, model, and system protection measures (Kairouz et al., 2021).

Interoperability. In practice, data interoperability issues create obstacles for the sharing health data. Data encoded by different mechanisms and communication protocols need to be converted into standardized formats before sharing. The diversity of sensors and devices complicates interoperability – individual parties may develop IoT ecosystems that may become isolated data islands defined by own set of devices and integration solutions. These IoT ecosystems include limited data insufficient for building high-quality ML models. Big Data sharing platforms address the issue of simple and standardized ways of data sharing (Bröring et al., 2017). Wang et al.

(2019b) proposed a medical big data sharing platform that utilizes blockchain and ring signature for access and data sharing. The access control mechanism in this system uses smart contract, while ring signature is used to protect user privacy. However, the efficiency of data utilization in this scheme is low, thus preventing the use of this system for advanced applications, such as disease prediction.

2.1. Federated learning and vulnerabilities of the scheme

FL enables multiple parties to jointly train machine learning models without exchanging their local data. It addresses the interoperability issues by training machine learning models on the local edge nodes and sharing model parameters instead of sharing the raw data for centralized training. This schema reduces the vulnerability of cybersecurity by eliminating the need for transmission of raw data across the system (Kairouz et al., 2021).

The FL concept emerged to address data sharing for training ML models (Konečný et al., 2016a, 2016b; McMahan et al., 2017). It is essentially a ML framework that allows multiple parties to collaboratively learn a shared prediction model. FL works without direct access to local data improving privacy and data security. Because training data owned by different participants are not shared between clients and servers, user data privacy is ensured. Since the types and format of data collected by wearable devices are basically the same, horizontal FL is used to enable participants to jointly train disease prediction models. The architecture of horizontal FL is shown in Fig. 3.

The specific model training steps include:

- Central server releases joint modeling task to seek clients. After individual clients are identified, the central server issues the initial parameter sets to each client.
- After receiving the initial parameter set, the client updates the local model iteratively using own private dataset. After local updating, the locally trained model is uploaded to the central server.
- The central server then aggregates these parameters according to suitable procedures to obtain global ML model parameter update.
- The global parameters will be sent to the client for the next round of training. When the loss function converges, the training is stopped, otherwise the training is iterated until the training termination conditions are met.

Although FL was devised to improve privacy and data security of collaborative machine learning processes, new cyber-security threats targeting FL systems have rapidly emerged (Kumar et al., 2021). The categories of threats to FL are determined by the source (insider vs. outsider), phase of ML (training phase vs. inference phase), attack intention (semi-honest vs. malicious), attack objective (random vs. targeted attacks), and attack target (model vs. privacy) – the descriptions are available in Lyu et al. (2020b) and Jere et al. (2020). Two main types of attack targets in FL are privacy attacks and model performance attacks (Jere et al., 2020). The

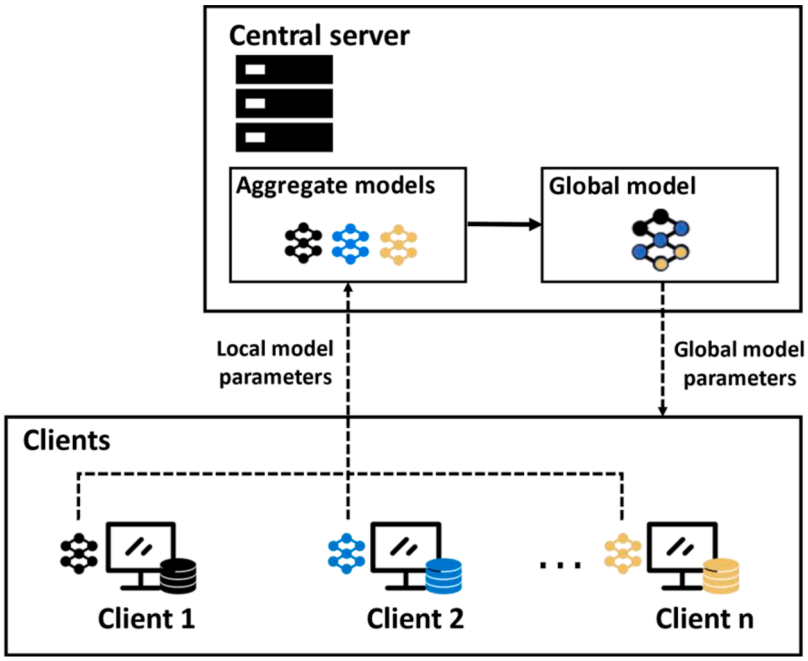


Fig. 3. The architecture of horizontal FL includes central server and multiple clients (edge nodes). Clients exchange model parameters with central server iterating through the loop: central server → global model parameters → client→ update parameters using local data → central server → check termination conditions.

goal of model performance attacks is to slow down training time or to lower the accuracy of the collaborative model. The attack methods target the integrity of the data (data poisoning) or the quality and performance of the model (model poisoning or evasion attack) (Lyu et al., 2020a). Model performance attacks are of malicious nature. Defenses against model performance attacks methods have been addressed by differential privacy, robustness aggregation, and poisoning detection algorithms (Liu et al., 2022a, Polap & Woźniak, 2021). Defenses against evasion attacks include empirical defense approaches, mostly by hiding information (model fusion, gradient mask, or randomization) or adversarial training (see Liu et al., 2022b). Privacy attackers extract meaningful information from model parameters. These types of attacks include model inversion, membership inference, and generative adversarial network (GAN) reconstruction (Jere et al., 2020). Classification of attacks by the attack type, target, and phase, as well as by the attacker role and intention is shown in Table 1.

Current privacy protection schemes in FL can be divided into two categories: data encryption, and adding data disturbance (Dwork, 2008; Liu et al., 2013; Li et al., 2019; Yoshida et al., 2020). A scheme that preserves privacy by adding noise in sensitive data using user-level differential privacy (DP) framework in FL was proposed by Lu et al. (2019). A novel framework based on DP, in which artificial noise is added to parameters was proposed by Wei et al. (2020). However, introduction of noise results in the loss of global model accuracy (Li et al., 2021). The difference between the training data and test data can be reduced by modifying vital features in the training data, so that the model can be defended against MIA (Wang et al., 2019a). Privacy can be preserved through secret sharing protocol and blockchain technology (Weng et al., 2019b). Participants encrypt and upload gradient updates derived from local training, while updated parameters are obtained through a secret sharing protocol. Privacy in FL can be preserving using homomorphic encryption of transmitted data (Yang et al., 2021). In this solution, the encrypted data are sent to cloud servers, and the results are also transmitted to the user in an encrypted format, ensuring that privacy is not disclosed during the execution of the entire program. However, homomorphic encryption (Gentry, 2009; Cheon et al., 2017) only supports addition and multiplication operations. If the algorithm involves some non-polynomial calculations, using polynomial fitting to approximate the function value will markedly increase computation complexity.

By launching membership inference attack (MIA) on a federated model, a malicious client or server can identify whether a particular data record is in the datasets which participates in model training (Shokri et al., 2017; Melis et al., 2019). Source inference attack (SIA) is a natural extension of the membership inference attack, aimed to determine the source of records and the identity of the client owner (Hu et al., 2021). If a data record participates in model training, the attacker can obtain a candidate set of potential owners (Fig. 4). To narrow down the list of candidate owners, the malicious server can launch a SIA based on a successful MIA. Because SIAs can find connections between the training data and the individuals in real world, it has potential to harm the data owners. For example, after mapping medical data to specific individuals, attackers could implement highly targeted marketing intrusions to participants in the FL system. Attackers can create additional safety risks such as opportunities to defraud data owners based on the knowledge of the owner's health status. Such safety risks appear to be higher than the risks resulting from direct access to physiological data containing identity information because access to FL enables them to access predictive modeling results for participating individuals.

User privacy has two components: data privacy and identity privacy. In FL environment for training medically relevant ML models, data privacy refers to the physiological data information of users participating in model training, while identity privacy refers to the association between user physiological information and FL clients. In FRESH, even if an attacker launches MIA and determines that a record belongs to the training dataset, the harm to the user who owns the data record caused by the attack might be limited because the medical dataset does not contain user's identity information. On the contrary, since the disclosure of identity privacy connects data to users, an attacker can infer the health status of specific user and launch further attacks with increased precision and severity. In the FL scenario, because the clients perform local model training only, there is no knowledge about the identity of other clients and FL clients are less likely to launch SIAs. On the other hand, central server receives local model parameter updates from the participants in FL, and it can derive identity information from frequent interactions with clients (Hu et al., 2021). Because of access to client information, central server has a greater success rate in launching SIAs. MIAs may be difficult to detect because normal parameter aggregation and attacks can occur simultaneously. In FRESH, we assume that only an honest-but-curious central server can launch SIAs. In this scenario honest-but-curious server can infer the privacy data about using the intermediate results obtained in the training process (Yang et al., 2022). Thew attacker can infer the model parameters and reconstruct an identical model that can reveal sensitive information of other participants, compromising the privacy of participants.

Table 1
Classification of attacks in Federated Learning environment by type, targets, role, phase, and intention.

Attack type	Attack Target	Attacker Role	Attack Phase	Attacker Intention
Data poisoning	Model	Client	Training	Malicious
Model poisoning	Model	Client	Training	Malicious
Evasion attack	Model	Client	Predicting	Malicious
Membership inference	Privacy	Server/Client	Training/Predicting	Malicious/Semi-honest
Source inference	Privacy	Server	Training	Malicious/Semi-honest
Attribute inference	Privacy	Server/Client	Prediction	Malicious/Semi-honest
Data reconstruction	Privacy	Server	Training	Malicious/Semi-honest
Model inversion	Privacy	Server/Client	Training/Predicting	Malicious
Model extraction	Privacy	Server/Client	Predicting	Malicious

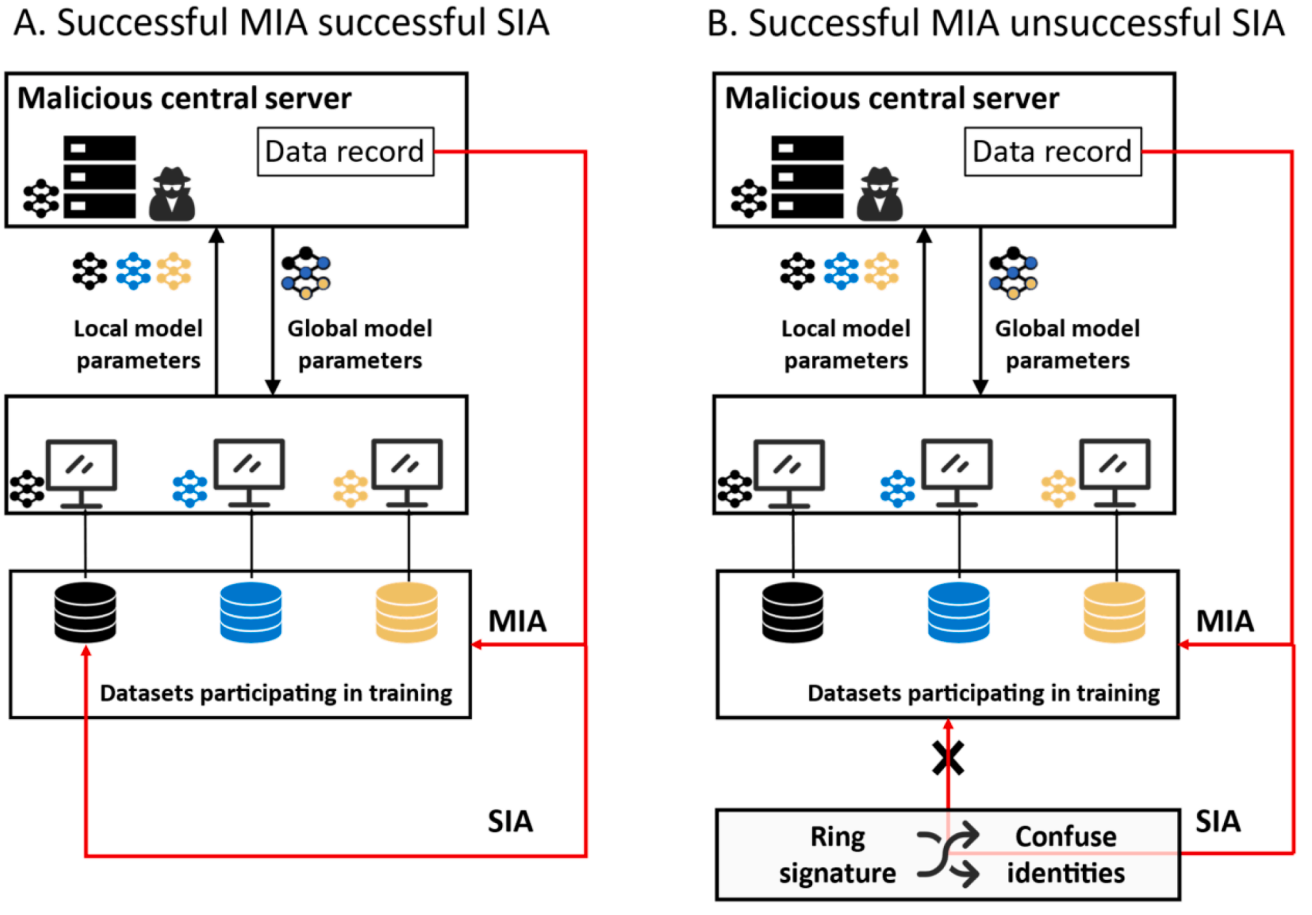


Fig. 4. MIA and SIAs attacks in federated learning environment. A. Datasets that participate in model training are separated between the nodes, making a SIA attack easy to succeed; B. Defending against SIAs using ring signature, where the signature of each client includes the characteristics of other participants, so that the attacker cannot identify the exact source of the data.

2.3. Ring signatures

Ring signature is a digital signature technology that allows verification that combines sender's public keys with other public keys, providing for the anonymity of all group members (Diffie & Hellman, 1976; Rivest et al., 1983; Schnorr, 1991). Ring signature was originally proposed to address the problem of signer anonymity (Rivest et al., 2001). Ring signature is different from group signature since it does not have the administrator node, and each member has equal status. The signer utilizes public key set and its own private key to construct the signature and sends both the signature and related parameters to the verifier. The verifier verifies the validity of the signature source by executing signature verification algorithm. During this process, the real identity of the signer is not visible to the verifier. A ring signature scheme based on elliptic curve and public key cryptosystem was proposed to reduce computational cost (Deng et al., 2020). This scheme takes public key verification of user groups as a part of hash input that verifies the correctness of signatures through hash calculation. When the original scheme is faced with multiple signatures from the same user group, its verification algorithm performs a series of multiplication operations. The analysis of computational cost has shown that the time needed for the elliptic curve multiplication is two orders of magnitude higher than the time needed for addition (Kittur & Pais, 2019). This property significantly reduces the efficiency of elliptic curve signature scheme. The limited efficiency is particularly pronounced when large number of users are in the same group.

2.4. The goal of this study

In this work, we propose an efficient and effective solution for SIA defense using the anonymity feature of ring signature. The performance evaluation of our solution, termed FRESH, shows that our solution meets the practical necessity in terms of time and performance. The ring signature algorithm ensures that the identity of each client includes the characteristics of others because public keys of all participants in a group are involved in signing. The identity feature confusion caused by ring signature is an efficient method for resisting SIAs. We propose batch verification to additionally support security by the ring signature scheme. This mechanism reduces time complexity of the signature verification algorithm from linear to constant and ensures that the additional computation cost to the server is minimal. The ring signature scheme offers a powerful privacy protecting mechanism, however when the number of clients is increasing, the computational complexity grows rapidly. We improved the ring signature scheme originally reported by Deng et al. (2020). Our elliptic curve scheme significantly reduces the number of multiplication operations during signature verification thus improving the efficiency of overall FL process. In FRESH, all signatures must be verified by the signature server. Therefore, the computing ability of signature server is an important factor to consider for the assessment of system performance. Because our new scheme retains linear operation of elliptic curve, the server can verify multiple signatures simultaneously. Experimental comparison and analysis indicated that the efficiency of verification in FRESH is significantly improved by batch verification. Our goal was to reduce the computing load of server node and to improve the overall efficiency of the system. Experiments were performed to assess time complexity of our algorithm and to achieve O(n) level of computational complexity when verifying multiple signatures.

3. System architecture

The abbreviations used in this work are shown in Table 2. FRESH consists of wearable devices, clients, signature server, and aggregation server (Fig. 5). The specific functions are described in detail in this section.

The construction of SIAs is based on Bayesian perspective, defined hereby:

Givenθ_{tk} and d_1, SIAs aim to infer the posterior probability of $d_1$belonging to u_i.The equation is:

$$S(\theta_{ti}, d_1) = \mathrm{E}_{\varepsilon}\left[\sigma\left(\log\left(\frac{\mathrm{P}(\theta_{ti}|s_{1k}=1, d_1, \varepsilon)}{\mathrm{P}(\theta_{ti}|s_{1k}=0, d_1, \varepsilon)}\right)+\mu_{\lambda}\right)\right] \tag{1}$$

where $\mu_{\lambda} = \log\left(\frac{\lambda}{1-\lambda}\right)$ and $\sigma(\cdot)$ is the sigmoid function. The s_j is an n-dimensional vector, indicating the owner of d_j.s_{ji} =1 means d_j

Table 2
Symbols and their definition used for various entities described in this work.

Symbol	Definition/Description	Symbol	Definition/Description
U	User set in a group	G	The generator of E_1
u_i	Any user, the i^{th} element in U	Zg^*	The multiplication group of modulo g
D	Training datasets	$H1{\sim}H3$	Three safe hash functions
D_i	Local dataset of u_i	v	System master key, $v \in Z_g^*$
d_j	The j^{th} record in D	G_{pub}	Public key of the system, $G_{pub} = vG$
m	The total number of records in D	t_i	Secret value of u_i, $t_i \in Z_g^*$
θ_{t-1}	Input global parameter for round t	T_i	Partial public key of u_i, $T_i = t_iG$
θ_{ti}	Local parameter set update by u_i in round t	PK_i	Public key of u_i, $PK_i = (T_i, R_i)$
S	Security parameters entered by the system	gp_i	the i^{th} group in FRESH
F_p	A finite field with prime P as module	pub_i	Public key set of gp_i, $pub_i = \{PK_1, \ldots, PK_n\}$
E	Elliptic curve based on Fp	Msg	Message to be signed
E_1	Additive cyclic subgroup from points on E	Ω	A ring signature
G	The order of E_1, large prime number $g > 2^s$	L_j	The set of users and public keys in group j, $L_j = \{(u_i, PK_i) \mid u_i \in U, PK_i \in pub_j\}$

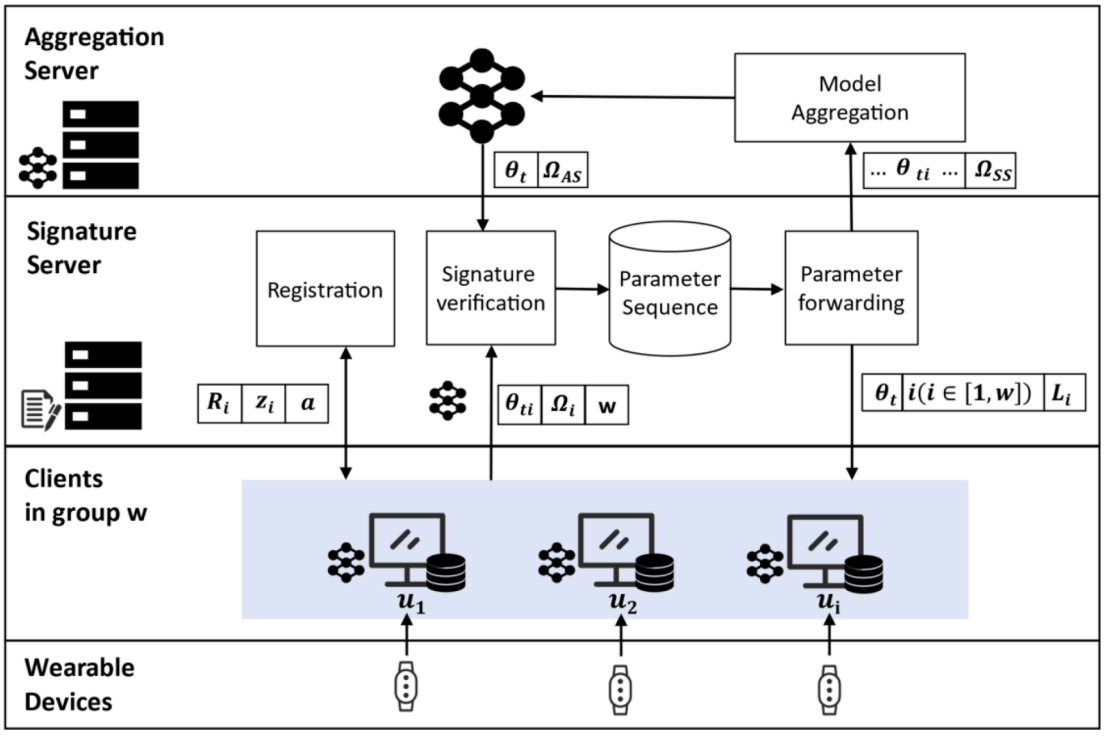


Fig. 5. The architecture of FRESH. The figure contains four types of entities. Wearable devices collect physiological data and send them to the clients. The main function of each client is to train local model θ_{ti} and sign it. The signature Ω is uploaded to the signature server to be verified by the signature server. Multiple signatures are sent to the aggregation server packaged together, to reduce the communication overhead. Aggregation server executes model averaging algorithm and obtains global model parameter θ_t. The dynamically generated group id (i) represents a group of clients that participate in the subsequent round of training.

belongs to u_i. The $\varepsilon = \{d_2, \cdots, d_n, s_2, \cdots, s_n\}$ denotes the data privacy and identity privacy information already known to the central server before the attack occurred.

3.1. Wearable devices

The main function of wearable devices in FRESH is data collection. Wireless terminal devices are highly integrated – their main function is health monitoring. In FRESH, wearables perform dynamic monitoring function of physiological signs, periodically collects users' physiological information (such as blood pressure and heart rate) and aggregate these data into private health data sets used for federated learning. The collected information is aggregated into user private health dataset at the federated learning client.

3.2. Clients

Clients are mainly responsible for training local models, signing local parameter updates, and uploading parameter updates along with the signature. In the proposed FL framework, mobile devices (e.g., mobile phones, tablet PCs) with considerable computer resources and storage capacities act as clients. Let $U = \{u_i\}_{i=1}^{n}$ denote the user set in any client groups, and u_i corresponds to client I, as shown in Fig. 5. All users in FRESH have equal status and each of them performs basically the same set of operations. We need to define common operations for all users based on the u_i representing any user in FRESH. When the client u_i participates local model parameters to joint training for the first time, it must first register to join a group. In this situation, u_i can use the public key set of the group to sign the message. Model updates submitted by unregistered client cannot be assessed as valid by signature server. After registering, the client starts to train the model and assign parameters to the locally trained model.

In registration phase, the client first applies for partial secret key information (R_i, z_i) and the group id. z_i is partial private key to be used in signature operation. The detailed operations of signature server related to user registration are discussed in Section 3.3. In the next step u_i selects $t_i \in Z_g^*$ as partial secret value and executes $t_i G$ to obtain partial public key T_i . Finally, u_i obtains public key $PK_i = (T_i, R_i)$ and sends $\{PK_i$,group id$\}$ information to the signature server. During the process done by a client, other users in the system do not know the owner of the public key, providing initial protection of the user's identity privacy.

When the user u_i is selected to participate in model training, in round t, u_i receives θ_{t-1} and trains local ML model (based on the local dataset D_i) to get θ_t.The ML algorithm in FRESH is based on logistic regression (Kleinbaum et al., 2002; Menard, 2002). It is a

standard method of supervised machine learning to classify data, and it has been successfully used in medical applications, for example to help determine whether a patient has a disease (Aono et al., 2016; Latifah et al., 2020). The u_i performs gradient descent algorithm to update local model, and finally obtains parameter update. After the preceding operations are complete, the user signs the parameter update and uploads it to the signature server. Participants use public key set pub_j to sign the update of parameters. We assume that the signer is $u_\lambda \in U$, and the message to be signed is $\theta_{t\lambda}$. Our signature generation procedure has six steps:

Step 1: Computing k_i by considering different values of i.

$$k_i = H_1 \parallel u_i \parallel R_i \parallel,\ 1 \le i \le n \text{ and } i \ne \lambda \tag{2}$$

Step 2: Selecting random point $B_i \in Z_g^*$ on the elliptic curve and compute Formulas 3 and 4 for different i.

$$a_i = H_2\left(\theta_{t\lambda} \parallel L_j \parallel B_i\right),\ 1 \le i \le n \text{ and } i \ne \lambda \tag{3}$$

$$l_i = H_3(\theta_{t\lambda} \parallel a_i \parallel u_i \parallel PK_i), 1 \le i \le n_i \text{ and } i \ne \lambda \tag{4}$$

Step 3: Selecting a random number $v \in Z_g^*$ and compute Formula 5.

$$A = sG + \sum_{i=1, i\ne\lambda}^{n} a_i\left(l_i T_i + R_i + k_i G_{pub}\right) \tag{5}$$

Step 4: Selecting a random number $b_\lambda \in Z_g^*$ and compute formulas 6 and 7.

$$B^{'} = A + b_\lambda G \tag{6}$$

$$B_\lambda = B^{'} - \sum_{i=1, i\ne\lambda}^{n} B_i \tag{7}$$

Step 5: Computing three formulas in turn.

$$a_\lambda = H_2\left(\theta_{t\lambda} \parallel L_j \parallel B_\lambda\right) \tag{8}$$

$$l_\lambda = H_3(\theta_{t\lambda} \parallel a_\lambda \parallel u_\lambda \parallel PK_\lambda) \tag{9}$$

$$\tau = s + b_\lambda - a_\lambda(l_\lambda t_\lambda + z_\lambda) \tag{10}$$

Step 6: Producing output signature.

Signature server

Inputs: $\boldsymbol{\theta_{ti}}, \Omega$

Outputs: parameters sequence, Ω_{ss}

```
send {R_i, z_i, a} to u_i
    if getMessages() includes θ_ti do
        verify the signature
        if verify qualified do
            params_seq[i++] ← θ_ti
        end if
    end if
    if meet send conditions
        send params_seq[0...n] to AS
    end if
return
```

Fig. 6. The description of the signature server task. Signature server sends partial secret key information (R_i, z_i) and to initialize the signature parameters of the client. The signature server does not immediately upload the signature to the aggregation server but places it into a sequence. When the trigger conditions are met, a packaged series of signatures is sent to the aggregation server, to reduce communication load.

$$\Omega = (\tau, \theta_{i\lambda}, a_1, \ldots, a_n, B^{'}) \tag{11}$$

At the end, the user u_λ uploads signature Ω to the signature server.

3.3. Signature server

The signature server (*SiS*) acts as the trusted broker server in FRESH. *SiS* coordinates the data interaction between nodes. The specific functions of *SiS* include initializing signature parameters, sending partial secret key information, grouping users, verifying signature and forwarding parameters (Fig. 6).

Initialization of signature parameters

Some parameters used in signature and verification are generated in the phase using steps:

Step 1:*SiS* selects an elliptic curve group E_1 whose order is g and the generator is G. g is a big prime number meeting a condition $g > 2^s$.

Step 2:*SiS* selects secure hash functions $H_1, H_2, H_3 : \{0,1\}^* \rightarrow Z_g^*$.

Step 3:*SiS* selects system master key $v \in Z_g^*$, and compute system public key $G_{pub} = vG$.

Step 4:SiS packages signature-related parameters into the system parameter list, *ParaList*. $ParaList = \{g, P, G, G_{pub}, H_1, H_2, H_3\}$. After packaging, *ParaList* is distributed to the clients.

User group policy

In ring signature architecture, public keys of all users participate in signing. Since all users contribute their public keys, the more public keys in the public key set, the more time is needed for signing and verifying. Excessive size of the public key set is a great challenge that needs to be addressed to ensure efficient system performance. Conversely, if the size of public key set is controlled by limiting the number of users, signature and verification time-consuming will be reduced, resulting in low accuracy of the model because of the small set of data used for training ML models. We devised a grouping policy that reconciles the size vs. accuracy contradiction. *SiS* divides all users into $GP = \{gp_j\}_{j=1}^{w}$ according to the grouping policy. With increasing number of participants joining FRESH, the value of w grows. The pub_j is the public key set of user group gp_j, containing the public keys of all users in gp_j. When users sign message, they use only the public keys in pub_i, not all of public keys in FRESH, significantly reducing the signing and verification time in each round. Only one group participates in model training in each round, and *SiS* determines the group *id* by generating a random number. The number of users n is highly correlated with the time cost in signing and verification – this is illustrated and described in detail in Sections 5.1 and 5.2. The relationship between system security performance and n, is discussed in Section 5.3.

Specifically, *SiS* manages groups through a table that contains mapping information between gp_j and pub_j. When *SiS* receives the request of partial secret key from u_i, it queries the table for groups with fewer than n users. At most one group meets the requirements, and others must only have n users. If all groups have n users, *SiS* will build a new group. The group meets the condition is the latest group pub_w. Then, *SiS* selects a random number $r_i \in Z_g^*$, calculates $R_i = r_i$G, and stores R_i to pub_w. After storing, *SiS* calculates $k_i = H_1(u_i \| R_i)$ and $z_i = r_i + k_i v$ sequentially. Finally, $\{R_i, z_i,$ group id$\}$ is returned to the user u_i . When *SiS* receives data tuple $\{PK_i,$ group id$\}$ from u_i, PK_i will be stored in the corresponding public key set.

Verification of signature

A) Single signature verification. After receiving the signature Ω from the client u_λ, *SiS* performs the following steps to verify the signature.

Step 1: Computing k_i according to Formulas 12 and 13 for different i.

$$k_i = H_1(u_i \| R_i, \ 1 \le i \le n \tag{12}$$

$$l_i = H_3(\theta_{i\lambda} \| a_i \| u_i \| PK_i), \ 1 \le i \le n \tag{13}$$

Step 2: Determining whether formula 14 is satisfied.

$$B^{'} = \tau G + \sum_{i=1}^{n}(a_i l_i T_i + a_i R_i) + \left(\sum_{i=1}^{n} k_i a_i\right) G_{pub} \tag{14}$$

When Eq. (14) is satisfied, the signature is valid, and the verification succeeds.

b) Multi-signature verification. Given that there are $n^{'}$ different signatures from a user group, and msg_j is the signed message. The signature is shown as:

$$\Omega_j = \left(\tau_j, msg_j, a_i^j, \ldots, a_n^j, B_j^{'}\right), \ i = 1, 2, \ldots, n \ j = 1, 2, \ldots, \ n^{'} \tag{15}$$

SiS then performs the following steps to verify multi-signature in batches:

Step 1: Computing formula 16 and formula 17 in different i.

$$k_i = H_1(u_i \| R_i),\ i = 1, 2, \ldots, n \tag{16}$$

$$l_i^j = H_3\left(msg_j \| a_i^j \| u_i \| PK_i\right),\ i = 1, 2, \ldots, n\ j = 1, 2, \ldots,\ n' \tag{17}$$

Step 2: Determining whether formula 18 is satisfied.

$$\sum_{j=1}^{n'} B_j' = \left(\sum_{j=1}^{n'} \tau_j\right) G + \sum_{i=1}^{n}\left[\left(\sum_{j=1}^{n'} a_i^j l_i^j\right) T_i + \left(\sum_{j=1}^{n'} a_i^j\right) R_i\right] + \left[\sum_{i=1}^{n} k_i \left(\sum_{j=1}^{n'} a_i^j\right)\right] G_{pub} \tag{18}$$

When Eq. (18) is satisfied, a batch of signatures are valid and multiple signatures are validated.

Parameter forwarding

SiS separates clients from the aggregation server (*AS*); the two are invisible to each other in FRESH. This architecture ensures that the information about the user's identity is shielded from *AS*, increasing the difficulty for *AS* to launch inference attacks. *SiS* is responsible for forwarding data between *AS* and clients. When the signature is proven to be effective, *SiS* puts the signed model update θ_{ti} into the local parameters sequence. The sending policy is satisfied when the number of parameters or the waiting time reach predefined thresholds. After sending policy is satisfied, *SiS* uses its private key to sign the parameter queue, and the signature is Ω_{ss}. Then, *SiS* sends parameters and Ω_{ss} to the *AS*. When *SiS* receives the latest global parameter θ_t from *AS*, it uses the public key of AS to verify that the parameters are not tampered with. Assuming that there are *w* groups, *SiS* generates a random number w' between 1 and *w* to serve as id of the selected group, for participation in the next round of model training. Finally, data tuple $\{w', L_{w'}, \theta_t\}$ is distributed to each client. The user of $gp_{w'}$ will train disease prediction model jointly, others do nothing else. During the communication between *SiS* and *AS*, the signature of the message ensures that no malicious node can impersonate *SiS* or *AS* to launch a poison attack. A series of parameter updates are sent at one time, reducing the communication overhead.

3.4. Aggregation server

AS obtains current global model parameter θ_t by executing model averaging algorithm (Fig. 7). When the loss function converges, the training stops, otherwise *AS* uses its own private key to sign θ_t, to get Ω_{as}. *AS then* sends θ_t and Ω_{as} to the *SiS* for the next round of

Model averaging algorithm

```
Inputs: parameter queue
Outputs: current global parameter, Ω_as
1.  params_seq[0…n-1] ← get_params_seq()
2.  initialize params_weight[0…n-1]
3.      for k=0 to n-1 do
4.              current_param ← current_param+
                params_weight[k]*params_seq[k]
5.      end for
6.      if loss function is convergent then
7.              obtain the model
8.              return
9.      end if
10. signature ← get_signature(current_param,sk)
11. send  the signature and current_param to SiS
12. Return
```

Fig. 7. The task of aggregation server. Model averaging algorithm multiplies each local parameter by its corresponding weight to obtain a series of intermediate global models. The loss function is expected to converge after several rounds of training and the final global model is generated.

training.

4. Identity privacy and algorithm security analysis

We performed theorical analysis of our FRESH solution to assess the privacy and security aspects. The defense against SIAs is based on ring signature mechanism. The probability of identifying the data owner from a group with n signers is n^{-1}. Security of our improved ring signature algorithms was analyzed under two game scenarios, as described in Subsections 4.1 and 4.2.

4.1. Identity privacy analysis

FRESH utilizes unconditional anonymity of ring signature (Fang et al., 2019; Zhang et al., 2019) to resist SIAs. In contrast to traditional FL systems, each user in FRESH adds noise to its identity through ring signature. From the point of view of *SiS*, each signature corresponds to a group of possible signers rather than a specific user. The *AS* and clients are completely separated. Since *AS* and clients do not interact directly, in the ideal case, *AS* has no knowledge about the identity of the client, which directly prevents SIAs (Figs. 4 and 5). Model training and parameter aggregation are independent of the *SiS*. Theoretically, *SiS* is invisible to both the details of the model aggregation and the global model. Hence, MIA cannot be launched, preventing SIAs. In the worst-case scenario, *SiS* and *AS* cooperate to steal users' identity privacy. In that case, *AS* contributes model information, and *SiS* contributes client information. Because of the anonymity attribute of ring signature technology, the identity of users in FRESH is represented as a group of users who participate in signing. If the attack is successful, the malicious node can only infer the group that the owner of record may belong to but cannot identify the specific owner.

In our scheme, because $r_i \in Z_g^*$ is a random number, $R_i \in E_1$ satisfies equal probability distribution. Both $k_i \in Z_g^*$ and $l_i \in Z_g^*$ are generated by hash functions, so they satisfy equal probability distribution too. Similarly, since $s \in Z_g^*$ is a random number, $A \in E_1, \tau \in Z_g^*$ satisfies equal probability distribution. We can, therefore, draw a conclusion that the probability that adversary will correctly infer the signer from a candidate group is n^{-1}. Here, n is the number of users whose public keys participate in the signing.

4.2. Algorithm security analysis

The analysis in Section 4.1 shows that the security of ring signature algorithm is an important factor for user identity privacy. This section focuses on the analysis of the security of improved ring signature algorithm. The proof of security is based on the random fable model. Because the proposed FRESH scheme is based on the Elliptic Curve Cryptosystem, each key belonging to the users in this system consist of two parts. One part is generated by *SiS* and the other by the users. To assess the privacy and security performance the system constructed in this study, assuming the system could be attacked under two games.

The adversary A_1 in Game I knows the secret value of users but not the partial key, and A_1 has the ability to replace partial public key of a user. The adversary A_2 in Game II obtains the partial private key of a user but did not know the user's secret value. A_1 and A_2 are polynomially bounded adversaries and intend to forge a signature in both Game I and Game II. The adversary is considered to win the corresponding game if the attempt to forge legal signature is successful.

Under the stated rules and assumptions, the security requirements of FRESH scheme in the random oracle model are defined as follows:

Definition. I: A FRESH scheme is unforgeable if the advantage of A_1 and A_2 are negligible in Games I and II.

Game I: Suppose the challenger C needs to solve a random instance (G, aG) of the *DL* problem, C invokes subroutine A_1 and serves as a Challenger of A_1 in Game I. C has to simulate an attack environment for A_1 and interacts with A_1 by performing a four-step process:

Step 1:Initialization. C runs the system initialization algorithm to generate parameter list $paraList = \{g, P, G, G_{pub}, H_1, H_2, H_3\}$, and responds to A_1 with this list.
Step 2:Queries. The adversary A_1 is a role with bounded compute ability, and A_1 will query C a limited number of times. C will respond to A_1 and matain multiple lists for storing those questions and answers. All lists are empty at the beginning.

(1) **Public key queries.** A_1 must perform public key queries before other queries. The challenger C maintains list L_{pk} to store every question and the corresponding answers. The structure of L_{pk} is (u_i, x_i, r_i). For each query (u_i) from A_1, the challenger C will perform steps as follows:
  1.1 If there is no result stored in L_{pk}, C will randomly select partial secret value x_i (equivalent to the t_i), $r_i \in Z_g^*$ and calculate $PK_i = (x_i G, r_i G)$. Then C returns this result to A_1, and puts both u_i and PK_i into L_{pk}.
  1.2 If L_{pk} already has context about the query from A_1, C will read corresponding answer to A_1.

In step 1.2, C returns $PK_\omega = (x_\omega G, aG)$ for one of the queries (u_ω).

(2) **Hash queries.** The challenger C creates three lists L_{H1}, L_{H2}, L_{H3} to store context related to hash queries. For each query (H_i) from A_1, the challenger C will perform the following steps:

2.1 When A_1 queries $H_1(\alpha_i)$, C reads list L_{H1} and then sends the corresponding result to A_1. If there is no record to answer $H_1(\alpha_i)$, C will randomly choose $k_i \in Z_g^*$, set $H_1(\alpha_i) = k_i$, and add (α_i, k_i) to L_{H1}. Then C responses to A_1 with k_i.

2.2 When A_1 queries $H_2(\beta_i)$, C reads list L_{H2} and then sends the corresponding result to A_1. If there is no record to answer $H_2(\beta_i)$, C will randomly choose $a_i \in Z_g^*$, set $H_2(\alpha_i) = a_i$, and add (β_i, a_i) to L_{H2}. Then C responds to A_1 with a_i.

2.3 When A_1 queries $H_3(\gamma_i)$, C reads list L_{H3} and then responds sends the corresponding result to A_1. If there is no record to answer $H_3(\gamma_i)$, C will randomly choose $l_i \in Z_g^*$, set $H_3(\gamma_i) = l_i$, and add (γ_i, l_i) to L_{H3}. Then C responds to A_1 with l_i.

(3) **Secret value of user's queries.** C maintains a list $L_x = (u_i, x_i)$ to store user's u_i and his secret value x_i. For all queries of A_1, C performs the following steps:

3.1 If L_x has no record about these queries, the challenger C will query L_{pk}, return x_i, and store (u_i, x_i) in L_x.

3.2 If L_x has corresponding result about this query, C seeks L_x and returns x_i to A_1.

(4) **Partial private key queries.** C maintains $L_z = (u_i, R_i, z_i)$ to respond to A_1 about his partial private key queries. The action of C includes query L_{pk} and L_{H1}, calculate (R_i, z_i) by running private key calculation algorithm, and add (R_i, z_i) to L_z. When querying message about u_ω, C has no ability to calculate the result, the game fails.

(5) **Partial public key replacement queries.** C maintains $\mathrm{L_T} = (u_i, T_i, T_i^{'})$, recording adversary A_1 replaces public key T_i with new public key $T_i^{'}$.

(6) **Signature queries.** Supposing there is a signer u_λ, $u_\lambda \in L$, public key and user set. If the condition is $u_\lambda \neq u_\omega$, C will run the signature algorithm in the scheme to generate ring signature Ω after receiving (msg, L) from A_1. Otherwise, C generates signature Ω using the algorithm:

1) Calculate $k_i = H_1(u_i \| R_i),\ i = 1, 2, \ldots, n$

2) Choose $B_i \in E_1$ randomly, and calculate:

$$a_i = H_2(msg \| L \| B_i)\ ,\ \mathrm{i} = 1, 2, \ldots, \lambda - 1, \lambda + 1, \ldots, \mathrm{n} \tag{19}$$

$$l_i = H_3(msg \| a_i \| u_i \| PK_i),\ \mathrm{i} = 1, 2, \ldots, \lambda - 1, \lambda + 1, \ldots, \mathrm{n} \tag{20}$$

3) Randomly choose $\mathrm{a}_\lambda, \tau \in \mathrm{Z_g^*}$, then calculate:

$$l_\lambda = H_3(msg \| a_\lambda \| u_\lambda \| PK_\lambda) \tag{21}$$

$$B^{'} = \tau G + \sum\nolimits_{i=1}^{n} a_i \left(l_i T_i + R_i + k_i P_{pub}\right) \tag{22}$$

4) Output signature $\Omega = (\tau, msg, a_1, \ldots, a_n, B^{'})$ and send information to list $L_{sig} = (u_\lambda, msg, L, \sigma)$.

Step 3:Solve the discrete logarithm problem: Suppose that adversary A_1 can win Game I with advantage of ρ in finite time, two valid ring signatures $\Omega_1 = (\tau_1, msg, a_1, \ldots, a_n, B^{'})$ and $\Omega_2 = (\tau_2, msg, a_1^*, \ldots, a_n^*, B^*)$ can be generated with probability $\rho^2 / \varphi \mathrm{C}_{\mathrm{g_{H2}}}^{\mathrm{n}}$ (φ is a constant) from the bifurcation results of H_2 queries. They satisfy $a_\lambda \neq a_\lambda^*, a_i = a_i^* (i \neq \lambda)$. If they also satisfy $u_\lambda = u_\omega$,then the following equation was established:

$$\begin{cases} \tau = s + \mathrm{b}_\lambda - \mathrm{a}_\lambda(\mathrm{l}_\lambda \mathrm{x}_\lambda + \mathrm{z}_\lambda) \\ \tau^* = s + \mathrm{b}_\lambda - \mathrm{a}_\lambda^* \left(\mathrm{l}_\lambda^* \mathrm{x}_\lambda + \mathrm{z}_\lambda\right) \end{cases} \mathrm{z}_\lambda = \mathrm{a} + \mathrm{k}_\lambda \mathrm{v} \tag{23}$$

Then C can solve the *DL* problem by querying list L_{pk}, L_{H1}, L_{H3} and L_{sig} and using formula:

$$\mathrm{a} = \frac{(\tau - \tau^*) - \left(\mathrm{a}_\lambda^* \mathrm{l}_\lambda^* - \mathrm{a}_\lambda \mathrm{l}_\lambda\right) \mathrm{x}_\lambda}{\mathrm{a}_\lambda^* - \mathrm{a}_\lambda} - \mathrm{k}_\lambda \mathrm{v} \tag{24}$$

Step 4:Probability calculation. Suppose the number of ring users is n, let Q_{pk} and Q_z represent the number of public key query and the number of partial private key query, respectively. $Event_1, Event_2$ *and* $Event_3$ represent different events as follows:

$Event_1$:C responds correctly to the query request of the adversary A_1 every time.
$Event_2$:u_ω belongs to the signer set L.
$Event_3$: u_ω is the real signer.
The calculated results are:

$$P_1 = P(Event_1) = \left(Q_{pk} - Q_z\right) / Q_{pk} \tag{25}$$

$$P_2 = P(Event_2|Event_1) = n/(Q_{pk} - Q_z) \tag{26}$$

$$P_3 = P(Event_3|Event_1 \cap Event_2) = 1/n \tag{27}$$

$$P_{success} = (P_3 \cap P_2 \cap P_1) = 1/Q_{pk} \tag{28}$$

This result indicates that A_1 can win Game I with probability $\rho_1^2/\varphi Q_{pk} C_{g_{H2}}^n$ in polynomial time.

Game II: Suppose the challenger C still need to solve a random instance (G, aG) of the *DL* problem, C invokes subroutine A_2 and serves as a Challenger of A_2 in Game II. C needs to simulate an attack environment for A_2 and interact with A_2 by performing a four-step process:

Step 1:Initialization. C inputs security parameter 1^s, and runs the system initialization algorithm to generate parameter list $paraList = \{g, P, G, G_{pub}, H_1, H_2, H_3\}$, and responses to A_2 with this list and the secret key v of the system.

Step 2:Queries. The adversary A_2 is a role with bounded compute ability, and A_2 will query C a limited number of times. Because A_2 has the secret value of system, he can calculate partial keys of other users. Unlike A_1 and A_2 has no ability to replace public key of other users. C will response to A_2 and matain multiple lists for storing those questions and answers. All the lists are empty at the beginning.

(1) **Public key queries.** A_2 must perform public key queries before other queries. The challenger C maintains list L_{pk} to store every question and the corresponding answers. The structure of L_{pk} is (u_i, x_i, r_i). For each query (u_i) from A_2, the challenger C will perform steps as follows:
   1.1 If there is no result stored in L_{pk}, C will randomly select partial secret value x_i (equivalent to the t_i), $r_i \in Z_g^*$ and calculate $PK_i = (x_i G, r_i G)$. Then C returns this result to A_2, and puts both u_i and PK_i into L_{pk}.
   1.2 If L_{pk} already has context about the query from A_2, C will read corresponding answer to A_2.

In step 1.2, C returns $PK_\omega = (aG, r_\omega G)$ for one of the queries (u_ω).

(2) **Hash queries.** Same rules as in Game I.
(3) **Secret value of user's queries.** C maintains a list $L_x = (u_i, x_i)$ to store the user u_i and his secret value x_i. For all queries of A_2, C performs the following steps:
   3.1 If L_x has no record about these queries, the challenger C will query L_{pk}, return x_i, and store (u_i, x_i) in L_x.
   3.2 If L_x has corresponding result about this query, C seeks L_x and returns x_i to A_2.
   3.3 If A_2 issues the request about u_ω,C will refuse to response to A_2, and Game II fails.
(4) **Partial private key queries.** C maintains $L_z = (u_i, R_i, z_i)$ to response to A_2 about partial private key queries. The challenger C performs steps as follows:
   4.1 If L_z has no record about these queries, The actions C does is query L_{pk} and L_{H1}, calculating (R_i, z_i) by running private key calculation algorithm and adding (R_i, z_i) to L_z.
   4.2 If L_x has corresponding result about this query, C seek L_z and return (R_i, z_i) to A_2.
(5) **Partial public key replacement queries.** Same rules as in Game I.
(6) **Signature queries.** Same rules as in Game I.

Step 3:Solve the discrete logarithm problem: Suppose that adversary A_2 can win Game II with advantage ofρin finite time, two valid ring signatures $\Omega_1 = (\tau_1, msg, a_1, \ldots, a_n, B')and\Omega_1 = (\tau_2, msg, a_1^*, \ldots, a_n^*, B^*)$ can be generated with probability $\rho^2/\varphi C_{g_{H2}}^n$from the bifurcation results of H_2 queries. They satisfy $a_\lambda \neq a_\lambda^*, a_i = a_i^* (i \neq \lambda)$. If they also satisfy $u_\lambda = u_\omega$, then the following equation was established:

$$\begin{cases} \tau = s + b_\lambda - a_\lambda(l_\lambda a + z_\lambda) \\ \tau^* = s + b_\lambda - a_\lambda^*(l_\lambda^* a + z_\lambda) \end{cases} z_\lambda = r_\lambda + k_\lambda v \tag{29}$$

Then C can solve the DL problem by querying list L_{pk}, L_{H1}, L_{H3}, and L_{sig} using formula:

$$a = \frac{(\tau - \tau^*) - (a_\lambda^* - a_\lambda) z_\lambda}{a_\lambda^* l_\lambda^* - a_\lambda l_\lambda} \tag{30}$$

Step 4: Probability calculation: Suppose the number of ring users is n, let Q_{pk} and Q_z represent the number of public key query and the number of partial private key query, respectively, and $Event_1, Event_2$ *and* $Event_3$ represent different events as follows:

$Event_1$: The challenger C responds correctly to the query request of the adversary A_2 every time.

$Event_2$: u_ω belongs to the signer set L.
$Event_3$: u_ω is the real signer.
The calculated results are:

$$P_1 = P(Event_1) = \left(Q_{pk} - Q_x - Q_T\right)/Q_{pk} \tag{31}$$

$$P_2 = P(Event_2|Event_1) = n/\left(Q_{pk} - Q_x - Q_T\right) \tag{32}$$

$$P_3 = P(Event_3|Event_1 \cap Event_2) = 1/n \tag{33}$$

$$P_{success} = (P_3 \cap P_2 \cap P_1) = 1/Q_{pk} \tag{34}$$

This result indicates that A_2 can win Game II with probability $\rho_2^2/\varphi Q_{pk} C_{g_{H2}}^n$ in polynomial time.

5. Experimental settings and performance evaluation

In the experiment, we comprehensively evaluated signature efficiency, verification efficiency, anti-attack effect, and communication cost in FRESH. We used personal computer to simulate both the *SiS* and the client. The configuration of personal computer was OS: windows 10, CPU:2.8 GHz Dual-core Intel Core i5, Memory: 8GB 1600 MHz DDR3. The ring signature scheme was implemented using JPBC library (De Caro & Iovino, 2011) and using Java language. JPBC library implements bilinear mapping, and the elliptic curve we selected is *d159.*, a complete curve, with defined parameters and equations (gas.dia.unisa.it/projects/jpbc/index.html#.Yz416FO-sR4)

The purpose of this experiment was to gather statistics on basic operation time in ring signature scheme we developed and presented. These results provide data support for theoretical analysis of further experiments and point out the direction for further improvement of ring signature algorithm applications. It takes five operations to generate and verify the signature. These operations include, hash, scalar multiplication on E_1, addition operation on E_1, multiplication on Zg^* and addition operation on Zg^*. For brevity of description, they denoted by some symbols in Table 3. We performed each operation 1000 times and calculated the average time (Table 3). Raw data plots are available in Supplemental Fig. 1. These plots indicate that majority of basic operations time values are stable, but some outliers are present as expected in regular operation of computers. The outliers of individual values of basic operations are independent of each other. Our estimate is that the number of outliers is smaller than 5% of the total number of signature cycles.

The experimental results show that Scalar multiplication ME_1 takes about two orders of magnitude more time than addition, while the time of *MZ* and *AZ* is almost negligible. We can draw a conclusion that reducing the number of ME_1 provides a much greater efficiency boost than reducing the number of AE_1 in FRESH. Because operation time of *MZ* and *AZ* is very short, less than several microseconds and these two operations together account for only 0.14% of the total operation time, we decided to exclude these two operations from further analysis.

5.1. Signature time

The purpose of this experiment is to evaluate the impact of public key set size n on signature time spent by each user. The experiment was divided into 20 groups where the public key set size was different in each group while all other conditions were the same. The first group has a size of 10, each subsequent group has 10 more than the previous group, so the last group (number twenty) had size of 200. Each group of experiments was conducted 10 times, and signature time was obtained every time we completed an experimental round. Finally, we calculated the average of ten results for each group of experiments (Fig. 8). Experimental results show that the signature time increases linearly with the increase of the public key set size. By analyzing the signature algorithm (described in Section 3.2), The computation required to generate user signature is calculated by formula $(3n-1) \times H + (4n-3) \times AE_1 + (3n-1) \times ME_1$.

The time required to run the algorithm is

$$T_s = (3T_H + 4T_{AE_1} + 3T_{ME_1}) \times n - (T_H + 3T_{AE_1} + T_{ME_1}) \tag{35}$$

Table 3
Average time of basic operation in FRESH, from 1000 simulations, in milliseconds. The major contributor to the basic operation time is the Scalar multiplication on E_1, consuming 97% of the total operation time. σ: standard deviation, Min: minimum value, Med: median value, Max: maximum value.

Basic operation	Symbol	Average Time ± σ	Min, Med, Max
Hash operation	H	0.101±0.143	0.0197, 0.0751, 2.88
Scalar multiplication on E_1	ME_1	4.61±1.60	3.27, 4.06, 16.2
Addition operation on E_1	AE_1	0.0383±0.0681	0.0171, 0.229, 1.43
Multiplication on Zg*	MZ	0.00543±0.0221	0.013, 0.033, 0.582
Addition operation on Zg*	AZ	0.00141±0.0015	0.0005, 0.0011, 0.0274
Total basic operation time	T_{BO}	4.75±1.65	3.31, 4.20, 16.4

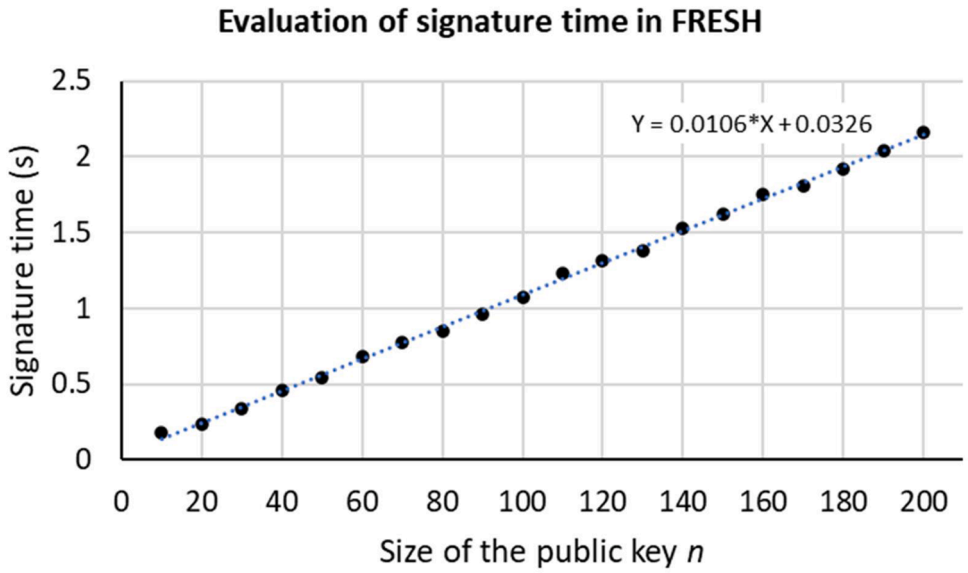


Fig 8. Signature time in FRESH. Twenty groups of different public key sizes were generated, ranging from 10 to 200, with increment of 10. The experiment was performed to determine signature time. Each point represents an average of 10 experiments.

The rate of change of Function 35 is constant: $3T_H + 4T_{AE_1} + 3T_{ME_1}$, The trend of time required for signature as function of the size of public key set is consistent with theoretical analysis results (Fig. 8).

The results of signature time experiment have important guiding role in understanding and solving the issue of client number vs. user signature time. The more users participate, the wider the data sources, and the higher the accuracy of the model. Access to high-quality data sets requires large number of users joining the FRESH. However, because all public keys in public key set participate in the signature operation, the increasing number of public keys will require longer time for users to sign. It is of great importance to design a mechanism that can satisfy a larger number of clients for joint training of prediction models and the improvement of the final training. For this reason we designed the user grouping mechanism in FRESH. The results in Fig. 8 can provide a reference for setting the number of users in each group. When n is less than 100, the time is less than 1.5 seconds, and we generally set n to 100 by default. System administrator can dynamically adjust n to achieve a new balance between time consumption and the duration of training cycles.

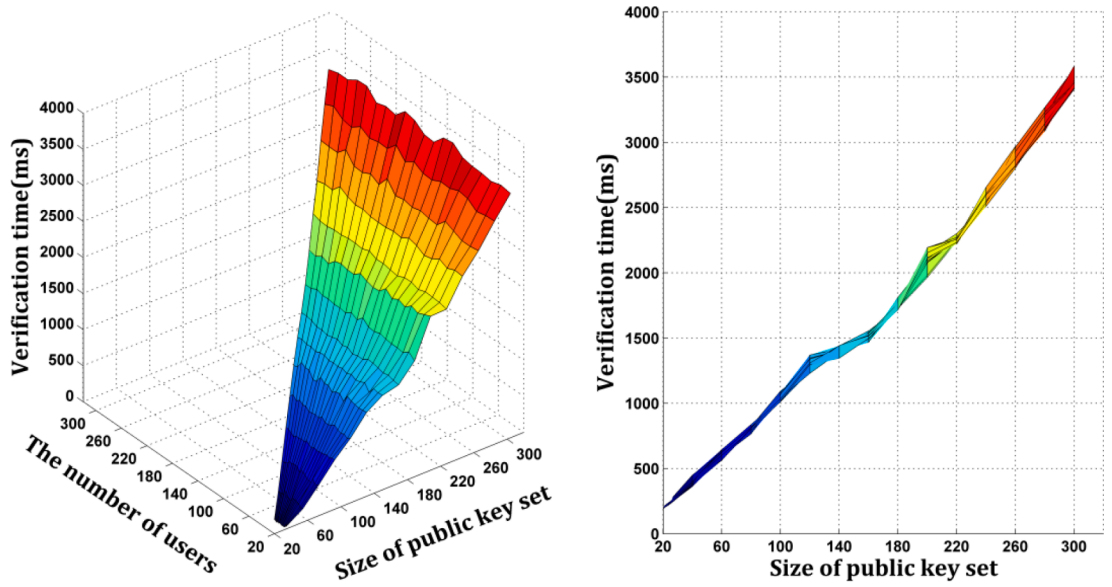


Fig 9. Batch verification time in FRSHE. When the size of public key set is fixed, the increase in the number of users increases verification time very little. On the other hand, when the number of users is fixed, the increase of public key set results in linear increase of verification time.

5.2. Batch validation time

In each training round, only one group of users participated in model training. Usually, not all users in the group jointly train a model. The main purpose of this experiment was to explore the relationship between the number of users participating in training (n'), and the signature verification time T_{BV}. The value of n is selected from the set {20, 40, 60, …, 280, 300}. There are 20 n' for each n, that are derived from set {1*n/20, 2*n/20, …, 19*n/20, n}. Each group of n corresponds to 20 n', a total of 300 combinations. The time consumption of each combination in batch validation was measured. The results of batch validation time are shown in Fig. 9.

We plotted the trend of change of validation time with changes of n and n' (Fig. 9). When n is constant, the change of T_{BV} is small with the increase of n', and the connections of all points almost remain level, indicating that no correlation was observed between n' and T_{BV} (Fig. 9 left panel). This feature is very important because it indicates it is possible to improve system performance (better privacy) and at same time it is possible to reduce data communication delay. T_V, non-batch verification time, will increase linearly with the increase of n', which will offset the gains from reduction of signature time by grouping policy. Even if T_s is limited within a reasonable range through grouping mechanism, if there is no batch validation, T_V will exceed limit when the number of users participating in the group increases. In these circumstances, the only option is to reduce the size of the public key set. However, small group of participants will greatly increase the number of training rounds, and at the same time diminish the security of the system afforded by the ring signature. On the other hand, when n' is constant, as n increases, T_{BV} increases linearly (Fig. 9, right panel). We performed a theoretical analysis to understand this behavior.

The analysis of single signature verification algorithm (described in Section 3.3), we can calculate the number of required computations by $(2\ \times n)\ \times H + (2\ \times n + 1)\ \times AE_1 + (2\ \times n + 2)\ \times ME_1$. When verifying the signatures of n' users, the total time is:

$$T_V = 2\ \times (T_H + T_{ME_1} + T_{AE_1}) \times n \times n' + (T_{AE_1} + 2\ \times T_{ME_1}) \times n' \tag{36}$$

The number of calculations for batch verification is $(1 + n')\ \times n \times H + (2n + n')\ \times AE_1 + (2n + 2)\ \times ME_1$. The total time is:

$$T_{BV} = T_H \times n \times n' + (T_H + 2\ \times T_{AE_1} + 2\ \times T_{ME_1}) \times n + T_{AE_1} \times n' + 2 \times T_{ME_1} \tag{37}$$

Taking the partial derivatives of formula 36 and Formula 37, we can get the following formulas.

$$\frac{\partial T_V}{\partial n} = 2\ \times (T_H + T_{ME_1} + T_{AE_1+}) \times n' \tag{38}$$

$$\frac{\partial T_V}{\partial n'} = 2\ \times (T_H + T_{ME_1} + T_{AE_1+}) \times n + (T_{AE_1} + 2\ \times T_{ME_1}) \tag{39}$$

$$\frac{\partial T_{BV}}{\partial n} = T_H \times n' + (T_H + 2\ \times T_{AE_1} + 2\ \times T_{ME_1}) \tag{40}$$

$$\frac{\partial T_{BV}}{\partial n'} = T_H \times n + T_{AE_1} \tag{41}$$

Although $\frac{\partial T_{BV}}{\partial n'}$ is not zero, it does not contain item T_{ME_1}. According to the experimental results in Table 3, $\frac{\partial T_{BV}}{\partial n'}$ is very small, which explains why T_{BV} has almost no amplitude of change along the n' direction. Moreover, $\frac{\partial T_V}{\partial n'}$ contains $2 \times (n+1) \times T_{ME_1}$, accounting for the majority of $\frac{\partial T_V}{\partial n'}$'s value. Therefore, T_V increases linearly in n' direction. Similar analyses explain the reasons why T_V and T_{BV} increase

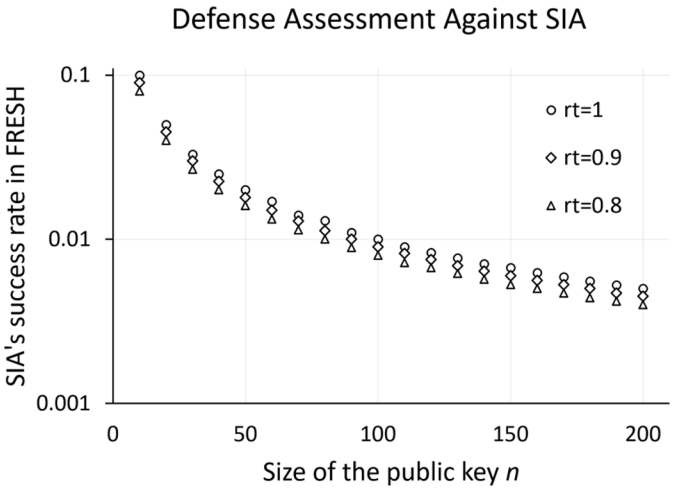


Fig 10. Defense assessment against SIAs. With the increase of n, the success rate of SIAs converges towards zero.

linearly in n direction. If we ignore T_H and T_{AE_1} because $T_V / T_{BV} \approx n'$, we can see that signature verification efficiency in FRESH is improved by n' times due to batch verification.

5.3. Defense effect evaluation

The defense effect against SIAs after introducing ring signature scheme is evaluated in this section. For the SIA attack method (Hu et al., 2021), we assume that the initial success rate is *rt*, and we set up the values of rt to 0.8, 0.9 and 1. After ring signature defense is introduced, the attacker can only identify the group to which the user belongs, reducing the success rate to *rt/n*. The success rate of SIA attacks immediately reaches the value that equals random guessing, and it keeps decreasing as the size of public key set n increases (Fig. 10). The SIA success rate curve converges rapidly and then slowly approaches zero. In the worst case, the initial success rate of the attack is 1. When n is greater than 10, the success rate after introducing ring signature algorithm is guaranteed to be no more than 10%. When the default public key set size is 100, theoretical SIA success rate drops to 1%. Once the time requirements of signing and verifying are satisfied, FRESH also meets the requirements of identity privacy security.

5.4. Evaluation of communication cost

Compared to the traditional FL framework, *AS* in FRESH adds only minor additional communication costs so we discuss the additional communication overhead on the client due to sign. For each round of training, the user uploads a signature $\Omega = (\tau, \theta_{t\lambda}, a_1, \ldots, a_n, B')$. In FRESH, the dimension of the user's physiological feature vector does not exceed 50, and we consider the maximum 50 parameters. So $\theta_{t\lambda}$ has a maximum dimension of 50 and occupies a maximum of 400 bytes (8 bytes $\times$ 50). The length of each field in the signature is composed of fields for τ (20 bytes), $\theta_{t\lambda}$ (400 bytes), a_i (20 bytes), and B' (40 bytes). The total length of signature is therefore 460+ 20 $\times$ n bytes. Additional communication cost corresponding to different public key set sizes n grows linearly because it is derived from formula $y = 0.16 \times n + 3.68$, where y is extra network rate in keys per second (kps), and n is the size of the public key set.

The extra network rate increases evenly with the expansion of the public key set size. The slope of formula $y = 0.16 \times n + 3.68$ is 0.16, indicating that when the user number of each group increases by 1, the additional network overhead required by uploading signature is only 0.16 (kps). The increase of the number of users in FRESH will not lead to the rapid growth of network demand. When the size of public key set is 100, the required extra network rate is not larger than 20 kps, a target that is very easy to achieve in WiFi or in 4G environments.

6. The practical implications of fresh

In the FL framework, MIA locates the owner of a record data to a group of participants, meaning that user's identity privacy is secure. However, malicious central server could precisely narrow the range of candidate owners to a single participant by launching SIAs. Because the information contained in the record will be associated with the people in the physical world, the record owners are vulnerable to various types of harm. This phenomenon is pronounced in the field of smart healthcare. Because medical data contain physiological information and health status of people, the value of medical data is high. The potential for financial gain may motivate the owner of malicious server to continuously launch SIAs. Once the identity privacy in smart healthcare system is leaked, users may suffer from various forms of real-life harm such as discrimination, fraud, extortion, or aggressive marketing, to name a few. Because of the potential for serious damage caused by leakage of identity and privacy breaches, the vulnerability to SIAs has become a serious bottleneck and a barrier to the development of FL.

In FRESH, ring signature scheme is used to defend against attacks, and batch validation is applied to improve the network communication throughput. Through experiments described in Section 5, we found that the success rate of SIAs declines at an exponential rate with the increase of the public key set size, but the signature time, batch validation time, and communication cost increase at a linear rate. Our results indicate that a small increase in time and communication costs can bring large practical improvement of the defense effectiveness. The scheme proposed in this paper successfully defends against SIAs and removes the hidden danger of identity privacy disclosure within a FL system. FRESH ensures that participants need not to worry about a variety of problems and possible financial losses caused by identity privacy disclosure.

The design of FRESH ensures that it is portable and can be applied to areas beyond smart healthcare. The algorithm that defends against SIAs is designed for universal federated framework, rather than for a specific scenario. We believe that FRESH may be easily expanded to help manage large FL environments, such as smart cities (Song et al., 2022).

7. Conclusion

With billions of wearable devices in use around the world, the amount of medical data owned by individuals has exploded (Zhang et al., 2015; Zhang et al., 2021). However, physiological data need to be converted into clinical information to realize the value of such data – this increasingly relies on the use of AI algorithms. The sensitive nature of medical data makes them difficult to share. We offer a solution for the problem of privacy protection during medical data sharing using FL. This paper first introduces the background and significance of medical data sharing and proposes to solve the difficulty of medical data sharing by using FL. We presented the framework of FRESH and analyzed its performance by simulation of two games. The experimental results show that FRESH can effectively resist SIAs with affordable communication and computing overhead.

The prospects of application of FRESH in practice are promising. The main advantages of FRESH are:

- It allows for build the models for disease diagnosis and prediction of disease progress by ML models that can be shared with interested parties. FRESH enables FL to break data sharing barriers between individuals that use smart wearable sensors. The key contribution of FRESH is ring signature technology that effectively protects user identity privacy at a minimal increase in data communication overhead. The residents of a community, or even a city, can collaborate to train disease prediction models using FRESH enabled systems. This will help improve health monitoring of individuals and at the same time contribute data to medical research for improving disease diagnosis and prognosis, without exposing participating individuals to privacy risks.
- FRESH can help realize real-time effective home-based medical care. Patients no longer need to go to the hospital for the measurement of signs of the diseases but can maintain home-based health monitoring. The physiological data collected by wearable devices enables the detection of early signs of disease that can be verified by healthcare professionals. This which is both convenient for patients and can greatly save the cost of healthcare.

There are some additional problems to be solved in the future:

- We plan to introduce incentive mechanism in FRESH based on blockchain technology, so that users who participate in training can derive benefits from sharing health data, motivating users to join FRESH.
- The prototype experiment results of the scheme proposed in this paper are satisfactory, but its efficiency needs to be further verified when used in large-scale applications with high real-time requirements.

The principal barriers to adoption of smart health technologies are reliability and efficiency of wearable devices, quality of physiological measurements, patient data privacy and security, and more recent requirements for legal and regulatory compliance (Baig & Gholamhosseini, 2013; Abouelmehdi et al., 2017; Zhang et al., 2022). The fresh technology contributes to the removing of these barriers particularly to patient data privacy and security and legal compliance. It also facilitates to the use of physiological measurements for precision medicine applications.

Credit author statement

Jindong Zhao with help from Vladimir Brusic designed the overall study. Xu Li and Xiuqin Qiu designed and implemented the ring signature scheme. Jindong Zhao and Wenshuo Wang designed the framework of FRESH, performed the experiments, and generated the results. Vladimir Brusic performed statistical analyses and provided medical context. Xiang Zhang performed the review of cybersecurity issues, discussed data issues, and described the smart health home concept and related issues.

Funding

This work was supported in part by the Shandong Provincial Natural Science Foundation, China under Grant ZR2020MF148, Shandong Provincial Natural Science Foundation, China under Grant ZR2020QF108 and National Natural Science Foundation, China under Grant 61972360. This work was supported by the Ningbo High-End Innovative Research Grant 2018A-08. Xiang Zhang is the recipient of the UNNC high-flier PhD scholarship 1903HFLY.

Data availability

The authors do not have permission to share data.

Acknowledgements

This work was done in part as internship training of Wenshuo Wang, Xu Li, and Xiuqin Qiu at Shandong Lengyan Medical Technology Inc.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ipm.2022.103167.

References

Abouelmehdi, K., Beni-Hssane, A., Khaloufi, H., & Saadi, M. (2017). Big data security and privacy in healthcare: A Review. *Procedia Computer Science, 113*, 73–80.
Ahmed, Z, Mohamed, K, Zeeshan, S, & Dong, X. (2020). Artificial intelligence with multi-functional machine learning platform development for better healthcare and precision medicine. *Database*, 2020. https://doi.org/10.1093/database/baaa010

Aono, Y., Hayashi, T., Trieu Phong, L., & Wang, L. (2016). Scalable and secure logistic regression via homomorphic encryption. In *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy* (pp. 142–144). https://doi.org/10.1145/2857705.2857731
Ashley, E. A. (2016). Towards precision medicine. *Nature Reviews Genetics, 17*(9), 507–522. https://doi.org/10.1038/nrg.2016.86
Atlam, H. F., & Wills, G. B. (2020). IoT security, privacy, safety and ethics. *Digital twin technologies and smart cities* (pp. 123–149). Cham: Springer. https://doi.org/10.1007/978-3-030-18732-3_8
Baig, M. M., & Gholamhosseini, H. (2013). Smart health monitoring systems: an overview of design and modeling. *Journal of medical systems, 37*(2), 1–14.
Banabilah, S., Aloqaily, M., Alsayed, E., Malik, N., & Jararweh, Y. (2022). Federated learning review: Fundamentals, enabling technologies, and future applications. *Information processing & management, 59*(6), Article 103061. https://doi.org/10.1016/j.ipm.2022.103061
Bilkey, G. A., Burns, B. L., Coles, E. P., Mahede, T., Baynam, G., & Nowak, K. J. (2019). Optimizing precision medicine for public health. *Frontiers in public health, 7*, 42. https://doi.org/10.3389/fpubh.2019.00042
Brahmecha, A., Sagathiya, M., Dalvi, R., & Halbe, A. (2021, September). LifeSpire: Detection and diagnosis of diseases. In *Proceedings of the 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)* (pp. 699–705). IEEE. https://doi.org/10.1109/ICIRCA51532.2021.9545064.
Bröring, A., Schmid, S., Schindhelm, C. K., Khelil, A., Käbisch, S., Kramer, D., & Teniente, E. (2017). Enabling IoT ecosystems through platform interoperability. *IEEE software, 34*(1), 54–61. https://doi.org/10.1109/MS.2017.2
Caprolu, M., Di Pietro, R., Lombardi, F., & Raponi, S. (2019). Edge computing perspectives: architectures, technologies, and open security issues. In *Proceedings of the2019 IEEE International Conference on Edge Computing (EDGE)* (pp. 116–123). IEEE. https://doi.org/10.1109/EDGE.2019.00035.
Cheon, J. H., Kim, A., Kim, M., & Song, Y. (2017). Homomorphic encryption for arithmetic of approximate numbers. In *Proceedings of theInternational Conference on the Theory and Application of Cryptology and Information Security* (pp. 409–437). Cham: Springer. https://doi.org/10.1007/978-3-319-70694-8_15.
De Caro, A., & Iovino, V. (2011). JPBC: Java pairing based cryptography. In *Proceedings of the 2011 IEEE symposium on computers and communications (ISCC)* (pp. 850–855). IEEE. https://doi.org/10.1109/ISCC.2011.5983948.
Demirkan, H. (2013). *A smart healthcare systems framework, 15* pp. 38–45). IT Professional. https://doi.org/10.1109/MITP.2013.35
Deng, L., Li, S., Huang, H., Jiang, Y., & Ning, B. (2020). Certificateless ring signature scheme from elliptic curve group. *Journal of Internet Technology, 21*(3), 723–731.
Dias, D., & Cunha, J. P. S. (2018). Wearable health devices—vital sign monitoring, systems and technologies. *Sensors, 18*(8), 2414. https://doi.org/10.3390/s18082414
Diffie, W., & Hellman, M. E. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, 644–654. https://doi.org/10.1109/TIT.1976.1055638
Dimitrov, D. V. (2016). Medical internet of things and big data in healthcare. *Healthcare informatics research, 22*(3), 156–163. https://doi.org/10.4258/hir.2016.22.3.156
Dunn, J., Runge, R., & Snyder, M. (2018). Wearables and the medical revolution. *Personalized Medicine, 15*(5), 429–448. https://doi.org/10.2217/pme-2018-0044
Dwork, C. (2008). Differential privacy: A survey of results. In *Proceedings of the International conference on theory and applications of models of computation* (pp. 1–19). Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-79228-4_1.
Fang, Y., Deng, J. Q., Cong, L. H., & Liu, C. Y. (2019). An improved scheme for PBFT blockchain consensus algorithm based on ring signature. *Computer engineering, 45*, 32–36. https://doi.org/10.19678/j.issn.1000-3428.0055794
Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing* (pp. 169–178). https://doi.org/10.1145/1536414.1536440
Hu, H., Salcic, Z., Sun, L., Dobbie, G., & Zhang, X. (2021). Source inference attacks in federated learning. In *Proceedings of the 2021 IEEE International Conference on Data Mining (ICDM)* (pp. 1102–1107). IEEE. https://doi.org/10.1109/ICDM51629.2021.00129.
Jia, Z., Wang, W., Wang, C., & Xu, W. (2017). Application and development of wearable devices in medical field. *China Medical Devices, 32*, 96–99.
Jere, M. S., Farnan, T., & Koushanfar, F. (2020). A taxonomy of attacks on federated learning. *IEEE Security & Privacy, 19*(2), 20–28.
Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., & Zhao, S. (2021). Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning, 14*(1–2), 1–210.
Kittur, A. S., & Pais, A. R. (2019). A new batch verification scheme for ECDSA* signatures. *Sādhanā, 44*(7), 1–12. https://doi.org/10.1007/s12046-019-1142-9
Kleinbaum, D. G., Dietz, K., Gail, M., Klein, M., & Klein, M. (2002). *Logistic regression* (p. 536). New York: Springer-Verlag.
Konečný, J., McMahan, H. B., Ramage, D., & Richtárik, P. (2016a). Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint*. https://doi.org/10.48550/arXiv.1610.02527. *arXiv:1610.02527*.
Konečný, J., McMahan, H. B., Felix, X. Y., Richtárik, P., Suresh, A. T., & Bacon, D. (2016b). Federated Learning: Strategies for Improving Communication Efficiency: Strategies for Improving Communication Efficiency. *arXiv preprint arXiv:1610.05492*. DOI: 10.48550/arXiv.1610.05492.
Kumar, R., Khan, A. A., Kumar, J., Golilarz, N. A., Zhang, S., Ting, Y., Zheng, C., & Wang, W. (2021). Blockchain-federated-learning and deep learning models for covid-19 detection using ct imaging. *IEEE Sensors Journal, 21*(14), 16301–16314. https://doi.org/10.1109/JSEN.2021.3076767
Latifah, F. A., Slamet, I., & Sugiyanto. (2020). Comparison of heart disease classification with logistic regression algorithm and random forest algorithm. In *, 2296*. *Proceedings of the AIP Conference Proceedings*. AIP Publishing LLC, Article 020021. https://doi.org/10.1063/5.0030579.
Le, D. N., Parvathy, V. S., Gupta, D., Khanna, A., Rodrigues, J. J., & Shankar, K. (2021). IoT enabled depthwise separable convolution neural network with deep support vector machine for COVID-19 diagnosis and classification. *International journal of machine learning and cybernetics, 12*(11), 3235–3248. https://doi.org/10.1007/s13042-020-01248-7
Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., Liu, X., & He, B. (2021). A survey on federated learning systems: vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*. https://doi.org/10.1109/TKDE.2021.3124599
Li, X., Huang, K., Yang, W., Wang, S., & Zhang, Z. (2019). On the convergence of FedAvg on non-IID data. *arXiv preprint arXiv:1907.02189*. DOI:10.48550/arXiv.1907.02189.
Liu, B., Ding, M., Shaham, S., Rahayu, W., Farokhi, F., & &Lin, Z (2021). When machine learning meets privacy: A survey and outlook. *ACM Computing Surveys (CSUR), 5*(2), 1–36, 54.
Liu, J., Huang, J., Zhou, Y., Li, X., Ji, S., Xiong, H., & Dou, D. (2022a). From distributed machine learning to federated learning: A survey. *Knowledge and Information Systems*, 1–33. https://doi.org/10.1007/s10115-022-01664-x
Liu, P., Xu, X., & Wang, W. (2022b). Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives. *Cybersecurity, 5*(1), 1–19. https://doi.org/10.1186/s42400-021-00105-6
Liu, Y., Muppala, J. K., Veeraraghavan, M., Lin, D., & Hamdi, M. (2013). *Data center networks: Topologies, architectures and fault-tolerance characteristics*. Springer Science & Business Media.
Lu, Y., Huang, X., Dai, Y., Maharjan, S., & Zhang, Y. (2019). Differentially private asynchronous federated learning for mobile edge computing in urban informatics. *IEEE Transactions on Industrial Informatics, 16*(3), 2134–2143. https://doi.org/10.1109/TII.2019.2942179
Lyu, L., Yu, H., Zhao, J., & Yang, Q. (2020a). Threats to federated learning. *Federated Learning* (pp. 3–16). Cham: Springer.
Lyu, M., Radenkovic, M., Keskin, D. B., & Brusic, V. (2020b). Classification of single cell types during leukemia therapy using artificial neural networks. In *Proceedings of the 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (pp. 1258–1261). IEEE.
Makkar, A., & Park, J. H. (2022). SecureCPS: Cognitive inspired framework for detection of cyber attacks in cyber–physical systems. *Information processing & management, 59*(3), Article 102914. https://doi.org/10.1016/j.ipm.2022.102914
Martin, Y. S., & Kung, A. (2018). Methods and tools for GDPR compliance through privacy and data protection engineering. In *Proceedings of the 2018 IEEE European symposium on security and privacy workshops (EuroS&PW)* (pp. 108–111). IEEE.
McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. *Artificial intelligence and statistics* (pp. 1273–1282). PMLR. https://doi.org/10.48550/arXiv.1602.05629
Melis, L., Song, C., De Cristofaro, E., & Shmatikov, V. (2019). Exploiting unintended feature leakage in collaborative learning. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)* (pp. 691–706). IEEE. https://doi.org/10.1109/SP.2019.00029.
Menard, S. (2002). *Applied logistic regression analysis, 106*. Sage. https://doi.org/10.4135/9781412983433

Molloy, J. C. (2011). The open knowledge foundation: open data means better science. *PLoS biology, 9*(12), Article e1001195. https://doi.org/10.1371/journal.pbio.1001195

Mothukuri, V., Parizi, R. M., Pouriyeh, S., Huang, Y., Dehghantanha, A., & Srivastava, G. (2021). A survey on security and privacy of federated learning. *Future Generation Computer Systems, 115*, 619–640. https://doi.org/10.1016/j.future.2020.10.007

Mukhopadhyay, S. C. (2014). Wearable sensors for human activity monitoring: A review. *IEEE sensors journal, 15*(3), 1321–1330. https://doi.org/10.1109/JSEN.2014.2370945

Nguyen, D. C., Pham, Q. V., Pathirana, P. N., Ding, M., Seneviratne, A., Lin, Z., Dobre, O., & Hwang, W. J. (2022). Federated learning for smart healthcare: A survey. *ACM Computing Surveys (CSUR), 55*(3), 1–37. https://doi.org/10.1145/3501296

Polap, D., & Woźniak, M. (2021). Meta-heuristic as manager in federated learning approaches for image processing purposes. *Applied Soft Computing, 113*, Article 107872. https://doi.org/10.1016/j.asoc.2021.107872

Rajula, H. S. R., Verlato, G., Manchia, M., Antonucci, N., & Fanos, V. (2020). Comparison of conventional statistical methods with machine learning in medicine: diagnosis, drug development, and treatment. *Medicina, 56*(9), 455.

Ramli, R., Zakaria, N., Mustaffa, N., & Sumari, P. (2012). Privacy issues in a psychiatric context: applying the ISD privacy framework to a psychiatric behavioural monitoring system. *IFAC Proceedings Volumes, 45*(10), 114–119.

Razaque, A., Amsaad, F., Khan, M. J., Hariri, S., Chen, S., Siting, C., & Ji, X. (2019). Survey: Cybersecurity vulnerabilities, attacks and solutions in the medical domain. *IEEE Access, 7*, 168774–168797.

Rivest, R. L., Shamir, A., & Adleman, L. (1983). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM, 26*(1), 96–99. https://doi.org/10.1145/359340.359342

Rivest, R. L., Shamir, A., & Tauman, Y. (2001). How to leak a secret. In *Proceedings of the International conference on the theory and application of cryptology and information security* (pp. 552–565). Berlin, Heidelberg: Springer. https://doi.org/10.1007/3-540-45682-1_32.

Saranya, P., & Asha, P. (2019). Survey on Big Data analytics in health care. In *Proceedings of the 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 46–51). IEEE. https://doi.org/10.1109/ICSSIT46314.2019.8987882.

Schnorr, C. P. (1991). Efficient signature generation by smart cards. *Journal of Cryptology, 4*(3), 161–174. https://doi.org/10.1007/BF00196725

Sheller, M. J., Edwards, B., Reina, G. A., Martin, J., Pati, S., Kotrotsou, A., & Bakas, S. (2020). Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific Reports, 10*(1), 1–12.

Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017). Membership inference attacks against machine learning models. In *Proceedings of the 2017 IEEE symposium on security and privacy (SP)* (pp. 3–18). IEEE. https://doi.org/10.1109/SP.2017.41.

Song, S., Fang, Z., & Jiang, J. (2022). Fast-DRD: Fast decentralized reinforcement distillation for deadline-aware edge computing. *Information processing & management, 59*(2), Article 102850. https://doi.org/10.1016/j.ipm.2021.102850

Summers, G., & Koehne, H. (2004). Data and databases. In H. Koehne (Ed.), *Developing Databases with Access* (pp. p4–p5). Nelson Australia Pty Limited.

Tuvshinjargal, B., & Hwang, H. (2021). Development of online service for brain disease prediction using machine learning. In *Proceedings of the 2021 International Conference on Information and Communication Technology Convergence (ICTC)* (pp. 505–508). IEEE. https://doi.org/10.1109/ICTC52510.2021.9620880.

Wang, C., Liu, G., Huang, H., Feng, W., Peng, K., & Wang, L. (2019a). MIASec: enabling data indistinguishability against membership inference attacks in MLaaS. *IEEE Transactions on Sustainable Computing, 5*(3), 365–376. https://doi.org/10.1109/TSUSC.2019.2930526

Wang, J., Spicher, N., Warnecke, J. M., Haghi, M., Schwartze, J., & Deserno, T. M. (2021 Jan 28). Unobtrusive health monitoring in private spaces: The smart home. *Sensors, 21*(3), 864.

Wang, R., Yu, S., Li, Y., Tang, Y., & Zhang, F. (2019b). Medical blockchain of privacy data sharing model based on ring signature. *Journal of UEST of China, 48*(06), 886–892. https://doi.org/10.3969/j.issn.1001-0548.2019.06.013

Wei, K., Li, J., Ding, M., Ma, C., Yang, H. H., Farokhi, F., & Poor, H. V. (2020). Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security, 15*, Article 34543469. https://doi.org/10.1109/TIFS.2020.2988575

Weng, J., Weng, J., Zhang, J., Li, M., Zhang, Y., & Luo, W. (2019). Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive. *IEEE Transactions on Dependable and Secure Computing, 18*(5), 2438–2455. https://doi.org/10.1109/TDSC.2019.2952332

Wu, D., Deng, Y., & Li, M. (2022). FL-MGVN: Federated learning for anomaly detection using mixed gaussian variational self-encoding network. *Information processing & management, 59*(2), Article 102839. https://doi.org/10.1016/j.ipm.2021.102839

Yang, J., Zheng, J., Zhang, Z., Chen, Q. I., Wong, D. S., & Li, Y. (2022). Security of federated learning for cloud-edge intelligence collaborative computing. *International Journal of Intelligent Systems, 37*(11), 9290–9308. https://doi.org/10.1002/int.22992

Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST), 10*(2), 1–19. https://doi.org/10.1145/3298981

Yang, Y., Zhang, Q., Zhang, Y., & Zuo, P. (2021). Design and implementation on homomorphic ciphertext fingerprint recognition system based on SEAL library. *Journal of Cryptologic Research, 8*(4), 616–629. https://doi.org/10.13868/j.cnki.jcr.000463

Yin, X., Zhu, Y., & Hu, J. (2021). A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. *ACM Computing Surveys (CSUR), 54*(6), 1–36. https://doi.org/10.1145/3460427

Yoshida, N., Nishio, T., Morikura, M., Yamamoto, K., & Yonetani, R. (2020). Hybrid-FL for wireless networks: Cooperative learning mechanism using non-IID data. In *Proceedings of the ICC 2020-2020 IEEE International Conference on Communications (ICC)* (pp. 1–7). IEEE. https://doi.org/10.1109/ICC40277.2020.9149323.

Zhang, X., Pike, M., Mustafa, N., & Brusic, V. (2022). Ethically informed software process for Smart Health Home. In *Proceedings of the 2022 IEEE 35th International Symposium on Computer-Based Medical Systems (CBMS)* (pp. 187–192). IEEE. https://doi.org/10.1109/CBMS55023.2022.00040.

Zhang, X., Zhang, J., Hang, C., & Tang, W. (2021). Verifiable statistical analysis scheme for encrypted medical data in cloud storage. *Computer Engineering, 47*(6), 32–37. https://doi.org/10.19678/j.issn.1000-3428.0058999

Zhang, Y., Qiu, M., Tsai, C. W., Hassan, M. M., & Alamri, A. (2015). Health-CPS: Healthcare cyber-physical system assisted by cloud and big data. *IEEE Systems Journal, 11*(1), 88–95. https://doi.org/10.1109/JSYST.2015.2460747

Zhang, Y., Song, T., & Zhang, Y. (2019). Password-based selective linkable convertible ring blind signature. In *, 1302. Proceedings of the In Journal of Physics: Conference Series*. IOP Publishing, Article 022039. https://doi.org/10.1088/1742-6596/1302/2/022039.

Original Paper

Advancing Privacy-Preserving Health Care Analytics and Implementation of the Personal Health Train: Federated Deep Learning Study

Ananya Choudhury[1,2*], MTech; Leroy Volmer[1,2*], MSc; Frank Martin[3], MSc; Rianne Fijten[1,2], PhD; Leonard Wee[1,2], PhD; Andre Dekker[1,2,4], PhD; Johan van Soest[1,2,4], PhD

[1]GROW Research Institute for Oncology and Reproduction, Maastricht University Medical Center+, Maastricht, Netherlands

[2]Clinical Data Science, Maastricht University, Maastricht, Netherlands

[3]Netherlands Comprehensive Cancer Organization (IKNL), Eindhoven, Netherlands

[4]Brightlands Institute for Smart Society (BISS), Faculty of Science and Engineering (FSE), Maastricht University, Heerlen, Netherlands

[*]these authors contributed equally

Corresponding Author:
Ananya Choudhury, MTech
GROW Research Institute for Oncology and Reproduction
Maastricht University Medical Center+
Paul Henri Spakalaan 1
Maastricht, 6229EN
Netherlands
Phone: 31 0686008485
Email: ananya.aus@gmail.com

Abstract

Background: The rapid advancement of deep learning in health care presents significant opportunities for automating complex medical tasks and improving clinical workflows. However, widespread adoption is impeded by data privacy concerns and the necessity for large, diverse datasets across multiple institutions. Federated learning (FL) has emerged as a viable solution, enabling collaborative artificial intelligence model development without sharing individual patient data. To effectively implement FL in health care, robust and secure infrastructures are essential. Developing such federated deep learning frameworks is crucial to harnessing the full potential of artificial intelligence while ensuring patient data privacy and regulatory compliance.

Objective: The objective is to introduce an innovative FL infrastructure called the Personal Health Train (PHT) that includes the procedural, technical, and governance components needed to implement FL on real-world health care data, including training deep learning neural networks. The study aims to apply this federated deep learning infrastructure to the use case of gross tumor volume segmentation on chest computed tomography images of patients with lung cancer and present the results from a proof-of-concept experiment.

Methods: The PHT framework addresses the challenges of data privacy when sharing data, by keeping data close to the source and instead bringing the analysis to the data. Technologically, PHT requires 3 interdependent components: “tracks” (protected communication channels), “trains” (containerized software apps), and “stations” (institutional data repositories), which are supported by the open source “Vantage6” software. The study applies this federated deep learning infrastructure to the use case of gross tumor volume segmentation on chest computed tomography images of patients with lung cancer, with the introduction of an additional component called the secure aggregation server, where the model averaging is done in a trusted and inaccessible environment.

Results: We demonstrated the feasibility of executing deep learning algorithms in a federated manner using PHT and presented the results from a proof-of-concept study. The infrastructure linked 12 hospitals across 8 nations, covering 4 continents, demonstrating the scalability and global reach of the proposed approach. During the execution and training of the deep learning algorithm, no data were shared outside the hospital.

Conclusions: The findings of the proof-of-concept study, as well as the implications and limitations of the infrastructure and the results, are discussed. The application of federated deep learning to unstructured medical imaging data, facilitated by the PHT framework and Vantage6 platform, represents a significant advancement in the field. The proposed infrastructure addresses the

challenges of data privacy and enables collaborative model development, paving the way for the widespread adoption of deep learning–based tools in the medical domain and beyond. The introduction of the secure aggregation server implied that data leakage problems in FL can be prevented by careful design decisions of the infrastructure.

Trial Registration: ClinicalTrials.gov NCT05775068; https://clinicaltrials.gov/study/NCT05775068

(JMIR AI 2025;4:e60847) doi: 10.2196/60847

KEYWORDS

gross tumor volume segmentation; federated learning infrastructure; privacy-preserving technology; cancer; deep learning; artificial intelligence; lung cancer; oncology; radiotherapy; imaging; data protection; data privacy

Introduction

Federated learning (FL) allows the collaborative development of artificial intelligence models using large datasets, without the need to share individual patient-level data [1-4]. In FL, partial models trained on separate datasets are shared, but not the data itself, hence a global model is derived from the collective set of partial models. This study introduces an innovative FL framework known as the Personal Health Train (PHT) that includes the procedural, technical, and governance components needed to implement FL on real-world health care data, including the training of deep learning neural networks [5]. The PHT infrastructure is supported by a free and open-source infrastructure known as "priVAcy preserviNg federaTed leArninG infrastructurE for Secure Insight eXchange," that is, Vantage6 [6]. We will describe in detail an architecture for training a deep learning model in a federated way with 12 institutional partners located in different parts of the world.

Sharing patient data between health care institutions is tightly regulated due to concerns about patient confidentiality and the potential for misuse of data. Data protection laws—including the European Union's General Data Protection Regulations; Health Insurance Portability and Accountability Act of 1996 (HIPAA) in the United States; and similar regulations in China, India, Brazil, and many other countries—place strict conditions on the sharing and secondary use of patient data [7]. Incompatibilities between laws and variations in the interpretation of such laws lead to strong reluctance about sharing data across organizational and jurisdictional boundaries [8-10].

To address the challenges of data privacy, a range of approaches have been published in the literature. Differential privacy, homomorphic encryption, and FL comprise a family of applications known as "privacy enhancing technologies" [11-13]. The common goal of privacy-enhancing technologies is to unlock positively impactful societal, economic, and clinical knowledge by analyzing data en masse, while obscuring the identity of study subjects that make up the dataset. Academic institutions are more frequently setting up controlled workspaces (eg, secure research environments [SREs]), where multiple researchers can collaborate on data analysis within a common cloud computing environment, but without allowing access to the data from outside the SRE desktop; however, this assumes that all the data needed have been transferred into the SRE in the first place [14,15]. Similarly, the National Institutes of Health has set up an "Imaging Data Commons" to provide secure access to a large collection of publicly available cancer imaging data colocated with analysis tools and resources [16]. Other researchers have shown that blockchain encryption technology can be used to securely store and share sensitive medical data [17]. Blockchain ensures data integrity by maintaining an audit trail of every transaction, while zero trust principles make sure the medical data are encrypted and only authenticated users and devices interact with the network [18].

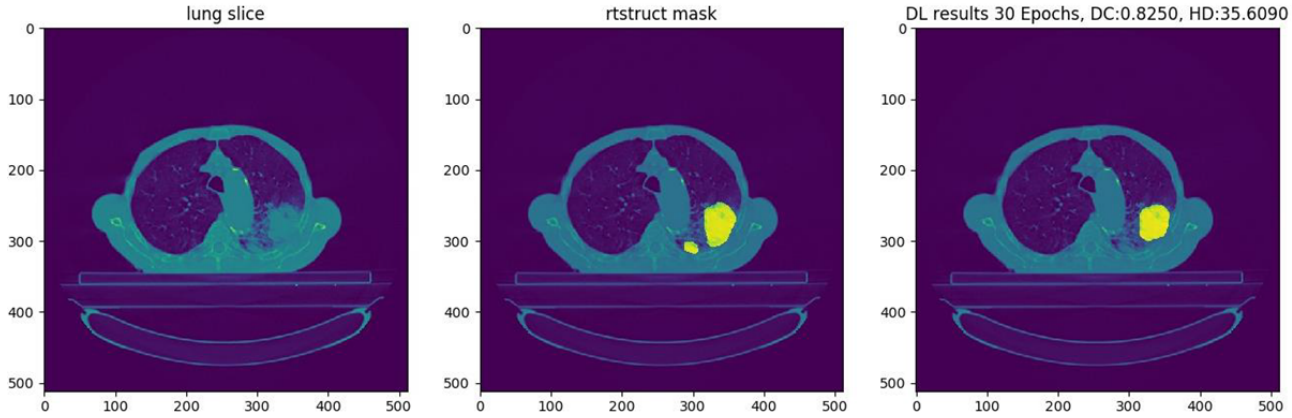
From a procedural point of view, the PHT manifesto for FL rules out the sharing of individual patient-level data between institutions, no matter if the patient data have been deidentified or encrypted [19]. The privacy-by-design principle here may be referred to as "safety in numbers," that is, any single individual's data values are obscured, by computing either the descriptive statistics or the partial model, over multiple patients. PHT allows sufficiently adaptable methods of model training, such as iterative numerical approximation (eg, bisection) or federated averaging (FedAvg [20]), and does not mandatorily require model gradients or model residuals, which are well-known avenues of privacy attacks [21-24]. Governance is essential with regards to compliance with privacy legislation and division of intellectual property between collaboration partners. A consortium agreement template for PHT has been made openly accessible [25], which is based on our current consortium ARGOS (artificial intelligence for gross tumor volume segmentation) [26]. Technologically, PHT requires 3 interdependent components to be installed—"tracks" are protected telecommunications channels that connect partner institutions, "trains" are Docker containerized software apps that execute a statistical analysis that all partners have agreed upon, and "stations" are the institutional data repositories that hold the patient data [23]. It is this technological infrastructure—the tracks, trains, and stations—that is supported by the aforementioned Vantage6 software, for which detailed stand-alone documentation exists [27].

The paper proposes a federated deep learning infrastructure based on the PHT manifesto [19], which provides a governance and ethical, legal, and social implications framework for conducting FL studies across geographically diverse data providers. The research aims to showcase a custom FL infrastructure using the open-source Vantage6 platform, detailing its technological foundations and implementation specifics. The paper emphasizes the significance of the implemented custom federation strategy, which maintains a strict separation between intermediate models from both internal and external user access. This approach is crucial for safeguarding the security and privacy of sensitive patient data,

as it prevents potential reverse engineering of intermediate results that could compromise confidentiality. This aggregation strategy is particularly important in the case of deep learning–based studies where multiple iterations of models or gradients are necessary to derive an optimal global model.

To demonstrate the infrastructure's robustness and practical applicability, the study presents a proof-of-concept involving the development of a federated deep learning algorithm based on 2D convolutional neural network (CNN) architecture [28]. This algorithm was implemented to automatically segment gross tumor volume (GTV) from lung computed tomography (CT) images of patients with lung cancer. Figure 1 [29] demonstrates a manual segmentation and deep learning–based segmentation of a tumor in the chest CT image of a patient. The subsequent sections provide a comprehensive account of the precise technical specifications of the infrastructure that links 12 hospitals across 8 nations, covering 5 continents. The algorithm developed learns from the distributed datasets and deploys it using the infrastructure. However, it is important to mention that the choice of the use case is only exemplary in nature, and the infrastructure is equipped to train any kind of deep learning architecture for relevant clinical use cases.

Figure 1. Illustrative result on a hold-out validation slice; the main bulk of the gross tumor volume as determined by the oncologist (middle) has been correctly delineated by the deep learning algorithm (right), but a small tumor mass adjacent and to the lower right of the main gross tumor volume mass has been missed (reproduced from Figure 6 of Chapter 4 of the thesis by Patil [29], which is published under the Taverne License [Article 25fa of the Dutch Copyright Act]).



The research used a deep learning architecture because in recent times the application of deep learning in health care has led to impressive results, specifically in the areas of natural language processing and computer vision (medical image analysis), with the promise for more efficient diagnostics and better predictions of treatment outcomes in future [30-35]. However, for robust generalizability, and to earn clinicians' acceptance, it is essential that artificial intelligence apps are trained on massive volumes of diverse and demographically representative health care data across multiple institutions. Given the barriers to data sharing, this is clearly an area where FL can play a vital role. Many studies have been published that present FL on medical data including federated deep learning [36-40]. However, only a limited number of studies have documented the use of dedicated frameworks and infrastructures in a transparent manner. The adoption of a custom federation strategy or absence of explicit reporting on the used infrastructure is observed in most of the studies. Table 1 summarizes the small number of FL studies that have been published in connection with deep learning investigations related to medical image segmentations to date.

The paper primarily focuses on demonstrating the training and aggregation mechanism of a deep learning architecture within a FL framework. It deliberately avoids delving into the optimization of model performance or clinical accuracy, as these aspects fall outside the paper's scope. Instead of emphasizing the selection of an optimal CNN architecture or aggregation strategy [39], the research concentrates on elucidating the functionality of the FL infrastructure. Existing literature has shown that FL models can achieve performance comparable to centrally trained models [38,41,45-47]. This supports the assumption that, given identical datasets and CNN architectures, a model trained using FL would likely yield similar results to one trained through centralized methods. The paper operates under this premise, prioritizing the explanation of the FL process over demonstrating performance parity with centralized training approaches.

The study highlights 3 key points as follows:

- FL is particularly well suited for deep learning applications, which typically require vast amounts of data. This makes it an ideal showcase for the federated approach.
- When implementing federated deep learning, it is crucial to have a robust infrastructure and use a customized, secure aggregation strategy. These elements are essential for safeguarding the privacy of sensitive patient information.
- FL in real-world medical data is not just a technological challenge; it requires a comprehensive strategy that addresses ethical, legal, governance, and organizational aspects, as highlighted by the PHT manifesto.

Table 1. Existing studies from the literature focusing on federated deep learning on medical images.

Infrastructure and clinical use case	Data type	Scale
NVIDIA FLARE/CLARA		
Prostate segmentation of T2-weighted MRI[a] [41]	DICOM MRI	3 centers
COVID-19 pneumonia detection [42]	Chest CT[b]	7 centers
Tensorflow federated		
COVID-19 prediction from chest CT images [43]	Chest CT	3 datasets
OpenFL		
Glioblastoma tumor boundary detection [44]	Brain MRI	71 centers

[a]MRI: magnetic resonance imaging.

[b]CT: computed tomography.

The findings of the proof-of-concept study, as well as the implications and limitations of the infrastructure and the results, are discussed. The subsequent section of the paper is structured as follows: the *Methods* section describes the approach taken, followed by the *Results*, which detail the implementation of the infrastructure and a proof-of-concept execution. Finally, the paper concludes with a *Discussion* section.

Methods

Overview

When conducting a federated deep learning study, it is crucial to consider several key perspectives, which include both technical as well as organizational and legal aspects. These key factors have been instrumental in designing the infrastructure architecture used for training the deep learning algorithm. In this section, we discuss the technical details while adhering to an Ethics-Legal-Social Impact framework as laid down by the PHT manifesto. The technical design decisions are based on the following assumptions:

Data Landscape

Understanding the data landscape is crucial in designing and deploying FL algorithms. The technological approaches for handling horizontally partitioned data, where each institution contains nonoverlapping human subjects but the domain of the data (eg, CT images of lung cancer) is the same across different institutions, can differ significantly from those used for vertically partitioned data, where each institution contains the same human subjects but the domain of the data do not overlap (eg, CT scans in one, but socioeconomic metrics in another). Additionally, unstructured data, such as medical images, requires different algorithms and preprocessing techniques compared with structured data. In this paper, the architecture will only focus on CT scans and horizontally partitioned patient data.

Data Preprocessing

In a horizontally partitioned FL setting, the key preprocessing steps can be standardized and sent to all partner institutions. However, the workflow needs to handle differences in patients, scan settings, and orientations. Anonymization, quality improvements, and DICOM standardization ensure homogeneity and high quality across hospitals. These offline preprocessing steps, applied consistently to the horizontally partitioned data, enabled using the same model across institutions, crucial for the FL study's success.

Network Topology of the FL Infrastructure

The network topology choice for implementing FL can vary from client-server, peer-to-peer, tree-based hierarchical, or hybrid topologies. While peer-to-peer architecture is more cost-effective and offers a high capacity, it has the disadvantages of a lack of security and privacy constraints and a complex troubleshooting process in the event of a failure. The choice of network topology for this study is based on a client-server architecture, offering a single point of control in the form of the central server.

Choice of Model Aggregation Site

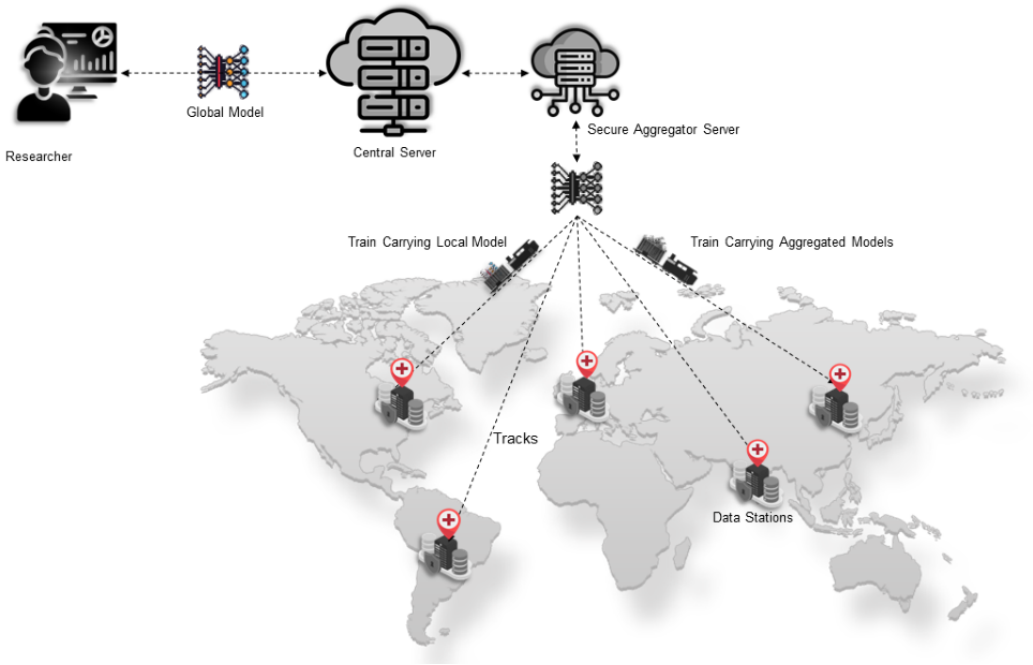
For a client-server architecture, the model aggregation can occur either in one of the data providers' machines, the central server, or in a dedicated aggregation server. For this implementation, we opted to use a dedicated aggregation server. The details and benefits of the implementation are discussed in the next section.

Training Strategy

The communication mechanism for transferring weights can be either synchronous, asynchronous, or semisynchronous, and weights can be consolidated using ensemble learning, FedAvg, split learning, weight transfer, or swarm learning. The strategy used for this study is based on a synchronous mechanism using the FedAvg algorithm. This gives a simple approach, where the averaging algorithm waits for all the data centers to transfer the locally trained model before initiating the averaging.

Based on the assumption, Figure 2 depicts the overall architecture of the federated deep learning study presented in the paper. The next section describes the FL Infrastructure in detail.

Figure 2. Overall architecture of ARGOS (artificial intelligence for gross tumor volume segmentation) federated deep learning architecture adapted from Vantage6. The figure depicts a researcher connected to the central server, a secure aggregation server, trains carrying models, connected data stations, and the communicating tracks.



The ARGOS Federated Deep Learning Infrastructure

Overview

In accordance with the PHT principles, the ARGOS infrastructure is comprised of 3 primary categories of components, labeled as the data stations, the trains, and the track. Furthermore, the architectural framework encompasses various roles that map to the level of permissions and access, specifically a track provider, the data providers, and the researcher. The infrastructure implementation can be further categorized into 3 important components: a central coordination server, a secure aggregation server (SAS), and the nodes located at each "data station." In the following sections, we attempt to describe each of these components and the respective stakeholders responsible for maintaining them.

Central Coordinating Server

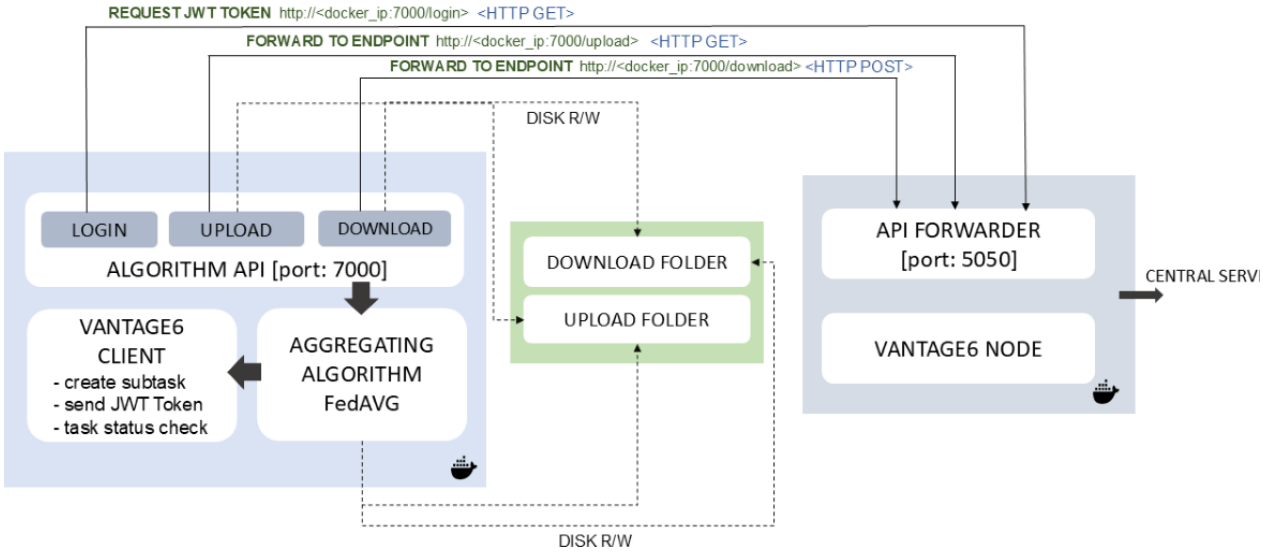
The central coordination server is located at the highest hierarchical level and serves as an intermediary for message exchange among all other components. The components of the system, including the users, data stations, and SAS, are registered entities that possess well-defined authentication mechanisms within the central server. It is noteworthy that the central acts as a coordinator rather than a computational engine. Its primary function is to store task-specific metadata relevant to the task initiated for training the deep learning algorithm. In the original Vantage6 infrastructure, the central server also stores the intermediate results. In the ARGOS infrastructure, the central server is designed to not store any intermediate results but only the global aggregated model at the end of the entire training process.

Secure Aggregation Server

The SAS refers to a specialized station that contains no data and functions as a consolidator of locally trained models. The aggregator node is specifically designed to possess a Representational State Transfer (REST)–application programming interface (API) termed as the API Forwarder. The API Forwarder is responsible for managing the requests received from the data stations and subsequently routing them to the corresponding active Docker container, running the aggregation algorithm.

To prevent any malicious or unauthorized communication with the aggregator node, each data station is equipped with a JSON Web Token (JWT) that is unique for each iteration. The API Forwarder only accepts communications that are accompanied by a valid JWT. The implementation of this functionality guarantees the protection of infrastructure users and effectively mitigates the risk of unauthorized access to SAS. Figure 3 shows the architecture and execution mechanism for the SAS.

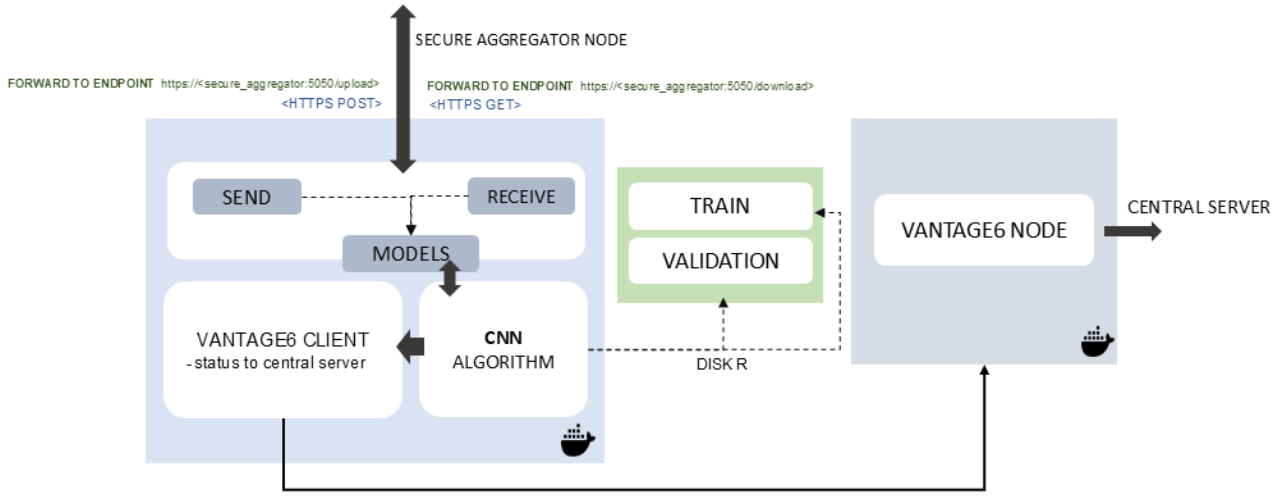
Figure 3. Architecture of the secure aggregation server, showing incoming and outgoing requests from the data station nodes. The upload and download folders are temporary locations used within the running Docker container to store the local and averaged models through disk read or write operations. The API forwarder, running at port 5050 and embedded within the Vantage6 infrastructure, forwards the incoming requests from the data station nodes to the algorithm API running at local port 7000 within the Docker container through HTTP requests. The SAS is hosted behind the firewall of a proxy server, which allows only hypertext transfer protocol secure (HTTPS) communication from the participating nodes. API: application programming interface; FedAvg: federated averaging; JWT: JSON Web Token.



Data Stations

Data stations are devices located within the confines of each hospital's jurisdiction that are not reachable or accessible from external sources other than Vantage6. The data stations communicate with the central server through a pull mechanism. Furthermore, the data stations not only serve as hosts for the infrastructure node but also offer the essential computational resources required for training the deep learning network. The infrastructure node is the software component installed in the data stations that orchestrates the local execution of the model and its communication with the central server and the SAS. Each data station is equipped with at least 1 graphics processing unit (GPU), which enables the execution of CNNs. Preprocessing of the raw CT images was executed locally, using automated preprocessing scripts packaged as Docker containers, and the preprocessed CT images are stored within a file system volume in each station. The CNN Docker is designed and allowed to access the preprocessed images during training. The primary function of the data station is to receive instructions from both the SAS and the central server, perform the computations needed for training the CNN algorithm, and subsequently transmit the model weights back to the respective sources. Figure 4 depicts the architectural layout of the data station and node component of the infrastructure.

Figure 4. Architecture of the data station node component. The node runs the CNN algorithm to learn from the local data. The node further sends and receives model weights from the secure aggregation server. The train and validation folders are persistent locations within the data stations, storing the preprocessed NIFTI images. At the end of each training cycle, the intermediate averaged model is first evaluated on the validation sample. CNN: convolutional neural network; HTTPS: hypertext transfer protocol secure; NIFTI: neuroimaging informatics technology initiative.



Train

The "train" in the form of a Docker image encompasses several components bundled together: an untrained U-Net [48,49], a type of CNN architecture designed for image segmentation tasks for training on local data; the aggregation algorithm used for consolidating the models; and a secondary Python Flask API known as the Algorithm API for facilitating the communication of these models. The Algorithm API is designed to cater to requests from the API Forwarder and is built within the algorithm container. Two levels of API ensured that the node could handle multiple requests and divert to appropriate Docker containers. Furthermore, the first level of API also helps in restricting malicious requests by checking the JWT token signature, so that the models within the master Docker container are protected. Each data station is responsible for training and transmitting the CNN model to the aggregator server. This suggests that the aggregation algorithm exhibits a waiting period during which it ensures that all data stations have effectively transmitted their models to the server before proceeding to the next iterations. The process is executed in an iterative manner until convergence is achieved or the specified number of iterations is attained.

Tracks and Track Provider

The various infrastructure components establish coordination among themselves through the use of secure communication channels commonly referred to as the "tracks." The communication channels are enabled with end-to-end encryption. The responsibility for the maintenance of the infrastructure, including the hosting of the central coordinating server and the specialized SAS, lies with the track provider. The track provider is additionally accountable for the maintenance of the "tracks" and aids the data providers in establishing the local segment of the infrastructure known as the "nodes."

Data Provider

Data providers refer to hospitals and health care organizations that are responsible for curating the pertinent datasets used for training the deep learning network. The responsibility of hosting the data stations within their respective local jurisdiction lies with the data provider. They exercise authority over the data as well as the infrastructure component called the node.

Researcher

The researcher is responsible for activating the deep learning algorithm and engaging in the authentication process with the central coordinating server using a registered username and password. This allows the researcher to establish their identity and gain secure access to the system, with their communication safeguarded through end-to-end encryption. The researcher can then assign tasks to individual nodes, monitor progress, and terminate tasks in the event of failure. Importantly, the researcher's methodology is designed to keep the intermediate outcomes of the iterative deep learning training process inaccessible, ensuring that the ultimate global model can only be obtained upon completion of all training iterations, thereby mitigating the risk of unauthorized access by malicious researchers to the intermediate models and providing a security mechanism against insider attacks.

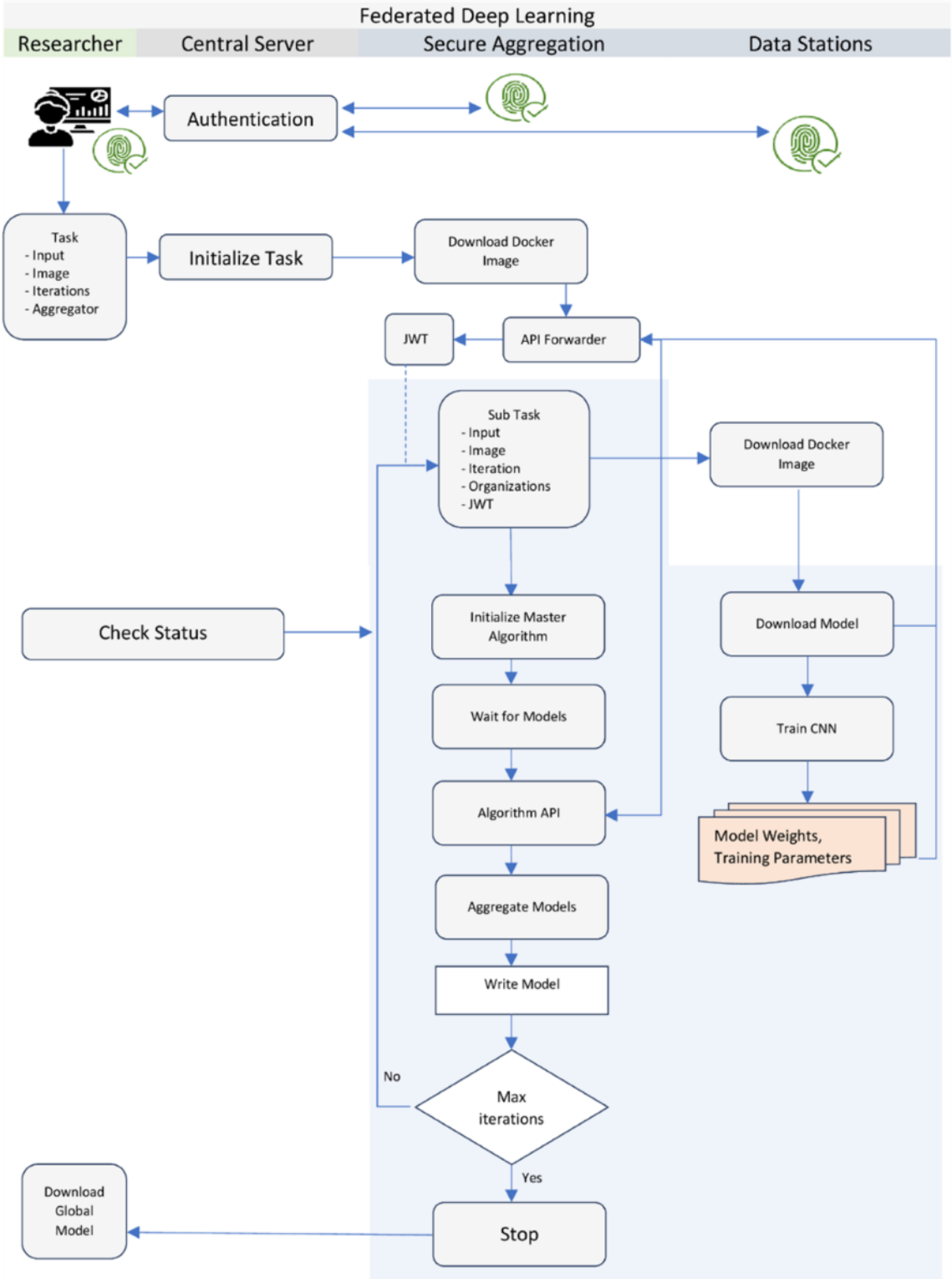
Training Process

Each of the components described above works in a coordinated manner to accomplish the convergence of the deep learning algorithm. The training process begins with the researcher authenticating with the central server. Upon successful authentication, the researcher specifies the task details, including a prebuilt Docker image, input parameters, number of iterations, and the identity of the SAS. The task is then submitted to the central server, which forwards it to the connected nodes. The SAS is the first to receive the task request. It downloads the specified Docker image from the registry and initiates the master algorithm. The master algorithm orchestrates the training at each data station node through the central server. The central server then forwards a subtask request to all the data stations. Like the SAS, the data nodes download the same Docker image and initiate the node part of the algorithm. The node algorithm runs the learning process on local data for the specified number of epochs. After each training cycle, the node algorithm sends the local model weights to the SAS.

The SAS verifies the JWT signature of each received model and forwards the request to the Algorithm API. The Algorithm API extracts the weight and metadata information of the models. Once the SAS receives all the required locally trained models for that cycle, it initiates the FedAvg algorithm to consolidate the models and create an intermediate averaged model, which is stored locally. This completes the first iteration of the training cycle. For the second and subsequent iterations, the data stations request the SAS to send the intermediate averaged model weights from the previous iteration. The SAS validates these requests and sends the model weights to the data stations, which then use them for further training on their local data. This cycle of training and averaging continues until the model converges or the desired number of iterations is reached.

At the end of the training process, the SAS sends a notification to the researcher indicating the successful completion of the task. The researcher can then download the final global model from the server. It is important to note that during the training iterations, the researcher or other users of the infrastructure do not have access to the intermediate averaged models generated by the SAS. This design choice prevents the possibility of insider attacks and data leakage, as users cannot regenerate patterns from the training data using the intermediate models. Figure 5 shows the diagrammatic representation of the training process spread across the infrastructure components.

Figure 5. Process illustration of federated deep learning training. All entities, including the researcher, the central aggregation server, and the data stations, first authenticate with the central server. The researcher creates a task description and submits the task to the central server, which then forwards the request to the secure aggregation node to start the master task. The master task then sends a request to all data stations to download the algorithm Docker image and start training on the local data. Researchers can monitor the algorithm's execution status on the central server using the "check status" function, which reports whether each iteration is completed or aborted as processed by the secure aggregation server and data stations. At the end of each local training, the data stations send the models to the API forwarder of the secure aggregation node by authenticating against a valid JWT token. The JWT token ensures that no unauthorized data station is able to send or receive models from the secure aggregation server. API: application programming interface; CNN: convolutional neural network; JWT: JSON Web Token.



Code Availability

The federated deep learning infrastructure and the algorithm used in this research are open source and publicly available. The codebase, encompassing the components of the infrastructure, the algorithm, and wrappers for running it in the infrastructure and the researcher notebooks, are all available and deposited on GitHub, a public repository platform, under the Apache 2.0 license. This open access allows the research community to scrutinize and leverage our implementation for further development in the field of FL.

The Vantage6 (version 2.0.0) [27,50] open-source software was customized to cater to the specific requirements for running the deep learning algorithm. The central server (Vantage6 version 2.0.0) and the aggregator server were hosted by Medical Data Works BV in 2 separate cloud machines (Microsoft Azure). At each participating center, the "node" component of the software was installed and setup either on a physical or cloud machine running Ubuntu (version 16.0) or above with an installation of Python, (version 3.7 or above; Python Software Foundation), Docker Desktop (personal edition), and NVIDIA CUDA GPU interface (version 11.0). The source code of the customized "node" [51] and setup instructions [52] are available on respective GitHub repositories. The federated deep learning algorithm was adapted to the infrastructure as Python scripts [53] and wrapped in a Docker container. Separately, the "researcher" notebooks [54] containing python scripts for connecting to the infrastructure and running the algorithms are also available on GitHub. Table 2 provides an outline of the resource requirement and computational cost of the experiment.

Table 2. Resource requirement and computational cost.

End points	Resource requirement		Average execution time (per iteration)
	Software	Hardware	
Central server	• Ubuntu (version 16) and above • Docker Desktop • Python (3.7 or above) • Vantage6 (version 2.0.0)	• 4 CPUsa • 16 GB RAM • 20 GB Disk Space	N/A[b]
Data station	• Ubuntu (version 16) and above • Docker Desktop • Python (3.7 or above) • Vantage6 (version 2.0.0) • CUDA GPU Interface (version 11.0)	• 4 CPUs • 1 GPUc • 16 GB RAM • 40 GB disk space	40 mins
Secure aggregation server	• Ubuntu (version 16) and above • Docker Desktop • Python (3.7 or above) • Vantage6 (version 2.0.0)	• 4 CPUs • 16 GB RAM • 40 GB disk space	60 seconds

[a]CPU: central processing unit.

[b]Not applicable.

[c]GPU: graphics processing unit.

Ethical Considerations

The work was performed independently with the ethics board's approval from each participating institution. Approvals from each of the participating institutions including soft copies of approval have been submitted to the leading partner. The lead partner's institutional review board approval (MAASTRO Clinic, The Netherlands) is "W 20 11 00069" (approved on November 24, 2020). The authors attest that the work was conducted by the ethical standards of the responsible committee on human experimentation (institutional and national) and with the Helsinki Declaration of 1975.

Results

Overview

The study was carried out and concluded in 4 primary stages using an agile approach as follows: planning, design and development, partner recruitment, and execution of federated deep learning. The planning phase of the study, which encompassed a meticulous evaluation and determination of the following inquiries, held equal significance to the description of the clinical issue and data requirements.

- What are the minimum resource requirements for each participating center?
- How to design a safe and robust infrastructure to effectively address the requirements of a federated deep learning study?
- How can a reliable and data-agnostic federated deep learning algorithm be designed?

- What are the operational and logistical challenges associated with conducting a large-scale federated deep learning study?

The second phase, that is, the design and development phase, primarily focused on the creation, testing, and customization of the Vantage6 infrastructure for studies specifically focused on deep learning. To meet the security demands of these investigations, this study involved the development of the SAS, which was not originally included in the Vantage6 architecture. The CNN algorithm was packaged as a Docker container and made compatible with the Vantage6 infrastructure, allowing it to be easily deployed and used within the Vantage6 ecosystem. Prior to the deployment of the algorithm, it underwent testing using multiple test configurations consisting of data stations that were populated with public datasets.

The primary objective of the third phase entailed the recruitment of partners who displayed both interest and suitability from various global locations. The project consortium members became part of the project by obtaining the necessary institutional review board approvals and signing an infrastructure user agreement. This agreement enabled them to install the required infrastructure locally and carry out algorithmic execution. The inclusion criteria for patient data, as well as the technology used for data anonymization and preprocessing, were provided to each center. The team collaborated with each partner center to successfully implement the local component of the infrastructure.

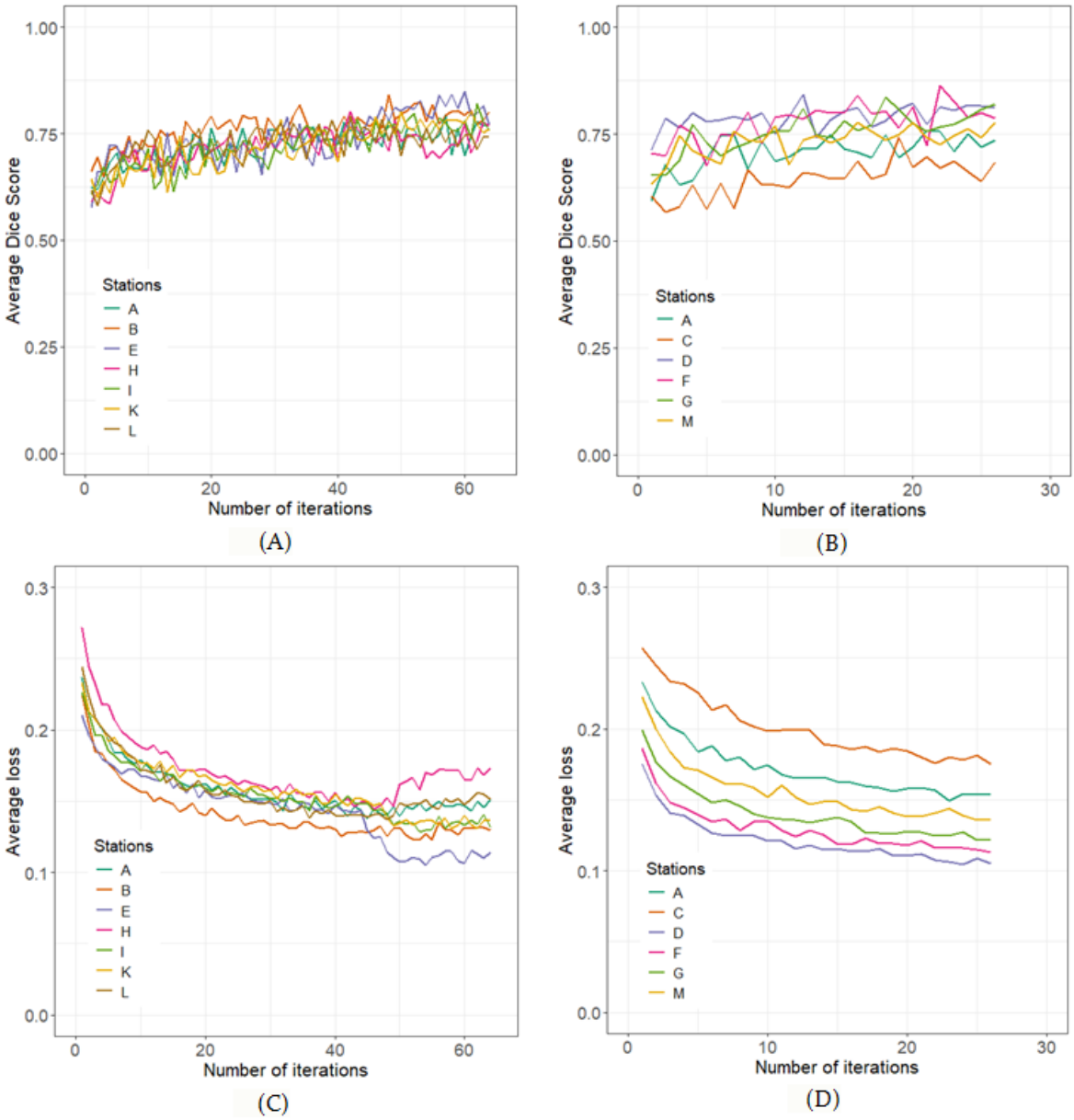
The concluding stage of the study involved the simultaneous establishment of connections between all partner centers and the existing infrastructure. The algorithm was subsequently initiated by the researcher and the completion of the predetermined set of federated iterations was awaited across all centers.

Proof of Concept

The architectural strategy described above was implemented among ARGOS consortium partners on real-world lung cancer CT scans. For an initial “run-up” of the system, we deployed the abovementioned PHT system across 12 institutions, located in 8 countries and 4 continents. A list of members participating in the ARGOS consortium can be found on the study protocol [26]. In total, 2078 patients’ data were accessible via the infrastructure for training (n=1606) and holdout validation (n=472). For this initial training experiment, the 12 centers were divided into 2 groups. The first, referred to as group A, comprised 7 collaborators, and we were able to reach a total of 64 iterations of model training each with 10,000 steps per iteration. Likewise, group B comprising 6 hospitals was able to train the deep learning model for 26 iterations. It was observed that no significant improvement of the model was observed for both groups after 26th iteration. The results from the proof-of-concept study are shown in Figure 6.

While the training time for the models was similar at each center, how quickly they could be uploaded and downloaded depended heavily on the quality of the internet connection. This meant the entire process was significantly slowed down by the center with the slowest internet.

Figure 6. Plots showing the results from training the convolutional neural network on two groups as follows: group 1 (A, B, E, H, I, K, L) and group 2 (A, C, D, F, G, M). (A) Average Dice score per iteration of the model trained on group 1. (B) Average Dice score per iteration of the model trained on group 2. (C) Average training loss per iteration of the model trained on group 1. (D) Average training loss per iteration of the model trained on group 2.



Discussion

This study demonstrated the feasibility of a privacy-preserving federated deep learning infrastructure and presented a proof-of-concept study for GTV segmentation in patients with lung cancer. Using the PHT framework, the infrastructure linked 12 hospitals across 8 nations, showcasing its scalability and global applicability. Notably, throughout the process, no patient data were shared outside the participating institutions, addressing significant data privacy concerns. The introduction of a SAS further ensured that model averaging occurred in a secure environment, mitigating potential data leakage issues in FL.

One of the most used methodologies in recent years has been the use of FL for promoting research on privacy-sensitive data. To orchestrate FL on nonstructured data in the horizontal partitioning context, it is essential to develop specialized software for edge computation and technical infrastructures for cloud aggregation. These infrastructures enable federated machine learning (FML) responsibilities to be carried out in a secure and regulated manner. However, only a limited number of these studies have documented the background governance strategies and the ethical, legal, and social implications framework for conducting such studies.

The study presented a novel approach for executing large-scale federated deep learning on medical imaging data, integrating geographically dispersed real-world patient data from cross-continental hospital sites. The deep learning algorithm was designed to automatically delineate the GTV from chest CT images of patients with lung cancer who underwent radiotherapy treatment. The underlying FL infrastructure architecture was designed to securely perform deep learning training and was tested for vulnerabilities from known security

threats. This paper predominantly discussed the FL infrastructure architecture and presented a firsthand experience of conducting such studies. The preliminary training of the deep learning algorithm serves as the feasibility demonstration of the methodology, and further refinement is required to achieve acceptable clinical-grade accuracy and generalizability.

The study used an open-source and freely accessible technological stack to demonstrate the feasibility and applicability of federated deep learning. Vantage6, a Python-based FL infrastructure, is used to train and coordinate deep learning execution. TensorFlow and Flask, both open-source Python libraries, are used for the development of the algorithm, subsequently encapsulated within Docker services for containerization purposes. The communication channels between the hospital, central server, and the aggregation node have been secured using Hypertext Transfer Protocol Secure and Secure Hash Algorithm encryption. The hospital sites' computer systems were based on the Ubuntu operating system and equipped with at least 1 GPU to enhance computational capabilities. The participating centers had the flexibility to choose any CUDA-compatible GPU devices and determine the number of GPUs to use, enabling resource-constrained centers to contribute. However, a limitation exists in terms of computational time due to the synchronous training process being dependent on the slowest participant.

The infrastructure has been tested against known security attacks and as defined by the Open Worldwide Application Security Project top-ten categories [55]. It has been found that the Vantage6 app is impeccable against insecure design, software and data integrity failures, security logging and monitoring failures, and server-side request forgery and sufficiently secured against broken access control, cryptographic failures, injection, security misconfigurations, vulnerable and outdated components, and finally identification and authentication failures. Since the infrastructure is dependent on other underlying technologies like Docker and Flask-API, the security measures in these technologies also affect the overall security of the infrastructure. Additionally, the infrastructure is hosted behind proxy firewalls, adding to its overall security against external threats.

In this study, we implemented a SAS positioned between the data nodes (eg, hospitals and clinics) and the central server. The SAS plays a crucial role in strengthening the privacy and confidentiality of the learning process. The SAS acts as an intermediary that temporarily stores the local model updates from the participating data nodes, ensuring complete isolation from the central server, researchers, and any external intruders. The key benefits of using a dedicated SAS over a random aggregation mechanism in FL are as follows:

- Privacy protection of individual user data and model updates:
  - The secure aggregation protocol ensures that the central server only learns the aggregated sum of all user updates, without being able to access or infer the individual user's private data or model updates.
  - By isolating the intermediate updates, the secure aggregation process prevents external attackers from performing model inversion attacks.
- Tolerance to user dropouts:
  - The SAS is designed to handle situations where some users fail to complete the execution. In the case of synchronous training, the server stores the latest successful model, enabling data nodes to pick up where they left off instead of restarting from scratch.
- Integrity of the aggregation process:
  - The secure aggregation protocol provides mechanisms to verify the integrity of the intermediate models by allowing only the known data nodes to send a model. This maintains the reliability and trustworthiness of the FL system.

FL offers 2 main approaches for model aggregation: sending gradients or weights [56,57]. In gradient sharing, data nodes update local models and transmit the gradients of their parameters for aggregation. Conversely, weight sharing involves sending the fully updated model weights directly to the server for aggregation. Sharing gradients have a higher risk of model inversion attacks. In the study presented here, the data nodes sent model weights instead of model gradients, thus preventing the "gradient leakage" problem. However, weight sharing is not failproof either [58], and the SAS plays a crucial role again in preventing users—internal or external—from accessing the weights from the aggregator machine.

The deployment of the FL infrastructure and training of the deep learning algorithm presented unique challenges that needed to be catered to. Some of them are listed below:

- Heterogeneity across hospitals: Initially, it was not possible to confirm the technology environment at each site. This required significant work to overcome the obstacles connected with each center while deploying a functional infrastructure, good communication, and efficient algorithms.
- Inconsistent IT policies: Standardizing the setup across institutions was hindered by varying IT governance and network regulations in different health care systems across different countries.
- Clinical expertise gap: The predominance of medical personnel over IT specialists at participating hospitals necessitated extensive documentation to ensure clinician comprehension of the FL process.
- Network bottlenecks: Network configurations at participating sites significantly impacted training duration, often leading to delays in model convergence.

The study presented in the paper has identified several areas that require further investigation and improvement. While the findings are valuable, the infrastructure, algorithm, and processes still need to be made more secure, private, trustworthy, robust, and seamless [59]. For example, incorporating homomorphic encryption of the learned models will enhance privacy and provide model obfuscation against inversion attacks. Finally, to further enhance confidence and trust in federated artificial intelligence, it is crucial to conduct additional studies involving a larger number of participating centers and a thorough clinical evaluation of the models.

Acknowledgments

We would like to express our sincere appreciation and gratitude to Integraal Kankercentrum Nederland (IKNL), the Netherlands, for their invaluable contribution in providing us with the necessary infrastructure support. We express our gratitude to Medical Data Works, the Netherlands, for their role as the infrastructure service provider in hosting the central and secure aggregation server. We also express our gratitude to Varsha Gouthamchand and Sander Puts for their contribution to the successful execution of the experiments. In conclusion, we express our gratitude to the various data-providing organizations for their substantial support and collaboration throughout all stages of the project. AC, LV, RF, and LW acknowledge financial support from the Dutch Research Council (NWO) (TRAIN project, dossier 629.002.212) and the Hanarth Foundation.

Conflicts of Interest

Dr AD and JvS are both cofounders, shareholders, and directors of Medical Data Works B.V.

References

1. Sun C, Ippel L, Dekker A, Dumontier M, van Soest J. A systematic review on privacy-preserving distributed data mining. Data Sci. Oct 2021;4(2):121-150. [doi: 10.3233/DS-210036]
2. Choudhury A, Sun, C, Dekker M, Dumontie J, van Soest. Privacy-preserving federated data analysis: data sharing, protection, bioethics in healthcare. In: El Naqa I, Murphy MJ, editors. Machine Deep Learning in Oncology. Cham, Switzerland. Springer International Publishing; 2022:135-172.
3. Deist TM, Dankers FJ, Ojha P, Scott Marshall M, Janssen T, Faivre-Finn C, et al. Distributed learning on 20 000+ lung cancer patients - the personal health train. Radiother Oncol. 2020;144:189-200. [FREE Full text] [doi: 10.1016/j.radonc.2019.11.019] [Medline: 31911366]
4. Choudhury A, Theophanous S, Lønne PI, Samuel R, Guren MG, Berbee M, et al. Predicting outcomes in anal cancer patients using multi-centre data and distributed learning - a proof-of-concept study. Radiother Oncol. 2021;159:183-189. [FREE Full text] [doi: 10.1016/j.radonc.2021.03.013] [Medline: 33753156]
5. Beyan O, Choudhury A, van Soest J, Kohlbacher O, Zimmermann L, Stenzhorn H, et al. Distributed analytics on sensitive medical data: the personal healt train. Data Intell. 2020;2(1-2):96-107. [FREE Full text] [doi: 10.1162/dint_a_00032]
6. Moncada-Torres A, Martin F, Sieswerda M, van Soest J, Geleijnse G. VANTAGE6: an open source privacy preserving federated learning infrastructure for secure insight exchange. AMIA Annu Symp Proc. 2020;2020:870-877. [FREE Full text] [Medline: 33936462]
7. Becker R, Chokoshvili D, Comandé G, Dove ES, Hall A, Mitchell C, et al. Secondary use of personal health data: when is it “Further Processing” under the GDPR, and what are the implications for data controllers? Eur J Health Law. 2022;30(2):129-157. [doi: 10.1163/15718093-bja10094]
8. El Naqa I, Ruan D, Valdes G, Dekker A, McNutt T, Ge Y, et al. Machine learning and modeling: data, validation, communication challenges. Med Phys. 2018;45(10):e834-e840. [FREE Full text] [doi: 10.1002/mp.12811] [Medline: 30144098]
9. van Stiphout R. How to share data and promote a rapid learning health medicine? In: Valentini HJ, Schmoll C, van de Velde JH, editors. Multidisciplinary Management of Rectal Cancer. Cham, Switzerland. Springer International Publishing; 2018:623-634.
10. Kazmierska J, Hope A, Spezi E, Beddar S, Nailon WH, Osong B, et al. From multisource data to clinical decision aids in radiation oncology: the need for a clinical data science community. Radiother Oncol. 2020;153:43-54. [FREE Full text] [doi: 10.1016/j.radonc.2020.09.054] [Medline: 33065188]
11. Fischer-Hübner S. Privacy-enhancing technologies. In: Liu T, Özsu MT, editors. Encyclopedia of Database Systems. Boston, MA. Springer; 2009:2142-2147.
12. Coopamootoo KPL. Usage patterns of privacy-enhancing technologies. In: ACM Digital Library. 2020. Presented at: CCS '20: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security; November 2, 2020; New York, NY. URL: https://dl.acm.org/doi/10.1145/3372297.3423347
13. Emerging privacy-enhancing technologies. OECD. URL: https://www.oecd.org/publications/emerging-privacy-enhancing-technologies-bf121be4-en.htm [accessed 2025-04-25]
14. Kavianpour S, Sutherland J, Mansouri-Benssassi E, Coull N, Jefferson E. Next-generation capabilities in trusted research environments: interview study. J Med Internet Res. 2022;24(9):e33720. [FREE Full text] [doi: 10.2196/33720] [Medline: 36125859]
15. Design a secure research environment for regulated data. Microsoft. URL: https://learn.microsoft.com/en-us/azure/architecture/ai-ml/architecture/secure-compute-for-research [accessed 2024-04-25]
16. Imaging data commons. National Cancer Institute Cancer Research Data Commons. URL: https://datacommons.cancer.gov/repository/imaging-data-commons [accessed 2024-04-25]
17. Kotter E, Marti-Bonmati L, Brady AP, Desouza NM. ESR white paper: blockchain and medical imaging. Insights Imaging. 2021;12(1):82. [FREE Full text] [doi: 10.1186/s13244-021-01029-y] [Medline: 34156562]

18. Sultana M, Hossain A, Laila F, Taher KA, Islam MN. Towards developing a secure medical image sharing system based on zero trust principles and blockchain technology. BMC Med Inform Decis Mak. 2020;20(1):256. [FREE Full text] [doi: 10.1186/s12911-020-01275-y] [Medline: 33028318]
19. Manifesto of the personal health train consortium. Data Driven Life Sciences. URL: https://www.dtls.nl/wp-content/uploads/2017/12/PHT_Manifesto.pdf [accessed 2024-03-11]
20. McMahan E, Moore D, Ramage S, Hampson BA. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of Machine Learning Research. 2017. Presented at: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics; April 20-22, 2017; Fort Lauderdale, FL. URL: https://proceedings.mlr.press/v54/mcmahan17a.html
21. Zhang C, Choudhury A, Shi Z, Zhu C, Bermejo I, Dekker A, et al. Feasibility of privacy-preserving federated deep learning on medical images. Int J Radiat Oncol Biol Phys. 2020;108(3):e778. [doi: 10.1016/j.ijrobp.2020.07.234]
22. Choudhury A, van Soest J, Nayak S, Dekker A. Personal health train on FHIR: a privacy preserving federated approach for analyzing FAIR data in healthcare. In: Bhattacharjee A, Kr. Borgohain S, Soni B, Verma G, Gao XZ, editors. Machine Learning, Image Processing, Network Security and Data Sciences. Singapore. Springer; 2020.
23. Gouthamchand V, Choudhury A, P Hoebers FJ, R Wesseling FW, Welch M, Kim S, et al. Making head and neck cancer clinical data findable-accessible-interoperable-reusable to support multi-institutional collaboration and federated learning,? BJR Artif Intell. 2024;1(1).
24. Sun C, van Soest J, Koster A, Eussen SJ, Schram MT, Stehouwer CD, et al. Studying the association of diabetes and healthcare cost on distributed data from the maastricht study and statistics Netherlands using a privacy-preserving federated learning infrastructure. J Biomed Inform. 2022;134:104194. [FREE Full text] [doi: 10.1016/j.jbi.2022.104194] [Medline: 36064113]
25. Railway governance. Medical Data Works. URL: https://www.medicaldataworks.nl/governance [accessed 2024-09-11]
26. Dekker A. ARtificial Intelligence for Gross Tumour vOlume Segmentation (ARGOS). National Library of Medicine. URL: https://clinicaltrials.gov/study/NCT05775068 [accessed 2024-01-11]
27. Overview: what is vantage6? Vantage6 documentation. URL: https://docs.vantage6.ai/en/main/ [accessed 2024-04-11]
28. U-Net: convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Cham, Switzerland. Springer; Nov 18, 2015.
29. Patil RB. Prognostic and prediction modelling with radiomics for non-small cell lung cancer. Maastricht University. 2020. URL: https://cris.maastrichtuniversity.nl/en/publications/prognostic-and-prediction-modelling-with-radiomics-for-non-small- [accessed 2020-10-06]
30. Tao Z, Lyu S. A survey on automatic delineation of radiotherapy target volume based on machine learning. Data Intell. 2023;5(3):814-856. [doi: 10.1162/dint_a_00204]
31. Liu X, Li KW, Yang R, Geng LS. Review of deep learning based automatic segmentation for lung cancer radiotherapy. Front Oncol. 2021;11:717039. [FREE Full text] [doi: 10.3389/fonc.2021.717039] [Medline: 34336704]
32. Ma Y, Mao J, Liu X, Dai Z, Zhang H, Zhang X, et al. Deep learning-based internal gross target volume definition in 4D CT images of lung cancer patients. Med Phys. 2023;50(4):2303-2316. [doi: 10.1002/mp.16106] [Medline: 36398404]
33. Zhang F, Wang Q, Li H. Automatic segmentation of the gross target volume in non-small cell lung cancer using a modified version of ResNet. Technol Cancer Res Treat. 2020;19:153303382094748. [doi: 10.1177/1533033820947484]
34. Xie H, Chen Z, Deng J, Zhang J, Duan H, Li Q. Automatic segmentation of the gross target volume in radiotherapy for lung cancer using transresSEUnet 2.5D network. J Transl Med. 2022;20(1):524. [FREE Full text] [doi: 10.1186/s12967-022-03732-w] [Medline: 36371220]
35. Raimondi D, Chizari H, Verplaetse N, Löscher BS, Franke A, Moreau Y. Genome interpretation in a federated learning context allows the multi-center exome-based risk prediction of Crohn's disease patients. Sci Rep. Nov 09, 2023;13(1):19449. [FREE Full text] [doi: 10.1038/s41598-023-46887-2] [Medline: 37945674]
36. Riedel P, von Schwerin R, Schaudt D, Hafner A, Späte C. ResNetFed: federated deep learning architecture for privacy-preserving pneumonia detection from COVID-19 chest radiographs. J Healthc Inform Res. 2023;7(2):203-224. [FREE Full text] [doi: 10.1007/s41666-023-00132-7] [Medline: 37359194]
37. Nazir S, Kaleem M. Federated learning for medical image analysis with deep neural networks. Diagnostics (Basel). 2023;13(9):1532. [FREE Full text] [doi: 10.3390/diagnostics13091532] [Medline: 37174925]
38. Shiri I, Vafaei Sadr A, Akhavan A, Salimi Y, Sanaat A, Amini M, et al. Decentralized collaborative multi-institutional PET attenuation and scatter correction using federated deep learning. Eur J Nucl Med Mol Imaging. 2023;50(4):1034-1050. [FREE Full text] [doi: 10.1007/s00259-022-06053-8] [Medline: 36508026]
39. Zhang M, Qu L, Singh P, Kalpathy-Cramer J, Rubin DL. SplitAVG: a heterogeneity-aware federated deep learning method for medical imaging. IEEE J Biomed Health Inform. 2022;26(9):4635-4644. [doi: 10.1109/jbhi.2022.3185956]
40. Shiri I, Vafaei Sadr A, Amini M, Salimi Y, Sanaat A, Akhavanallaf A, et al. Decentralized distributed multi-institutional PET image segmentation using a federated deep learning framework. Clin Nucl Med. 2022;47(7):606-617. [doi: 10.1097/rlu.0000000000004194]

41. Sarma KV, Harmon S, Sanford T, Roth HR, Xu Z, Tetreault J, et al. Federated learning improves site performance in multicenter deep learning without data sharing. J Am Med Inform Assoc. 2021;28(6):1259-1264. [FREE Full text] [doi: 10.1093/jamia/ocaa341] [Medline: 33537772]
42. Harmon SA, Sanford TH, Xu S, Turkbey EB, Roth H, Xu Z, et al. Artificial intelligence for the detection of COVID-19 pneumonia on chest CT using multinational datasets. Nat Commun. 2020;11(1):4080. [FREE Full text] [doi: 10.1038/s41467-020-17971-2] [Medline: 32796848]
43. Durga R, Poovammal E. FLED-block: federated learning ensembled deep learning blockchain model for COVID-19 prediction. Front Public Health. 2022;10:892499. [FREE Full text] [doi: 10.3389/fpubh.2022.892499]
44. Pati S, Baid U, Edwards B, Sheller M, Wang S, Reina GA, et al. Federated learning enables big data for rare cancer boundary detection. Nat Commun. 2022;13(1):7346. [FREE Full text] [doi: 10.1038/s41467-022-33407-5] [Medline: 36470898]
45. Leroy V, Ananya C, Aiara LG, Andre D, Leonard W. Feasibility of training federated deep learning oropharyngeal primary tumor segmentation models without sharing gradient information. Research Square. Preprint published online 25 July, 2024. [FREE Full text] [doi: 10.21203/rs.3.rs-4644605/v1]
46. Schmidt K, Bearce B, Chang K, Coombs L, Farahani K, Elbatel M, et al. Fair evaluation of federated learning algorithms for automated breast density classification: the results of the 2022 ACR-NCI-NVIDIA federated learning challenge. Med Image Anal. 2024;95:103206. [doi: 10.1016/j.media.2024.103206] [Medline: 38776844]
47. Pati S, Kumar S, Varma A, Edwards B, Lu C, Qu L, et al. Privacy preservation for federated learning in health care. Patterns (N Y). 2024;5(7):100974. [FREE Full text] [doi: 10.1016/j.patter.2024.100974] [Medline: 39081567]
48. Oreiller V, Andrearczyk V, Jreige M, Boughdad S, Elhalawani H, Castelli J, et al. Head and neck tumor segmentation in PET/CT: the HECKTOR challenge. Med Image Anal. 2022;77:102336. [FREE Full text] [doi: 10.1016/j.media.2021.102336] [Medline: 35016077]
49. Iantsen A, Jaouen V, Visvikis D, Hatt M. Squeeze-and-excitation normalization for brain tumor segmentation. In: Crimi A, Bakas S, editors. Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. Cham, Switzerland. Springer International Publishing; 2021.
50. IKNL/vantage6: Docker CLI package for the vantage6 infrastructure. GitHub. URL: https://github.com/IKNL/vantage6/tree/DEV3 [accessed 2024-05-01]
51. Martin F. Featured communities. Zenodo. URL: https://doi.org/10.5281/zenodo.3686944 [accessed 2024-05-06]
52. MaastrichtU-CDS/argos-infrastructure. GitHub. URL: https://github.com/MaastrichtU-CDS/argos-infrastructure [accessed 2024-05-01]
53. MaastrichtU-CDS/projects_argos_argos-code-repo_full-algorithm. GitHub. URL: https://github.com/MaastrichtU-CDS/projects_argos_argos-code-repo_full-algorithm [accessed 2024-05-01]
54. MaastrichtU-CDS/projects_argos_argos-code-repo_researcher-notebooks. GitHub. URL: https://github.com/MaastrichtU-CDS/projects_argos_argos-code-repo_researcher-notebooks [accessed 2024-05-01]
55. OWASP top ten. OWASP Foundation. URL: https://owasp.org/www-project-top-ten/ [accessed 2024-05-02]
56. Moshawrab M, Adda M, Bouzouane A, Ibrahim H, Raad A. Reviewing federated learning aggregation algorithms; strategies, contributions, limitations and future perspectives. Electronics. 2023;12(10):2287. [doi: 10.3390/electronics12102287]
57. Liu P, Xu X, Wang W. Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives. Cybersecurity. 2022;5(1). [doi: 10.1186/s42400-021-00105-6]
58. Boenisch F, Dziedzic A, Schuster R, Shamsabadi S, Shumailov I, Papernot N. When the curious abandon honesty: federated learning is not private. 2023. Presented at: IEEE 8th European Symposium on Security and Privacy (EuroS&P); July 07, 2023; Delft, theNetherlands. [doi: 10.1109/eurosp57164.2023.00020]
59. Sebastian G, George A, Jackson G. Persuading patients using rhetoric to improve artificial intelligence adoption: experimental study. J Med Internet Res. 2023;25:e41430. [FREE Full text] [doi: 10.2196/41430] [Medline: 36912869]

Abbreviations

API: application programming interface
ARGOS: artificial intelligence for gross tumor volume segmentation
CNN: convolutional neural network
CT: computed tomography
FedAvg: federated averaging
FL: federated learning
FML: federated machine learning
GPU: graphics processing unit
GTV: gross tumor volume
HIPAA: Health Insurance Portability and Accountability Act
JWT: JSON Web Token
PHT: Personal Health Train
REST: Representational State Transfer

SAS: secure aggregation server
SRE: secure research environment

Edited by Y Huo; submitted 23.05.24; peer-reviewed by A-T Tran, G Sebastian; comments to author 02.07.24; revised version received 01.10.24; accepted 17.10.24; published 06.02.25

Please cite as:
Choudhury A, Volmer L, Martin F, Fijten R, Wee L, Dekker A, Soest JV
Advancing Privacy-Preserving Health Care Analytics and Implementation of the Personal Health Train: Federated Deep Learning Study
JMIR AI 2025;4:e60847
URL: https://ai.jmir.org/2025/1/e60847
doi: 10.2196/60847
PMID:

AP2FL: Auditable Privacy-Preserving Federated Learning Framework for Electronics in Healthcare

Abbas Yazdinejad, Ali Dehghantanha, *Senior Member, IEEE*, and Gautam Srivastava, *Senior Member, IEEE*

***Abstract*—The growing application of machine learning (ML) techniques in healthcare has led to increased interest in federated learning (FL), which enables the secure and private training of robust ML models. However, conventional FL methods often fall short of providing adequate privacy protection and face challenges in handling non-independent and identically distributed (Non-IID) training data. These shortcomings are of significant concern when employing FL in electronic devices in healthcare. To address these issues, we propose an Auditable Privacy-Preserving Federated Learning (AP2FL) model tailored for electronics in healthcare settings. By leveraging Trusted Execution Environments (TEE), AP2FL ensures secure training and aggregation processes on both client and server sides, effectively mitigating data leakage risks. To manage Non-IID data within the proposed framework, we incorporate the Active Personalized Federated Learning (ActPerFL) model and Batch Normalization (BN) techniques to consolidate user updates and identify data similarities. Additionally, we introduce an auditing mechanism in AP2FL that reveals the contribution of each client to the FL process, facilitating the updating of the global model following diverse data types and distributions. In other words, it ensures the FL process's integrity, transparency, fairness, and robustness. Our results demonstrate that the proposed AP2FL model outperforms existing methods in accuracy and effectively eliminates privacy leakage.**

***Index Terms*—Privacy, FL, auditing, non-IID, healthcare.**

I. Introduction

HEALTHCARE research often involves collecting data from various sources, such as healthcare providers, pharmacies, health insurers, and academic institutions. As health information is highly sensitive, legal and social implications arising from its disclosure render privacy a major concern in the healthcare sector. Consequently, increasing numbers of governments have enacted regulations to safeguard personal information, including electronic devices in healthcare [1]. Applying machine learning (ML) techniques in healthcare has fueled interest in federated learning (FL) to ensure data privacy in electronic healthcare. FL enables privacy preservation in ML by sharing model parameters between clients and servers rather than raw data [2], [3]. Although FL is universally acknowledged as a method that preserves privacy (PP), it isn't exempt from privacy compromises and security risks, including Inversion Attacks and Inference Attacks. These susceptibilities pose potential threats to the inherent privacy attributes of FL [4]. Recent studies have revealed that adversaries can extract sensitive information through model parameter-based attacks [5]. Data reconstruction and inference attacks [6], [7] exemplify such threats, which can occur when ML models inadvertently embed irrelevant information from training data. Both clients and servers can initiate such attacks in FL settings.

A further challenge in FL lies in the varying data distributions among clients. Non-independent and identically distributed (non-IID) data is a particular concern in healthcare, given the diverse demographics, lifestyles, patients, and countries represented in healthcare data sets [8]. This issue can adversely impact healthcare applications, especially those involving FL. Furthermore, it is crucial to understand the contributions of individual participants in a federated setup, which allows for assessing each client's performance and developing a reliable global model. An audit mechanism can also help identify malicious clients who intentionally tamper with training data (poisoning attacks). Thus, incorporating auditability into the FL model design is highly desirable. To successfully implement ML models in healthcare, addressing these challenges in FL is essential. However, existing FL models often lack privacy guarantees and audit capabilities, may leak privacy information, and struggle to handle non-IID data among clients effectively. In response, we propose an Auditable Privacy-Preserving Federated Learning (AP2FL) model for healthcare applications. This model employs Trusted Execution Environments (TEE) on both the client and server sides to prevent privacy leakage in FL settings. The selection of TEE is based on their ability to facilitate trustworthy interactions between domain and data models, which is crucial for maintaining the confidentiality of medical data collaborations.

To address the non-IID data distribution across various clients, we incorporate the Active Personalized Federated Learning (ActPerFL) [9] and Batch Normalization (BN) in our DL model training within an FL setup. The ActPerFL model strikes a productive balance between local and global training phases while also indirectly aiding other clients' training. This model signifies an active learning strategy in FL designed to counter the challenges posed by non-IID data. The approach picks the most significant data points for training, aiming to

Manuscript received 25 April 2023; revised 26 June 2023; accepted 13 September 2023. Date of publication 22 September 2023; date of current version 26 April 2024. *(Corresponding author: Gautam Srivastava.)*

Abbas Yazdinejad and Ali Dehghantanha are with the Cyber Science Lab, Canada Cyber Foundry, University of Guelph, Guelph, ON N1G 4S7, Canada (e-mail: ayazdine@uoguelph.ca; adehghan@uoguelph.ca).

Gautam Srivastava is with the Department of Mathematics and Computer Science, Brandon University, Brandon, MB R7A 6A9, Canada, also with the Research Centre for Interneural Computing, China Medical University, Taichung City 404, Taiwan, and also with the Department of Computer Science and Mathematics, Lebanese American University, Beirut 1102-2801, Lebanon (e-mail: srivastavag@brandonu.ca).

Digital Object Identifier 10.1109/TCE.2023.3318509

lower the data requirements and boost overall performance. Meanwhile, BN plays a crucial role in optimizing model performance by mitigating the variations in input distribution across different batches of data. Additionally, we introduce an auditing mechanism in AP2FL to track the impact of each client in FL, enabling updates to the global model based on diverse data types. Our contributions include:

- Design and implement the privacy-preserving federated learning (PPFL) model for electronics in healthcare.
- In FL-based healthcare environments, we adopt TEE for local training on both client and server sides while the auditor component communicates with them.
- Provide support for non-IID clients in healthcare environments using the ActPerFL model and BN in a DL model while preserving each client's specificity.
- Propose an audit method to not only validate participants' private training data but also ensure the FL process's integrity, transparency, fairness, and robustness.

The rest of the paper can be broken down as follows. Discussion of PP FL works presented in Section II. Section III provides an overview of the AP2FL framework and Section IV gives the security analysis. We assess the proposed model in Section V. Finally, in Section VI, we conclude and outline future directions for the paper.

II. Related Works

Due to the critical nature of PP in FL [10], particularly in healthcare, we provide a concise review of studies focused on mitigating privacy leakage and non-IID data in federated environments. FL research has explored membership inference to determine if a specific data sample is part of the training set [11], [12]. Moreover, investigators have found that exchanged data can be exploited to extract unintended private information, such as the presence of eyeglasses [13]. Li et al. [14] demonstrated that the most significant privacy threat arises when determining the optimal input-label pairs corresponding to exchanged gradients. Building on this approach, [15] proposed an analytical method to extract label information, although its applicability is limited to shallow networks trained on low-resolution images. Geiping et al. [6] extended this to restore ImageNet-level high-resolution data from deeper networks using a magnitude-invariant loss design.

In more recent work, Yin et al. [16] employed BN statistics to encode a strong prior, enabling image batch reconstruction. Furthermore, the study by [17] introduced multi-agent federated reinforcement learning, focusing on a mobile and fog agents-based paradigm. This approach aims to create a consumer-centric, cyborg-efficient training and testing system within the Internet of Medical Things (IoMT). In non-FL, Sun et al. [18] provided PP in medical record searching for IoT Healthcare. To achieve PP in FL, [19] proposed a PPFL method via TEE. Greedy layer-wise training was utilized for local training within the trusted area on clients; however, this approach entails synchronization and communication overheads. The non-IID assumption in this work was oversimplified, with each client selecting samples from only two random classes. Another study [20] presented a personalized

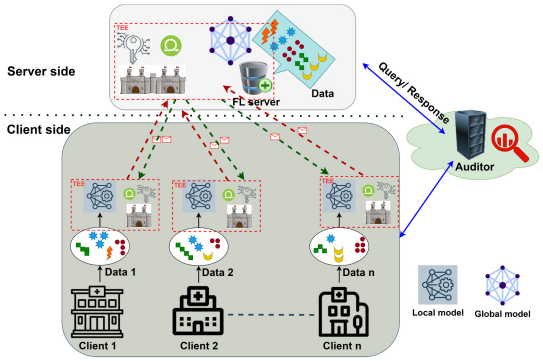


Fig. 1. A schematic diagram of the AP2FL framework.

FL approach with BN for healthcare, which supports non-IID scenarios, addresses domain shifts, and generates personalized models for local clients. However, this method neglects PPFL concerns. Although comprehensive experiments demonstrated satisfactory accuracy, the popular FL algorithm FedAvg [21] often outperforms other methods [22]. FedAvg [23] is ill-suited for handling non-IID data from multiple clients, as it directly averages parameter values from all participants. Several algorithms have been developed to address non-IID situations, such as FedProx [24], which is tailored for non-identical data. However, FedProx cannot generate personalized models for clients, as it learns a global model for all participants. Accessing large public datasets like FedHealth [25] is often infeasible in real-world applications. FedBN [26] addresses the non-IID issue by learning local BN layers for each client but fails to consider cross-client similarities that could enhance the personalization. Although previous research has aimed to achieve PP in FL or address non-IID issues in FL environments, no comprehensive work has tackled these challenges in healthcare applications. Furthermore, no additional PP measures or defences, such as auditable features, have been incorporated in FL environments within healthcare. This paper highlights these concerns as our primary contributions and discusses them in detail throughout the work.

III. The AP2FL Framework

This section introduces a novel auditable PPFL model for healthcare that aims to minimize privacy leakage and non-IID issues in federated healthcare environments. Fig. 1 provides a comprehensive view of the proposed model, which includes multiple hospitals (clients A, B, etc.) with distinct data distribution characteristics (e.g., clients A and B have different activities and lifestyles), an FL server, and an Auditor component. In our proposed model, we employ TEE to prevent data leakage at both client and server levels, support non-IID situations by implementing the ActPerFL model and BN method, and incorporate an auditing feature into our model design. The Auditor component serves as a security policy manager, facilitating client and server communication. In the subsequent sections, we elaborate on TEE, the ActPerFL model, BN, and auditing.

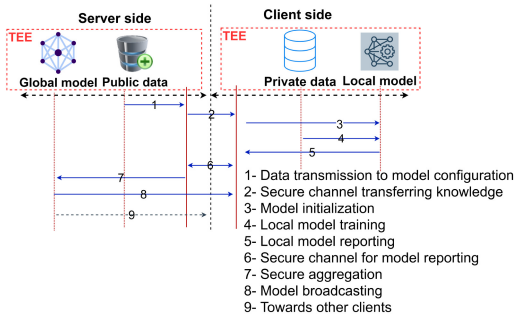


Fig. 2. Workflow of the AP2FL framework.

A. Remove Privacy Leakage

To mitigate privacy leakage in federated healthcare environments, we adopt a TEE-based approach, as TEE can effectively prevent data leakage on both server and client sides within a federated context. Executing ML and DL models within TEE can conceal model parameters from Rich Operating System Execution Environment (REE) adversaries and maintain privacy, as already demonstrated during model training and lightweight data analytics. TEE are emerging security technologies that offer promising solutions for mitigating system attacks [27]. They facilitate data processing with fine-grained access control and memory protection through a hardware root of trust [27]. TEE have various use cases, including cloud computing, the Internet of Things (IoT), multi-party computations (MPC), and ML.

In the AP2FL model, privacy risk is dependent on the aggregation level, as gradients remain inaccessible to adversaries while being updated within TEE. This approach also defends against privacy attacks such as data reconstruction attacks (DRAs) and parameter inference attacks (PIAs) [19]. As illustrated in Fig. 2, the AP2FL model is schematically depicted, showcasing the elimination of privacy leaks through the use of TEE. The server uses random weights or publicly available information to assign weights to the global model. Utilizing random or public data for global model initialization in AP2FL promotes privacy and fairness, despite potential slower convergence or initial performance issues. In this way, the model m is prepared for broadcast after it has been initialized (m). By using TEE, the server constructs secure communication with participating clients. During steps 1 and 2, data will be transmitted to the model configuration through a secure channel. Once the model is initialized, clients do local training and model initialization (steps 3 and 4). A client loads the received model parameters from the server and then decrypts and loads the target model in their TEE. In step 5, the model parameters will be reported, and in step 6, the server will receive the model reporting. After receiving client updates, servers decrypt weights and parameters and perform secure aggregation and averaging inside their TEE. The server side does secure aggregation and broadcasts the model; steps 8 and 9 update, other participants. Steps 4-9 must be repeated until the model m converges or a certain amount of training rounds have been completed. Also, the model training and aggregation procedure is detailed in Algorithm 1 for both the client and server sides.

Algorithm 1 Workflow of TEE in AP2FL

Input: Global model (m), Number of clients: N, Data (D), Communication rounds (R), Local data [X] and labels [Y], Number of local training epochs: E
Output: Updated parameters of local model
  Aggregated final parameters:$\{P_0, P_1, \ldots, P_m\}$
Initialization step
  Server side:
  Initialize participating client list $T = []$
  Load D $\rightarrow$ m
for *n=1 to N* **do**
    Build secure channel
    Data Transmission in the list T
    Initialize model m $\rightarrow$ **TEE**
end
Client-side:
for *e=1 to E* **do**
  for *t=1 to T* **do**
    Model configuration
    Local model initialization
  end
  model running $\rightarrow$ **TEE**
end
Training step in FL
for $r = 1$ *to* R **do**
  for $\{x, y\}$ *to* X **do**
    Local training
    local updating
  end
  Model reporting $\rightarrow$ **TEE**
end
Secure aggregate parameters:$\{P_0, P_1, \ldots, P_m\}$ $\rightarrow$ **TEE**
Model Broadcasting $\rightarrow$ all clients
Return Aggregated final parameters

The server initializes the global model using random weights or publicly available information. In this manner, the model m is prepared for broadcasting after initialization. Utilizing TEE, the server establishes secure communication with participating clients. During steps 1 and 2, data is transmitted to the model configuration through a secure channel. Following the model initialization, clients perform local training and model initialization (steps 3 and 4). A client loads the received model parameters from the server, decrypts them, and loads the target model into their TEE. In step 5, the model parameters are reported, and in step 6, the server receives the model reporting. Upon receiving client updates, servers decrypt weights and parameters, then carry out secure aggregation and averaging within their TEE. The server performs secure aggregation, broadcasts the model, and updates other participants in steps 8 and 9. Steps 4-9 must be repeated until the model m converges or a predefined number of training rounds are completed. The model training and aggregation procedure is also outlined in Algorithm 1 for both the client and server sides. It begins with initializing the global model using public data, establishing secure channels, and local training on the client side. It concludes with the server securely aggregating local updates and broadcasting the updated global model to all clients.

B. Addressing Non-IID Data in AP2FL

Conventional FL methods, such as FedAvg [21] and other approaches [28], often struggle to effectively handle non-IID data, particularly when clients possess limited data to train a model in FL settings. A significant contribution of AP2FL in healthcare environments is its ability to handle non-IID situations by incorporating ActPerFL [9] and BN in our DL models within the FL framework. Specifically, the data quality is evaluated for each client based on the statistical information obtained from multiple clients, which subsequently informs the update of the model's hyperparameters and BN layer.

ActPerFL represents a self-aware, personalized FL approach that intelligently balances the training of individual local models with the global model, indirectly contributing to the training of other clients' models [9]. BN integration in DL is crucial to improving the model's performance and managing domain shifts effectively. BN layers provide sufficient statistics, including mean and standard deviation, which are essential in the process [29]. To represent the data distributions of clients, BN is primarily utilized. Consequently, clients' feature distributions are maintained through local BN. Additionally, BN-related statistics are used to determine client similarity, enhancing support for non-IID scenarios through weighted aggregation.

1) Problem Formulation in AP2FL: In FL, there are N different clients (organizations or users), indicated as $\{H_1, H_2, \ldots, H_N\}$ and each client has its own dataset, i.e.,$\{D_1, D_2, \ldots, D_N\}$. Each dataset D_i includes two parts, i.e., a train dataset Di_{train} and Di_{test}. Therefore, the entire number of samples is: $n_i = ni_{train} + ni_{test}$ and $D_i = Di_{train} \cup Di_{test}$. The distributions of the datasets are all different, $P_{Di} \neq P_{Dj}$. Clients have their own models defined as $\{F_i\}_{i=1}^{N}$. To learn a good model, we aggregate information from all clients in order to learn f_i for the local dataset of each client D_i without private data leakage:

$$\min_{\{F_i\}_{i=1}^{N}} = \frac{1}{N} \sum_{1}^{N} \frac{1}{n\frac{te}{i}} \sum_{j=1}^{n\frac{te}{i}} \eta(Di) \tag{1}$$

where η is a loss function in Eq. (1).

2) Incorporating ActPerFL and Batch Normalization: To effectively apply ActPerFL, we need three critical components: 1- Appropriate initialization for local clients at each round, 2- Automatic determination of local training steps, and 3- Discrepancy-aware aggregation rules for the global model. Algorithm 2 outlines the overall process of ActPerFL and the BN approach. At round t, the parameters from client i are represented as $\omega_i^t = \alpha_i^t + \beta_i^t$. After updating ω_i^t with local data from the i-th client, the parameters become $\omega_i^{*t} = \alpha_i^{*t} + \beta_i^{*t}$. The $*$ notation indicates updated parameters. The server uses the updating strategy to aggregate data via Eq. (2):

$$\omega_i^{t+1} = \omega^* i^t \Rightarrow \alpha i^{t+1} + \beta_i^{t+1} = \alpha^* i^t + \beta^* i^t \tag{2}$$

To address this issue, the prediction layer aggregates mean and variance values over the entire training set. As training becomes distributed, each device has its local data, and the batch is also distributed. Each device trains on a local batch before global aggregation. During training, BN performs normalization on local batch statistics. This is crucial when handling non-identical data distributions, as each device's batch statistics do not represent global statistics, leading to prediction discrepancies.

Algorithm 2 The ActPerFL and BN

Input: Global model m, client activity rate C, learning rate η, data of N clients $D_1, D_2, \ldots, D_N$
Output: Client models F_i for $i = 1, 2, \ldots, N$ using ActPerFL and BN
Initialize: Calculate empirical variance [9], set convergence criteria and maximum rounds Update global model m **while** *not converged or maximum rounds not reached* **do**
  for *n=1 to N* **do**
    Distribute global model m to client $\rightarrow C$
    Each client computes $D_i = D_{i,train} \cup D_{i,test}$
    Compute initial parameters $\omega_i^{(t)} = \alpha_i^{(t)} + \beta_i^{(t)}$ **while** *client i not converged* **do**
      Perform local training using SGD with BN
      Automatically determine local training steps
      Update local parameters to $\omega_i^{*(t)} = \alpha_i^{*(t)} + \beta_i^{*(t)}$
      Aggregate local parameters on the server using discrepancy-aware aggregation rules DUpdate global model $\rightarrow m$
    end
  end
end

The weights are evaluated by ω. We calculate only the statistics of BN layers' inputs. The BN statistics of the i^{th} client are acquired by Eq. (3):

$$\left(\varphi^i, \mu^j\right) = \left[\left(\varphi^{i1}, \mu^{j1}\right), \left(\varphi^{i2}, \mu^{j2}\right), \ldots, \left(\varphi^{in}, \mu^{jn}\right)\right] \tag{3}$$

The distance between two clients i and j is computed using the Kantorovich-Rubinstein metric. This metric, which measures the distance between clients' probability distributions, is particularly suited for FL due to its sensitivity to displacements in data distribution, robustness to noise, and adherence to the principles of a mathematical distance (symmetry and triangle inequality), providing a reliable measure of distribution discrepancies. The computation is based on Eq. (4):

$$d_{ij} = \sum_{l=1}^{L} \left(\left|\mu^j - \mu^i\right|^2 + \left|\varphi^j - \varphi^i\right|^2\right)^{\frac{1}{2}} \tag{4}$$

A large d_{ij} value indicates a significant distribution distance between the i-th and j-th clients. Therefore, the larger d_{ij} is, the less similar the two clients are, resulting in a smaller ω_{ij}.

C. Auditing Method in AP2FL

In AP2FL, auditing is essential to verify all clients' contributions of model gradients and weights to the FL server. This process, while complex, is necessary to ensure the global model accurately reflects diverse data types and distributions.

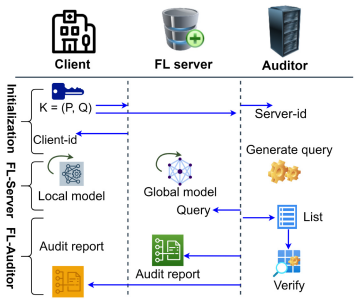


Fig. 3. Workflow of the AP2FL framework.

To streamline this process, we've established a comprehensive auditing protocol that integrates smoothly into the AP2FL framework.

1) Auditing Protocol: The protocol has been divided into three major phases that work harmoniously to provide reliable auditing:

- *Phase 1: Initialization*
  1) Each client, hospital, during initialization of its local models in TEE generates a public-private key-pair $K = (sk; pk)$ and public parameters based on the security parameter λ.
  2) Each client should register with public key (pk) in both FL server and auditor and get $Client - ID$.
  3) Auditor detects all FL servers in the FL setup and gets the list of FL server and their $server - ID$.
- *Phase 2: FL-Server*
  1) Run the FL environment, aggregating model parameters in the FL server and updating the global model's parameters.
  2) The FL server generates authentication tags, $SigVerif$, for each client that received its update, records this issue and counts the number of participants while updating the global model.
- *Phase 3: FL-Auditor*
  1) Auditor sends the query to the FL server. If the FL server had updated the global model, the server would like a response.
  2) The auditor gets the number of participants from the FL server after updating the global model.
  3) The auditor verifies the aggregation outcome in the FL server, recognizing participants via their public keys. This mechanism addresses any client's contention about the data included in the server's aggregation, and the auditor can validate or refute the claim.
  4) Lastly, the auditor generates an audit report summarizing the model update events and sends the response to the FL server and the participants, clarifying their participation in the global model update.

The auditing mechanism ensures transparency and validity in the FL process, confirming that all contributions are recognized and the global model update is accurately performed. Fig. 3 presents the auditing process in the AP2FL framework, displaying the interactions among clients, the FL server, and the auditor.

IV. Security Analysis

A. Threat Model and Defense Strategies

Incorporating TEE in our FL model equips us to address potential threats on both the client and server sides.

Client-side: These involve adversaries compromising client devices to access sensitive data during local model training or manipulate local model updates. *Defense Strategy:* TEE secure the model training and data processing within an isolated environment. Therefore, even in case of device compromise, sensitive information remains protected inside the TEE.

Server-side: These pertain to unauthorized attempts to access the global model or tamper with the server's aggregation process. *Defense Strategy:* TEE fortify security by performing client update aggregations in a secure server-side environment, preventing unauthorized manipulations or access to sensitive client updates. With the auditing protocol, we effectively combat these threats and preserve privacy in our FL model. We enhance our FL model's PP through TEE usage by ensuring secure local training, encrypted model updates, and secure aggregation. However, we do not explore TEE and their SDKs concerning side-channel and physical attacks.

B. Privacy-Preserving Proof

To formally prove privacy preservation for the threat models, we will define the security properties of the FL model and provide mathematical proof for the defensive strategies using TEE. Let D_i denote the sensitive data held by client i and M_i represent the local model updates computed by client i. Let A_{agg} be the aggregation function performed on the server, and M_{global} be the global model after aggregation.

Security Property 1 (Confidentiality): For any adversary $\mathcal{A}$ and client i, the probability that $\mathcal{A}$ can access sensitive data D_i or local model updates M_i should be negligible. Mathematically, we can define this property as Eq. (5):

$$Pr[\mathcal{A} \text{ can access } D_i \text{ or } M_i] \leq negl(\lambda) \tag{5}$$

where λ is the security parameter, and $negl(\lambda)$ represents a negligible function in λ.

Security Property 2 (Integrity): For any adversary, $\mathcal{A}$, the probability that $\mathcal{A}$ can manipulate the aggregation process A_{agg} or tamper with the global model M_{global} should be negligible. Mathematically, we can define this property as Eq. (6):

$$Pr\big[\mathcal{A} \text{ can manipulate } A_{agg} \text{ or tamper with } M_{global}\big] \leq negl(\lambda) \tag{6}$$

Proof of Confidentiality and Integrity Using TEE: Since TEE provide a secure and isolated execution environment, we can assume that the probability of adversaries compromising a TEE is negligible. We denote this as Eq. (7):

$$Pr\big[\mathcal{A} \text{ can compromise TEE}\big], \leq negl(\lambda) \tag{7}$$

Now, it is feasible to demonstrate that the defensive tactics involving TEE meet the stringent security requirements of confidentiality and integrity.

Confidentiality: For client-side threats, TEE protect the sensitive data D_i and local model updates M_i. Thus, the probability that an adversary can access D_i or M_i is bounded

by the probability of compromising the TEE, which is negligible.

Integrity: For server-side threats, TEE ensure that the aggregation process A_{agg} and global model M_{global} remain secure. Thus, the probability that an adversary can manipulate A_{agg} or tamper with M_{global} is also bounded by the probability of compromising the TEE, which is negligible.

Based on these proofs, we can conclude that the use of TEE in the FL model effectively addresses the identified threat models and ensures privacy preservation by satisfying the confidentiality and integrity security properties.

C. Effectiveness of Auditing Mechanism

Our proposed auditing mechanism fortifies the FL process with integral characteristics of integrity, transparency, fairness, and robustness while simultaneously ensuring privacy and security. This balance, fundamental to the mechanism's functionality, is supported by corresponding theorems and lemmas, affirming its effectiveness.

Theorem 1: Integrity and Transparency.

Proof: To prove the integrity and transparency of the auditing mechanism in the AP2FL framework, we need to show that the contributions of each client are accurately incorporated into the global model and that the FL server provides accurate information about clients' participation in the process. ■

Lemma 1 (Authentication Tags): The authentication tags, *SigVerif*, generated by the server can be used to verify the contributions of each client to the aggregation process, ensuring that their local model updates are properly accounted for. This guarantees the integrity of the FL process.

Lemma 2 (Query Response): The FL server's ability to provide accurate information about clients' participation in the FL process in response to a query from the auditor ensures transparency in the FL process.

Theorem 2: Fairness.

Proof: To prove the fairness of the auditing mechanism in the AP2FL framework, we need to show that the weights assigned to each client's contribution in the aggregation process are determined fairly based on their contributions. ■

Lemma 3: Weight Calculation: The auditor accurately calculates the weights W_i based on the distances δ_i, ensuring a fair representation of each client's contribution in the aggregation process. This guarantees the fairness of the FL process.

Theorem 3: Robustness.

Proof: To prove the robustness of the auditing mechanism in the AP2FL framework, we need to show that the auditing mechanism can identify and mitigate the impact of adversarial clients in the FL process. ■

Lemma 4: Adversarial Client Detection: The auditing mechanism can detect clients in $\mathcal{A}$ by identifying discrepancies in their contributions, such as malicious or unregistered clients. This contributes to the robustness of the FL process.

Lemma 5: Mitigating Adversarial Impact: By identifying and excluding clients' contributions in $\mathcal{A}$, the auditing mechanism can reduce the impact of adversarial clients on the global model, ensuring the robustness of the FL process.

V. Performance Evaluation

To evaluate the AP2FL framework, we have equipped our testing system with 16 GB DDR4 memory, RTX 2070 GPU, and Intel (R) Core(TM) i7 CPU-10700KF@ 3.80 GHz, which support Intel Soft Guard Extensions (SGX). To build our setup, we used libraries such as *PyTorch*/*torch* and integrated *PySyft* (Python library for secure and private Deep Learning) with Intel SGX as PySyft + Intel SGX [30]. The privacy risks associated with DL have been extensively studied in Convolutional Neural Networks (CNN). As a result, the proposed framework uses the LetNet5 [21] model, one of the most well-known CNN models in modern DL-based computer vision.

A. Datasets

The proposed framework was assessed using the MedMNIST collection [31], a standardized collection of biomedical images similar to MNIST that includes 12 2D datasets and 6 3D datasets. Each image in MedMNIST is preprocessed to 28×28 (2D) or $28 \times 28 \times 28$ (3D), making it accessible to users without background knowledge. From the 12 2D datasets in MedMNIST, we selected the three with the most classes: OrganMNIST Sagittal, OrganMNIST Coronal, and OrganMNIST Axial [31].

B. Experimental Analysis

As shown in Fig. 4, the distribution of samples across 20 clients is visualized for OrganMNIST Sagittal, Coronal, and Axial datasets. In MedMNIST, training data is used for training as well as validation data is used for fine-tuning hyperparameters, and test data is used for testing. Therefore, the LeNet5 [21] model uses these datasets for training and prediction. Assuming a learning rate of 10^{-2}, the models are trained using cross-entropy loss and SGD optimization. If there is no local update epoch setting, our default value is $E = 3$, where E means training epochs in one round. Our method's θ value is 0.5 since θ has little influence on accuracy and only affects convergence speed. The AP2FL model is compared with common FL methods and FL methods designed for non-ID data: *FedAvg* [21], all client models are aggregated without non-IID data. *FedProx* [24], Update FedAvg to include a proximal term and allow for partial information aggregation. *FedPer* [32], ayers are preserved locally by each client. *FedBN* [26], each client preserves the local BN.

Utilizing three distinct MedMNIST datasets, Table I, Table II, and Table III provide a detailed breakdown of classification results per client, respectively. The inclusion of an auditing mechanism in our proposed framework enables us to monitor each participant's contributions during the federated learning process. Analyzing these results, it is evident that our approach demonstrates superior performance in terms of accuracy, consistently outperforming other tested methods. In particular, while FedBN struggles to achieve satisfactory results due to its focus on handling feature shifts, our model successfully navigates label shifts via the ActPerFL, boosting its accuracy. The presented results in Table I, Table II, and Table III represent each user's optimal performance during runtime, respectively. The term 'Ave', used in our tables,

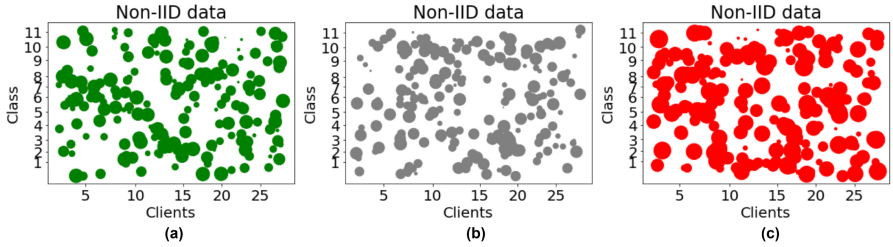


Fig. 4. Allocated samples to each client per class.

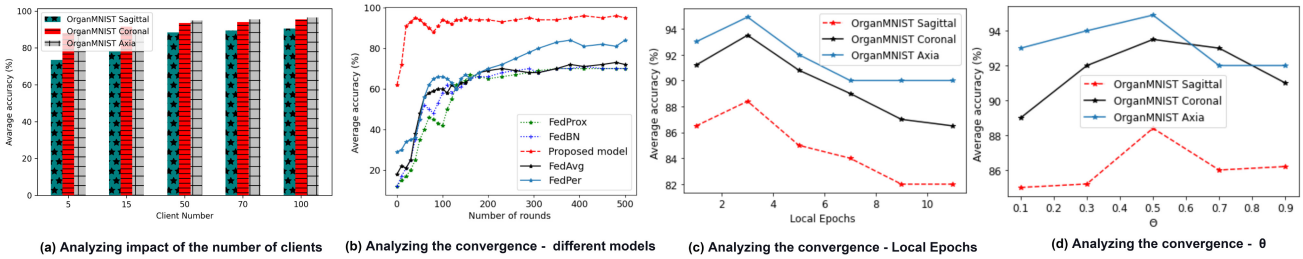


Fig. 5. Parameter sensitivity analysis.

TABLE I
ACCURACY OF USERS ON ORGANMNIST SAGITTAL

User	FedAvg	FedBN	FedProx	FedPer	Proposed model
5	47.39	71.87	66.3	85.74	93.51
10	68.45	78.8	52.3	85.74	90.69
15	78.2	91.78	71.25	64.56	88.75
17	36.55	45.04	41.93	58	98.5
20	41.3	53.48	83.57	92.42	89.1
Ave	64.8	80.44	64.9	74.55	88.4

TABLE II
ACCURACY OF USERS ON ORGANMNIST CORONAL

User	FedAvg	FedBN	FedProx	FedPer	Proposed model
5	79.22	88.7	79.73	77.87	92.8
10	85.83	96.46	80.61	80.07	98.82
15	81.23	61.38	90.39	51.52	91.89
17	67.45	83.81	73.99	65.37	99
20	54.56	90.71	88.18	70.61	89.5
Ave	78.36	88.18	78.3	71.14	93.5

TABLE III
ACCURACY OF USERS ON ORGANMNIST AXIA

User	FedAvg	FedBN	FedProx	FedPer	Proposed model
5	80.03	93.14	80.37	83.22	94.5
10	92.32	84.1	77.58	41.71	99.1
15	74.66	87.84	82.98	53.22	91.79
17	83.31	62.77	74.39	62.06	88.1
20	81.32	93.2	90.42	93.95	96.6
Ave	84.06	89.28	84.11	70.02	94.9

stands for 'Average', referring to the average value based on all participants.

1) Analyzing Client Number, Convergence, and Parameter Sensitivity: Fig. 5(a) underscores how variations in client distributions contribute to the differential difficulty levels experienced among clients. This underscores the value of using specific clients' local data to achieve particular distributions, further enhancing our method's accuracy. As the number of clients increases, accuracy increases because we have more parameters for aggregation (Fig. 5(a)). Moreover, we examine our method's convergence and parameter sensitivity. As shown in Fig. 5(b), our method almost converged after the 20th round. In contrast, other methods require more than 350 rounds. On three MedMNIST benchmarks, we tested AP2FL's parameter sensitivity by local epochs, and value of θ. In our presentation of parameter sensitivity, one parameter was changed, and the other parameters were fixed. Our method achieved acceptable results that according to the graph in Fig. 5(c), the best value for local epochs is 3, and our method is the best with this value. Insufficient communication between the clients and keeping the total number of epochs constant has caused it to decrease with more local epochs. Based on Fig. 5(d), our method's average accuracy and convergence rate are affected by θ. As $\theta = 0.5$, AP2FL consistently performs better.

TABLE IV
AUDITING SUMMARY BASED ON PARTICIPANTS

Query	Info	Valid	CF	Tag ID	RND
1	Get clients' IDs & Tags	Yes	0.9	—	1
4	Audit client# 5	Yes	0.05	7dc023b5	3
5	Audit client# 8	Yes	0.1	df73e6c7	4
9	Get clients' IDs & Tags	Yes	0.65	—	6
15	Audit client# 20	No	0.2	2b9100c6	5

C. Auditing Functionality

Table IV summarizes the information that can be inferred based on one of the datasets (OrganMNIST Axia) for analyzing the functionality of the audit protocol. The *Query* field refers to the queries sent from the Auditor to the FL server, and the *Info* field indicates the query context like #*Participants*. The *Valid* field indicates whether clients have registered on both the server and Auditor sides. The *Tag ID* fields indicate the total number of clients and their unique IDs during runtime. The *CF* field represents the ratio of clients updated to the server. The Auditor uses Euclidean distance to measure the difference between the parameter models for different clients and the global model. The *RND* field indicates the

round in which the global model was updated. 'Audit client' represents individual clients or nodes participating in the FL process, each with a unique model contributing to the update of the global model. For example, if 25 clients participated in updating the global model during several rounds in a federated setup, the *Query* = 1 query would ask the FL server to provide all client IDs and tags, and the *Valid* field would be "Yes" since all clients were valid for updating the global model in *RND* 1. In this round, the *CF* field indicates that 90% of clients updated the server, indicating their significant impact on the global model. Similarly, in *Query* = 15, the Auditor audits *client* 20 and discovers that it has not registered with the Auditor but has participated in updating the global model on the FL server. The Auditor reports any malicious or unregistered clients and suspends that round of updating the global model.

VI. Conclusion and Future Work

We proposed the Auditable Privacy-Preserving Federated Learning (AP2FL) framework for electronics in healthcare. that preserves privacy via Trusted Execution Environments (TEE) for secure aggregation and local training on the client and server sides. Also, to address the non-IID issue, we used the combination of ActPerFL and Batch Normalization (BN) to learn similarities between clients, automated tune local model parameters, and model aggregation. We also devised an auditing method to monitor the impact of each client in Federated Learning (FL) for averaging and updating the global model based on different data distributions. In future research, we will enhance the AP2FL with blockchain for training Machine Learning (ML) models and auditing, thereby increasing transparency and traceability. Furthermore, the potential single-point-of-failure issue in AP2FL with auditors could be mitigated through multiple independent auditors or a consensus-based approach.

References

[1] M. Wazid, A. K. Das, and S. Shetty, "BSFR-SH: Blockchain-enabled security framework against ransomware attacks for smart healthcare," *IEEE Trans. Consum. Electron.*, vol. 69, no. 1, pp. 18–28, Feb. 2023.

[2] B. Jayaraman and D. Evans, "Evaluating differentially private machine learning in practice," in *Proc. 28th USENIX Security Symp. (USENIX Security)*, 2019, pp. 1895–1912.

[3] Y.-T. Lee, W.-H. Hsiao, Y.-S. Lin, and S.-C. T. Chou, "Privacy-preserving data analytics in cloud-based smart home with community hierarchy," *IEEE Trans. Consum. Electron.*, vol. 63, no. 2, pp. 200–207, May 2017.

[4] X. Jiang, X. Zhou, and J. Grossklags, "Comprehensive analysis of privacy leakage in vertical federated learning during prediction," in *Proc. Privacy Enhanc. Technol.*, vol. 2022, no. 2, 2022, pp. 263–281.

[5] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proc. IEEE Symp. Security Privacy (SP)*, 2019, pp. 691–706.

[6] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 16937–16947.

[7] D. Saraswat et al., "Blockchain-based federated learning in UAVs beyond 5G networks: A solution taxonomy and future directions," *IEEE Access*, vol. 10, pp. 33154–33182, 2022.

[8] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," *J. Healthcare Informat. Res.*, vol. 5, no. 1, pp. 1–19, 2021.

[9] H. Chen et al., "ActPerFL: Active personalized federated learning," in *Proc. Workshop Feder. Learn. Nat. Lang. Process. (ACL)*, 2022, p. 7.

[10] A. Yazdinejad, R. M. Parizi, A. Dehghantanha, and H. Karimipour, "Federated learning for drone authentication," *Ad Hoc Netw.*, vol. 120, Sep. 2021, Art. no. 102574.

[11] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *Proc. IEEE Symp. Security Privacy (SP)*, 2019, pp. 739–753.

[12] A. Yazdinejad, A. Dehghantanha, R. M. Parizi, M. Hammoudeh, H. Karimipour, and G. Srivastava, "Block hunter: Federated learning for cyber threat hunting in blockchain-based IIoT networks," *IEEE Trans. Ind. Informat.*, early access, Feb. 7, 2023, doi: 10.1109/TCE.2023.3242375.

[13] K. Ganju, Q. Wang, W. Yang, C. A. Gunter, and N. Borisov, "Property inference attacks on fully connected neural networks using permutation invariant representations," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2018, pp. 619–633.

[14] Z. Li, J. Zhang, L. Liu, and J. Liu, "Auditing privacy defenses in federated learning via generative gradient leakage," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10132–10142.

[15] B. Zhao, K. R. Mopuri, and H. Bilen, "IDLG: Improved deep leakage from gradients," 2020, *arXiv:2001.02610*.

[16] H. Yin, A. Mallya, A. Vahdat, J. M. Alvarez, J. Kautz, and P. Molchanov, "See through gradients: Image batch recovery via gradinversion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 16337–16346.

[17] P. Tiwari, A. Lakhan, R. H. Jhaveri, and T.-M. Gronli, "Consumer-centric Internet of Medical Things for cyborg applications based on federated reinforcement learning," *IEEE Trans. Consum. Electron.*, early access, Feb. 7, 2023, doi: 10.1109/TCE.2023.3242375.

[18] Y. Sun, J. Liu, K. Yu, M. Alazab, and K. Lin, "PMRSS: Privacy-preserving medical record searching scheme for intelligent diagnosis in IoT healthcare," *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 1981–1990, Mar. 2022.

[19] F. Mo, H. Haddadi, K. Katevas, E. Marin, D. Perino, and N. Kourtellis, "PPFL: Privacy-preserving federated learning with trusted execution environments," in *Proc. 19th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2021, pp. 94–108.

[20] W. Lu et al., "Personalized federated learning with adaptive batchnorm for healthcare," *IEEE Trans. Big Data*, early access, May 23, 2022, doi: 10.1109/TBDATA.2022.3177197.

[21] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Stat.*, 2017, pp. 1273–1282.

[22] T. Ching et al., "Opportunities and obstacles for deep learning in biology and medicine," *J. Roy. Soc. Interface*, vol. 15, no. 141, 2018, Art. no. 20170387.

[23] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAVG on non-IID data," 2019, *arXiv:1907.02189*.

[24] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, vol. 2, 2020, pp. 429–450.

[25] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "FedHealth: A federated transfer learning framework for wearable healthcare," *IEEE Intell. Syst.*, vol. 35, no. 4, pp. 83–93, Jul./Aug. 2020.

[26] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "FedBN: Federated learning on non-IID features via local batch normalization," 2021, *arXiv:2102.07623*.

[27] M. Müller, A. Simonet-Boulogne, S. Sengupta, and O. Beige, "Process mining in trusted execution environments: Towards hardware guarantees for trust-aware inter-organizational process analysis," in *Proc. Int. Conf. Process Min.*, 2022, pp. 369–381.

[28] H. Gao, A. Xu, and H. Huang, "On the convergence of communication-efficient local SGD for federated learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 7510–7518.

[29] Y. Li, N. Wang, J. Shi, X. Hou, and J. Liu, "Adaptive batch normalization for practical domain adaptation," *Pattern Recognit.*, vol. 80, pp. 109–117, Aug. 2018.

[30] "PySyft + Intel SGX." Accessed: Apr. 15, 2020. [Online]. Available: https://blog.openmined.org/pysyft-pytorch-intel-sgx/

[31] J. Yang, R. Shi, and B. Ni, "Medmnist classification decathlon: A lightweight AutoML benchmark for medical image analysis," in *Proc. IEEE 18th Int. Symp. Biomed. Imag. (ISBI)*, 2021, pp. 191–195.

[32] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," 2019, *arXiv:1912.00818*.

Abbas Yazdinejad received the B.Sc. degree in computer engineering from the Shahid Bahonar University of Kerman, Iran, and the M.Sc. degree specializing in computer system architecture from the University of Isfahan, Iran. He is currently pursuing the Ph.D. degree with the Cyber Science Lab, Canada Cyber Foundry, School of Computer Science, University of Guelph, ON, Canada. He is also associated with the Decentralized Science Lab, Kennesaw State University, Kennesaw, GA, USA, and with the Smart Cyber-Physical Lab, University of Calgary. His research interests span cybersecurity, blockchain, federated learning, SDN, and IoT/IIoT.

Ali Dehghantanha (Senior Member, IEEE) is an Academic Entrepreneur in Cybersecurity, the Canada Research Chair in Cybersecurity and Threat Intelligence, and an Associate Professor of Cybersecurity with the University of Guelph, ON, Canada. He is the Director of Cyber Science Lab—a research lab dedicated to advanced research and training in cybersecurity—and the Director and a Founder of the Master of Cybersecurity and Threat Intelligence Program, University of Guelph.

Gautam Srivastava (Senior Member, IEEE) received the B.Sc. degree from Briar Cliff University, USA, in 2004, and the M.Sc. and Ph.D. degrees from the University of Victoria, Victoria, BC, Canada, in 2006 and 2012, respectively. He is currently a Full Professor with Brandon University, Brandon, MB, Canada, where he is currently active in various professional and scholarly activities. He also holds research positions with China Medical University, Taiwan, as well as Lebanese American University, Lebanon. He is popularly known as Dr. G. In his five years as a research academic, he has published a total of 180 papers in high-impact conferences in many countries and in high-status journals (SCI and SCIE) and has also delivered invited guest lectures on big data, cloud computing, Internet of Things, and cryptography at many universities worldwide. He is active in research in the field of cryptography, data mining, security and privacy, and blockchain technology. He is an editor of several SCI/SCIE journals. He is also an associate editor of many IEEE journals.

electronics

MDPI

Article

Masking and Homomorphic Encryption-Combined Secure Aggregation for Privacy-Preserving Federated Learning

Soyoung Park *, Junyoung Lee, Kaho Harada and Jeonghee Chi

Department of Computer Science and Engineering, Konkuk University, Seoul 05029, Republic of Korea; junzero@konkuk.ac.kr (J.L.); mamt4825@konkuk.ac.kr (K.H.); jhchi@konkuk.ac.kr (J.C.)
* Correspondence: soyoungpark@konkuk.ac.kr; Tel.: +82-2-450-0482

Abstract: Secure aggregation of local learning model parameters is crucial for achieving privacy-preserving federated learning. This paper presents a novel and practical aggregation method that effectively combines the advantages of masking-based aggregation with those of homomorphic encryption-based techniques. Each node conceals its local parameters using a randomly selected mask, independently chosen, thereby eliminating the need for additional computations to generate or exchange mask values with other nodes. Instead, each node homomorphically encrypts its random mask using its own encryption key. During each federated learning round, nodes send their masked parameters and the homomorphically encrypted mask to the federated learning server. The server then aggregates these updates in an encrypted state, directly calculating the average of actual local parameters across all nodes without the necessity to decrypt the aggregated result separately. To facilitate this, we introduce a new multi-key homomorphic encryption technique tailored for secure aggregation in federated learning environments. Each node uses a different encryption key to encrypt its mask value. Importantly, the ciphertext of each mask includes a partial decryption component from the node, allowing the collective sum of encrypted masks to be automatically decrypted once all are aggregated. Consequently, the server computes the average of the actual local parameters by simply subtracting the decrypted total sum of mask values from the cumulative sum of the masked local parameters. Our approach effectively eliminates the need for interactions between nodes and the server for mask generation and sharing, while addressing the limitation of a single key homomorphic encryption. Moreover, the proposed aggregation process completes the global model update in just two interactions (in the absence of dropouts), significantly simplifying the aggregation procedure. Utilizing the CKKS (Cheon-Kim-Kim-Song) homomorphic encryption scheme, our method ensures efficient aggregation without compromising security or accuracy. We demonstrate the accuracy and efficiency of the proposed method through varied experiments on MNIST data.

Keywords: secure aggregation; federated learning; homomorphic encryption; multi-key homomorphic encryption

Academic Editor: Aryya Gangopadhyay

Received: 13 November 2024
Revised: 29 December 2024
Accepted: 30 December 2024
Published: 3 January 2025

Citation: Park, S.; Lee, J.; Harada, K.; Chi, J. Masking and Homomorphic Encryption-Combined Secure Aggregation for Privacy-Preserving Federated Learning. *Electronics* **2025**, *14*, 177. https://doi.org/10.3390/electronics14010177

1. Introduction

Federated learning [1] effectively protects the privacy of each user's local data by only requiring the transmission of local model parameters, not the local data itself. However, local data can still be deduced from these parameters using the inverse attack [2–4], necessitating secure transmission of local parameters to the federated learning server. To conceal the local model parameters, several techniques have been proposed, including

secure aggregation through secure multiparty computation [5,6], differential privacy [7,8], homomorphic encryption (HE) [9–21], and masking techniques [22–29]. Among these, mask-based aggregation and HE-based aggregation are the most widely adopted for ensuring privacy in federated learning.

The mask-based aggregation effectively conceals the actual local parameters by adding random mask values, which are removed post-aggregation, allowing only the computation of the sum of local updates. Although simple in computation, it requires an additional round for generating and sharing the mask values. The crucial aspect of this method is the efficient and secure generation of random masks that can be automatically removed after aggregation. To achieve this, each node must generate pairwise masks shared with all other nodes, necessitating an extra communication round for mask sharing. Moreover, to ensure robustness against dropouts, nodes must generate and share additional shares needed to reconstruct dropped masks.

HE-based aggregation allows nodes to encrypt their local parameters, with the federated learning server aggregating the updates in an encrypted form. Each node then retrieves the actual average of the parameters by decrypting the aggregated result. This method decreases the necessity for communication between nodes and the server for global model updates. However, HE computation is considerably more complex and computationally intensive than masking-based methods. Additionally, homomorphic operations are generally limited to ciphertexts encrypted using the same key. In federated learning, employing a single key for HE poses a security risk, as any malicious node or attacker who compromises a node could decrypt all ciphertexts produced by other nodes using the same key.

Multi-key homomorphic encryption (MKHE) [16] solves this problem by employing different keys for HE [15,16]. Homomorphic operations are possible on ciphertexts encrypted with different keys. However, decrypting these ciphertexts necessitates collaboration among all nodes. Each node contributes by generating a partial decryption using its own secret key, with the final decryption achieved by aggregating all these partial decryptions.

In this paper, we propose a novel and practical masking and HE-combined secure aggregation (MHESA) protocol that effectively integrates the advantages of both approaches and enables nodes to use different keys for HE. Each node's local parameters are simply masked with a randomly chosen value by the node. Only the random mask is encrypted using the node's own secret key, based on our proposed MKHE scheme. In each round of federated learning, each node transmits its masked parameters along with the homomorphically encrypted mask to the federated learning server. The server aggregates these updates by performing homomorphic additions on the ciphertexts, then obtains the sum of actual local parameters across all nodes without separately decrypting the aggregated result.

Compared to existing aggregation methods, our proposed method has several distinctions. First, our masking technique requires only one mask value per node, which is selected by the node itself, eliminating the need to create additional shared mask values. This approach removes any need for communication between nodes and the server to manage mask values.

Second, we introduce a new MKHE protocol based on CKKS (Cheon-Kim-Kim-Song) HE [21], specifically designed for secure aggregation in federated learning environments. In this protocol, each node creates its own private-public key pair while the server generates a group public key for all participating nodes. Since each mask is encrypted using an individual encryption key through our MKHE scheme, the ciphertext of each mask remains secure from all other participating nodes. Moreover, the individual keys of nodes are initially set in the setup phase of a new federated learning process and continue to be used until the termination of the entire process.

The essential feature of our model is that it does not require a separate decryption process to obtain the actual global update. The server calculates the total sum of the actual parameters through homomorphic addition of all ciphertexts provided by the nodes. In other words, decryption occurs automatically during homomorphic aggregation, overcoming the limitation of MKHE, which necessitates collaborative decryption from all nodes. To enable this, our proposed HE protocol produces a ciphertext that includes its partial decryption. Consequently, during the aggregation of the ciphertexts, the partial decryption components of individual ciphertexts are also aggregated simultaneously. As a result, the aggregated ciphertext is automatically decrypted after the aggregation. Importantly, not an individual ciphertext generated by each node but the total sum of ciphertexts is decrypted. Thus, the server can retrieve the actual sum of all original mask values by simply computing the sum of all ciphertexts. The server then computes the sum of the actual local parameters by subtracting the sum of mask values from the total sum of all masked parameters. In this process, the server determines only the sum of all mask values and cannot infer individual mask values from the ciphertexts. This information suffices for federated learning.

Thirdly, our protocol effectively addresses dropout scenarios without necessitating updates to individual keys. As previously mentioned, each ciphertext inherently includes its partial decryption. If a dropout occurs, the server cannot produce a correct aggregated result due to the absence of partial decryption components from the dropped nodes. Therefore, the remaining active nodes must regenerate their ciphertexts. In such instances, the server promptly generates a new group public key for the updated aggregation group and distributes it among the nodes in the new group. Subsequently, each node updates its local update with a new mask value and creates a new ciphertext of the mask value using its own key and the new group public key. Although nodes are required to regenerate and send their updated ciphertexts to the server, they do not need to alter or update individual encryption keys, even if the group of active nodes changes. This is a distinct advantage because, in our model, only the mask value is encrypted—not the local parameters, contrary to existing HE-based aggregation models that encrypt local parameters. In traditional models, active nodes must change their encryption keys to produce new ciphertexts for the same local parameters, necessitating additional computation and communication to update their keys. However, in our model, refreshing the mask value generates a new local update, eliminating the need for nodes to change their individual keys.

The main contributions of this paper can be summarized as follows:

- Hybrid Secure Aggregation Strategy: We introduce a novel and practical secure aggregation strategy that integrates the advantages of both masking-based and HE-based aggregation methods. Compared to traditional masking-based techniques that often require complex collaborative mask management, our method eliminates the operational overhead associated with mask coordination and employs a new MKHE technique to encrypt masks. The encrypted masks are efficiently removed from the aggregated result using a homomorphic additive operation. This eliminates the need for collaborative operations while ensuring robust privacy for local model parameters.
- Automatic Decryption and Efficient Global Model Update: The proposed MKHE technique allows the use of different keys and facilitates automatic decryption of the aggregated ciphertexts. Consequently, the server can directly retrieve the actual global model update without requiring an additional decryption process, a limitation seen in conventional MKHE models. Thus, our approach minimizes communication rounds, requiring only two interactions to complete a global model update under no-dropout conditions.
- Key Management Independence: In our model, nodes can manage their keys independently of the aggregation group's composition. In the existing MKHE-based

aggregation techniques, where local parameters are directly encrypted, nodes need to update their keys whenever the aggregation group changes. This introduces considerable computational and operational overhead due to the need for the collaborative key updates among the newly formed group. In contrast, our method ensures that each node's keys remain unchanged throughout the entire federated learning process. This is because only the mask value, not the local parameters, is encrypted, and individual mask values are independently refreshed by each node in every round of federated learning. Even in the presence of dropouts, nodes can perform the re-aggregation using their existing keys, eliminating the need for additional key update protocols.

We provide an overview of related works in Section 2, detail the protocols of our proposed model in Section 3, explore the correctness and security of our model in Section 4, analyze simulation performance regarding accuracy and efficiency, and conclude the paper in Section 5.

2. Related Works

We provide a brief overview of recent advancements in secure aggregation methods aimed at preserving privacy in federated learning, which are highly pertinent to our research.

The masking-based aggregation approach employs pairwise masks to conceal local parameters from the server. K. Bonawitz et al. [22,23] introduced an additive masking secure aggregation technique for federated learning, wherein users obscure their local updates with paired perturbations, removed during aggregation, allowing the server access only to the sum of all local updates. They further enhanced this method [23] to tolerate user dropouts by integrating Shamir's secret sharing scheme. J. So et al. [28] developed turbo aggregation, performing circular aggregation across multiple groups and reducing communication overhead by limiting mask and data sharing to group members instead of all nodes. Additionally, J. Kim et al. [29] proposed a group-based aggregation strategy clustering nodes by similar response times based on their local processing time and locations, utilizing an additive masking technique to effectively address dropout situations without relying on Shamir's secret sharing scheme and providing public verification of mask value integrity.

Y. Aono et al. introduced a deep learning algorithm [13] that uses HE to encrypt local model parameters, protecting the privacy of both local and global model parameters through homomorphic operations on ciphertexts, though it requires all participants to use the same private key for HE. H. Fang and Q. Qian proposed a HE-based federated learning strategy [14], also requiring a shared encryption key among all participants. In contrast, J. Park and H. Lim developed a federated learning model using HE [15], allowing participants to encrypt their local model parameters with individual private keys, enabling the server to update global model parameters using these variously encrypted local parameters within a distributed cryptosystem. However, this approach requires a third-party computation provider alongside the cloud server, necessitating collaboration between both parties to decrypt the encrypted local parameters.

W. Liu et al. introduced a round-efficient federated learning model [16] using multi-key fully homomorphic encryption (MKFHE), which enables computations on data encrypted across different parties and reduces the interactions required per federated learning round. Furthermore, nodes can dynamically join the homomorphic computation at any time by generating their own refreshing keys with the proposed multi-hop MKFHE. The refresh key converts old ciphertexts into new ciphertexts for newly formed node groups. In this process, each node must generate partial refresh keys for all others, and the server aggregates these into a single refresh key for each node. To minimize the size of encrypted local parameters, W. Jin et al. proposed a selective parameter encryption method [17] using HE in federated

learning. This method selectively encrypts the most privacy-sensitive parameters using a HE key, although it still necessitates all nodes to use the same HE key.

Beyond these methods, several case studies on HE-based federated learning have been conducted. For instance, F. Wibawa et al. [18] developed a privacy-preserving method that integrates federated learning with HE to train convolutional neural network (CNN) models for COVID-19 detection. Local training takes place on lung X-ray images gathered from each hospital, and the model updates are transmitted and aggregated on the server in a homomorphically encrypted state. N. M. Hijazi et al. [19] proposed secure federated learning using fully HE in an Internet of Things (IoT) environment, while S. P. Sanon et al. [20] suggested a secure federated learning approach for training network traffic prediction models using encrypted network traffic. These implementations reveal that federated learning models incorporating HE can effectively and practically safeguard the privacy of local data in real-world applications.

3. MHESA: Masking and Homomorphic Encryption-Combined Secure Aggregation

The proposed model is built on the CKKS's RLWE-based HE scheme (denoted as *CKKS-HE* in the following paper) [21]. Before describing our MHESA-based federated learning model, we first briefly describe the federated learning architecture used in this paper and the *CKKS-HE* scheme. We then present an overview of the proposed model and provide a detailed protocol.

3.1. Background and System Overview

3.1.1. Federated Learning Model

The federated learning system consists of a single central federated learning server (denoted as *FS* in the following paper) and N (mobile) users (or nodes). Throughout the rest of the paper, *FS* and U refer to the federated learning server and a subset of nodes, respectively, where each node is represented as u_i.

FS trains a global model $w \in \mathbb{R}^d$ with dimension d using data stored on mobile devices. The goal of this training process is to minimize a global objective function $F(w)$,

$$\arg\min_{w} F(w) \text{ where } F(w) = \sum_{i=1}^{N} \frac{x_i}{x} F_i(w). \tag{1}$$

Here, F_i is the local objective function of u_i, x_i represents the private data size of u_i, and $x = \sum_i x_i$. The local objective function $F_i(w)$ for the global model w is defined as

$$F_i(w) = \frac{1}{x_i} \sum_{j=1}^{x_i} f_i(w) \text{ where } f_i(w) = l(X_i, Y_i; w). \tag{2}$$

$f_i(w)$ represents the loss of the prediction on example (X_i, Y_i) made with model parameters w.

For a fixed learning rate η, *FS* trains the global model by iteratively performing distributed stochastic gradient descent (SGD) using the currently available mobile nodes. At iteration t, the server distributes the current global algorithm state (e.g., the current model parameters), w^t, to the mobile nodes. Each u_i then computes $\nabla F_i(w^t)$, which is the average gradient on its local data using the current model w^t, and generates its local update w_i^{t+1},

$$w_i^{t+1} := w^t - \eta \nabla F_i(w^t). \tag{3}$$

u_i iterates the local update several times before transmitting the update to *FS*. Subsequently, *FS* combines these gradients and updates the global model for the subsequent iteration,

$$\begin{aligned} w^{t+1} &:= \sum_{i=1}^{N} \frac{x_i}{x} w_i^{t+1} \\ &= w^t - \eta \sum_{i=1}^{N} \frac{x_i}{x} \nabla F_i(w^t) = w^t - \eta \nabla F(w^t), \end{aligned} \tag{4}$$

Since the loss gradient $\nabla F(w^t)$ can be expressed as a weighted average across nodes, $\nabla F(w^t) = \sum_{i=1}^{N} \frac{x_i}{x} \nabla F_i(w^t)$.

3.1.2. RLWE-Based HE Scheme

CKKS-HE leverages modular arithmetic and noise management as its core technique to enable secure computation on encrypted data. It represents data in ciphertexts that are defined over polynomial rings modulo a large integers. This ensures that encrypted computations remain within the bound on the encrypted system, allowing for secure and efficient addition and multiplication of encrypted values. Every operation on encrypted data introduces a small amount of noise into the ciphertext. *CKKS-HE* controls the growth of this noise by reducing the ciphertext's modulus with its advanced noise management techniques such as rescaling and modulus switching. This makes the accumulated noise remain manageable and preserves the integrity and decryptability of the encrypted data. The detailed protocols for encryption and decryption are given follows. In this paper, since re-encryption and homomorphic multiplication operations are not required, descriptions of the related protocols such as evaluation key generation, rescaling, and modulus switching are omitted. *CKKS-HE* is based on the ring learning with errors (RLWE) assumption. For a modulus q and a base $p > 0$, let $q_l = p^l \cdot q$ for a level $0 < l \leq L$. For a positive integer M, $\Phi(M)$ is the M-th cyclotomic polynomial of degree $n = \phi(M)$. $R = Z[X]/(X^n + 1)$ is a power-of-two degree cyclotomic ring, and $R_q = Z_q[X]/(X^n + 1)$ be the residue ring of R modulo q. A polynomial $A(x) \in R_{q_L}$ is defined as $A(x) = \sum_{0 \leq i < n} a_i \cdot X^i$ with the vector of its coefficients $(a_0, \ldots, a_{n-1})$ in $Z_{q_L}^n$. The coefficient vector $(a_0, \ldots, a_{n-1})$ is denoted as A. For a real $\sigma > 0$, $D(\sigma^2)$ represents a distribution over R, sampling n coefficients independently from the discrete Gaussian distribution with variance σ^2. For a positive integer h, $HWT(h)$ is the set of signed binary vectors in $\{-1,\ 0,\ 1\}^n$ whose Hamming weight is exactly h. For a real $0 \leq \rho \leq 1$, the distribution $ZO(\rho)$ draws each entry in the vector from $\{-1,\ 0,\ 1\}^n$, assigning a probability of $\rho/2$ for both -1 and +1, and a probability of $1 - \rho$ for 0.

Given the parameters (n, h, q, p, σ), *CKKS-HE* employs five key algorithms: *KeyGen*, *Ecd*, *Dcd*, *Enc* and *Dec*.

- *KeyGen*(n, h, q, p, σ) samples $s \leftarrow HWT(h)$, $A \leftarrow R_{q_L}$ and $e \leftarrow D(\sigma^2)$. It sets the secret key as $(1, s)$ and the public key pk as (B, A) in $R_{q_L}^2$ where $B = -A \cdot s + e \pmod{q_L}$.
- *Ecd*$(z; \Delta)$ generates an integral plaintext polynomial $m(X)$ for a given $(n/2)$-dimensional vector $z = (z_j)_{j \in T} \in Z[i]^{n/2}$. It calculates $m(X) = \mu^{-1}\left(\left\lfloor \Delta \cdot \pi^{-1}(z) \right\rceil_{\mu(R)}\right) \in R$, where $\Delta \geq 1$ represents a scaling factor, and π is a natural projection defined by $(z_j)_{j \in Z_M^*} \mapsto (z_j)_{j \in T}$ for a multiplicative subgroup T of Z_M^* satisfying $Z_M^*/T = \{\pm 1\}$. μ is a canonical embedding map from integral polynomial to elements in the complex field $\mathbb{C}^n$. It computes a polynomial whose evaluation values at the roots, the complex primitive roots of unity in the extension field $\mathbb{C}$, correspond to the given vector of complex numbers.
- *Dcd*$(m; \Delta)$ returns the vector $z = \pi \circ \mu(\Delta^{-1} \cdot m)$ for an input polynomial m in R, i.e., $z_j = \Delta^{-1} \cdot m\left(\varsigma_M^j\right)$ for $j \in T$.
- *Enc*(m, pk), for a public key pk and a plaintext polynomial m, samples $v \leftarrow ZO(0.5)$, and $e_0, e_1 \leftarrow D(\sigma^2)$, and outputs a ciphertext $c = (c_0, c_1)$, where $c_0 = v \cdot A + e_0 \pmod{q_L}$ and $c_1 = v \cdot B + m + e_1 \pmod{q_L}$.

- *Dec*(c, sk), for a ciphertext $c = (c_0, c_1)$ and a secret key sk, outputs $c_1 + c_0 \cdot s \bmod q_l$.

3.1.3. System Overview

We briefly outline our proposed federated learning model and detail the protocol in the subsequent section. The most notable features of the proposed model include (1) each node possesses its individual private and public key pair; (2) each node creates a masked local update, with only the mask value encrypted using a unique encryption key; and (3) the *FS* can directly derive a global model update by simply aggregating all local updates, without requiring further decryption.

To facilitate this, we propose a new MKHE scheme based on *CKKS-HE*, tailored to our federated learning model. We briefly describe the proposed MKHK scheme. In addition to the system parameters (n, h, q, p, σ) used in *CKKS-HE*, a public ring element $A \leftarrow R_{q_L}$ and $Q = q^2$ are additionally set. Given the parameters (n, h, q, p, σ, A, Q), the proposed MKHE scheme consists of six algorithms: *KeyGen*, *GroupKeyGen*, *Ecd*, *Dcd*, *Enc* and *Dec*. Here, *Ecd* and *Dcd* are identical to the algorithms of *CKKS-HE*, so the description of those algorithms is omitted.

- *KeyGen*(n, h, q, p, σ, A, Q) generates the public-private key pair <pk_i, sk_i> and the commitment c_i of each node u_i. It samples $s_i \leftarrow HWT(h)$, $v_i \leftarrow ZO(0.5)$ and $e_i, e_{0i} \leftarrow D(\sigma^2)$. It sets the secret key as $sk_i = (1, s_i)$ and public key $pk_i = -A \cdot sk_i + e_i \pmod Q$. Then, it sets a commitment c_i for v_i such as $c_i = A \cdot v_i + e_{0i} \pmod Q$.
- *GroupKeyGen*(PK, C, U_T) generates a group public key PK_T and a group commitment C_T for a given node set $U_T \subseteq U$, where $PK = \{pk_i\}$ and $C = \{c_i\}$ for all u_i in U. It sets PK_T as $\sum_{u_j \in U_T} pk_j$ and C_T as $\sum_{u_j \in U_T} c_j$.
- *Enc*(m_i, PK_T, C_T, sk_i) outputs a ciphertext E_i for a plaintext polynomial m_i. For given group public key PK_T and group commitment C_T, it samples $e_{1i} \leftarrow D(\sigma^2)$, and outputs a ciphertext $E_i = v_i \cdot PK_T + m_i + e_{1i} + C_T \cdot sk_i \pmod Q$.
- *Dec*(E_T, U_T) adds all ciphertexts E_i in E_T, where $E_T = \{E_i\}$ for all u_i in U_T. If ciphertexts obtained from all nodes in U_T are added, it outputs the sum of plaintext polynomials generated by all nodes in U_T such as $\sum(m_i + e')$.

The primary difference between the proposed protocol and the original *CKKS-HE* lies in its objective: the focus is not on decrypting individual ciphertexts but on decrypting the sum of ciphertexts. Therefore, instead of using individual public keys during encryption, a group public key and commitment—composed of the public keys and commitments of all participating nodes in the aggregation—are utilized. Additionally, to enable automatic decryption of the aggregated ciphertext sum without a separate decryption process, a partial decryption using individual secret keys is embedded into the ciphertext. Since the ciphertexts are encrypted with the group public key, individual ciphertexts cannot be decrypted unless the corresponding partial decryptions from all secret keys matching the group public key for individual plaintexts are combined. This approach ensures the confidentiality of individual ciphertexts while allowing the decryption of the aggregated ciphertext sum, thereby guaranteeing secure aggregation in federated learning environments.

Our federated learning model comprises two main phases: *Setup* and *Learning*.

Setup: This initial phase occurs when a new federated learning session starts. Its primary task is to generate system parameters and all necessary keys for our MKHE together with all nodes participating in the federated learning.

Learning: *FS* and nodes repeat this phase until the entire federated learning terminates. The *Learning* phase consists of two sub steps: *initiation* and *aggregation*. At the t-th (>0) round, in the initiation step, *FS* determines a set of nodes participating in the t-th round of *Learning* and generates a group public key for the nodes. Once the initiation is complete, the aggregation step begins. Each node u_i updates its local model parameters w_i and utilizes w_i

to generate a local update D_i masked with a random secret M_i, and concurrently produces an encryption E_i of M_i. Node u_i then transmits the tuple <D_i, E_i> to *FS*. *FS* collects these updates and modifies the global model w by summing all local model parameters. In case any data is missing during aggregation, or a local update does not reach *FS* within the set aggregation period, the *Learning* phase is repeated using only the data from available nodes. Once the global model update is finalized, it is distributed to all nodes, and the *Learning* process repeats based on the updated global model.

We assume that nodes communicate solely with *FS*, and that both *FS* and the nodes operate under an 'honest-but-curious' model. Although they strictly follow the protocol, they remain continuously interested in extracting meaningful data from the interaction. In this threat model, the proposed MHESA satisfies specific security requirements:

(1) Privacy of local datasets and model parameters: All data stored on each node's local device and the local model parameters transmitted over the network must remain confidential, shielded not only from other nodes but also from *FS*. *FS* only has access to the aggregated sum of all local updates provided by the nodes.
(2) Robustness to dropouts: In scenarios where data transmission is disrupted due to network issues or device malfunctions, *FS* must still be able to accurately compute a global model update.

3.2. MHESA Protocol

In this section, we elaborate on the specific details of the MHESA protocol. At the start of a new phase, *FS* initiates the *Setup* phase with all nodes in U.

Setup: FS collaboratively establishes system parameters and keys with each node u_i as follows:

1. *FS* sets n, h, q and σ as described in Section 3.1.2 and generates $Q = q^2$ and a public ring element $A \leftarrow R_Q$ with the vector of coefficients $[a_0, \ldots, a_{n-1}]$ in Z_Q^n. *FS* publishes <n, h, q, σ, A, Q> to all nodes.
2. Each $u_i \in U$ generates its key pair <pk_i, sk_i> and a commitment c_i by *KeyGen*(n, h, q, σ, A, Q), where $sk_i = (1, s_i)$, $pk_i = -A \cdot sk_i + e_i \pmod{Q}$ and $c_i = A \cdot v_i + e_{0i} \pmod{Q}$. Then, u_i responds to *FS* with <pk_i, c_i, x_i>, where x_i represents the size of u_i's local dataset.
3. *FS* sets $PK = \{pk_i\}$, $C = \{c_i\}$ and $X = \{x_i\}$ for all u_i in U.

Once *Setup* is completed, *FS* and nodes repeat the subsequent *Learning* phases until the federated learning process is concluded.

Learning: FS updates the global model by aggregating local model parameters from all available nodes, and nodes update their local models with the updated global model parameters. At the t-th > 0 iteration,

[Initiation]

1. *FS* sends a start message to all nodes.
2. All available nodes respond to *FS* with their x_is, where x_i represents the size of u_i's local dataset.
3. *FS* sets U_T as the t-th node group for all replied nodes and generates the t-th group parameters PK_T and C_T for all nodes in U_T by *GroupKeyGen*(PK, C, U_T), where $PK_T = \sum_{u_i \in U_T} pk_i$ and $C_T = \sum_{u_i \in U_T} c_i$. It also sets the total size of datasets as $X_T = \sum_{u_i \in U_T} x_i$. Then, it broadcasts <$PK_T$, C_T, X_T> to all nodes in U_T.

[Aggregation]

4. For each u_i in U_T, let w_i^t represent a set of local model parameters of u_i at the t-th iteration. u_i selects a random real number $M_i^t \in \mathbb{R}$ and generates a masked local update D_i^t according to the following Equation (5):

$$D_i^t = \frac{x_i}{X_T} \cdot w_i^t + M_i^t \tag{5}$$

Next, u_i generates a plaintext polynomial $m_i^t(x) = Ecd\left(M_i^t; \Delta\right)$ and calculates the encryption of m_i^t, $E_i^t = Enc\left(m_i^t,\ PK_T,\ C_T,\ sk_i\right)$, as shown in Equation (6).

$$E_i^t = v_i \cdot PK_T + m_i^t + e_{1i} + C_T \cdot sk_i \pmod{Q} \tag{6}$$

Note that E_i^t includes both the encryption of m_i^t using the group public key PK_T and the partial decryption by u_i. u_i sends <D_i^t, E_i^t> to *FS*.

5. *FS* calculates $D_T^t = \sum_{u_i \in U_T} D_i^t$ and $E_T^t = \sum_{u_i \in U_T} E_i^t \pmod{q}$. Here, $E_T^t = Dec\left(\left\{E_i^t\right\}, U_T\right) = \sum\left(m_i^t + e'\right)$. *FS* computes $E_T^{t\,\prime} = Dcd\left(E_T^t; \Delta\right) = \sum M_i^t$. Finally, *FS* updates the t-th global update w^t with the average of all local updates as described in Equation (7):

$$w^t = D_T^t - E_T^{t\,\prime} = \sum_{u_i \in U_T} \frac{x_i}{X_T} w_i^t \tag{7}$$

FS distributes w^t to all nodes in U.

6. Each u_i updates its local model w_i^{t+1} with w^t.

In our protocol, only homomorphic addition is required to aggregate all local updates, so generation of an evaluation key used in *CKKS-HE* is unnecessary. Furthermore, once the aggregation is complete, nodes refresh their masks and generate new ciphertexts for them. Therefore, no process involving rescaling or leveled homomorphic encryption required by *CKKS-HE* is needed. For simplicity, we used only two moduli Q and q for encryption and homomorphic addition.

3.3. Dropout Management

The existing aggregation protocol does not account for dropout situations. However, due to environmental network factors, the local updates from some nodes might not reach *FS* promptly, yet the protocol must remain robust in handling dropout nodes. Since *FS* must repeatedly perform the *Learning* phase, it cannot wait indefinitely for all local updates to arrive in each round. To address this, a predefined waiting time is set for receiving local parameters. If any local update fails to be transmitted within this timeframe, the corresponding node is considered a dropout node. This situation can arise in two scenarios: (1) the transmission of the local update may be interrupted due to network issues, or (2) the update is delayed and arrives after the predetermined waiting time. In the second scenario, *FS* can still receive the local update from a node it previously classified as a dropout. However, even if the local update from the dropout node eventually reaches *FS*, it is impossible for *FS* to determine the mask value or the actual local parameters from that update. Yet, complications arise when there is only a single dropout node and the global model parameters are updated using the remaining nodes, excluding the dropout. Let U denote the set of all nodes joined at the t-th round of the *Learning* phase, and let U_D represent the set of nodes identified as dropouts. Then, U_A represents the set of available nodes, defined as $U - U_D$. Suppose that $U_D = \{u_d\}$ (i.e., a single dropout node) and *FS* updated the t-th global model parameter w_A using only the nodes in U_A. Consider that D_d and E_d, generated by u_d, are subsequently received by *FS*. In this case, *FS* can compute an additional global update w_U, incorporating all nodes in U, including u_d, by adding D_d and E_d to the parameters D_i and E_i previously provided by nodes in U_A. Using this additional

update, FS could then derive u_d's local parameter w_d through the equation $w_d = w_U - w_A$. This scenario occurs when only a single dropout node exists. To prevent such leakage of local information, it is necessary to ensure that at least two sets of local parameters have to be hidden from FS, whenever dropouts occur.

Therefore, to effectively manage dropout nodes, we consider two situations: (1) when two or more dropout nodes occur, and (2) when only one dropout node occurs. In the second case, an additional node must be excluded from the aggregation. Since the local update of the excluded node is not reflected in the global update, the accuracy of the global model may be compromised. To minimize this impact, FS randomly selects a node to exclude from a group of nodes with relatively smaller local datasets. As will be explained in Section 4.2, our experiments results demonstrate that the overall accuracy of the federated learning model is mainly affected by the accuracy of individual local nodes. Furthermore, the accuracy of each local node is found to be highly sensitive to the size of its local dataset. Thus, FS sets U_A by excluding one more node u_j with relatively small x_j (the size of dataset) from the actual U_A. Then, FS updates PK_A, C_A and X_A for U_A and share these public parameters with nodes in U_A. Then, each u_j in U_A selects a new $M_j{}'$ and generates a new pair of $D_j{}'$ and $E_j{}'$ using $M_j{}'$, PK_A, C_A and X_A. With these updates, FS can compute the global update w_A for U_A as follows:

$$\begin{gathered} w_A = \left(D_A - E_A{}'\right) = \sum_{u_i \in U_A} \frac{x_i}{X_A} w_i, \\ \text{where } D_A = \sum_{u_i \in U_A} D_i{}', \; E_A = \sum_{u_i \in U_A} E_i{}' \text{and } E'_A = Dcd(E_A; \, \Delta). \end{gathered} \tag{8}$$

If some data from nodes in U_D are never delivered to FS, meaning they are completely lost, then the global update w_U for all nodes in U cannot be computed at all, and the previously delivered local updates from all nodes remain secure. However, on rare occasions, if all data deemed as dropouts are later delivered to FS, FS can compute both w_A for U_A and w_U for U. In this situation, in the first scenario, FS remains unable to discern the individual local update of any node u_d in U_D, but FS can calculate $w_D = w_U - w_A$, which represents the cumulative sum of w_d from all u_d in U_D. Even in a single dropout situation, U_A is modified to ensure that there are at least two dropout nodes, so that w_d of each dropout node remains secure.

4. Analysis

4.1. Correctness and Security

In this section, we first demonstrate the correctness of the proposed scheme and subsequently analyze the privacy of local model parameters. For the protocol to function properly, the federated learning server must calculate the sum of original mask values from the encrypted mask values. Consequently, we establish that the homomorphic sum E_U^t of all encrypted masks E_i^t accurately reveals the sum of all original m_i^t.

For $PK_U = \sum_{u_i \in U} pk_i$, let V_U be $\sum_{u_i \in U} v_i$ and SK_U be $\sum_{u_i \in U} sk_i$. Intuitively, PK_U becomes the group public key generated from the public keys of all nodes, and SK_U is the corresponding group secret key for PK_U, while V_U is an aggregated group random vector for encryption. With these parameters, we derive the sum of all m_i^t from E_U^t as follows:

$$
\begin{aligned}
1:\quad & E_U^t = \textstyle\sum_{u_i \in U} E_i^t && (\mathrm{mod}\ q)\\
2:\quad & = \textstyle\sum_{u_i \in U} v_i \cdot PK_U + m_i^t + e_{1i} + C_U \cdot sk_i && (\mathrm{mod}\ q)\\
3:\quad & = PK_U \cdot \textstyle\sum v_i + \sum m_i^t + \sum e_{1i} + C_U \cdot \sum sk_i && (\mathrm{mod}\ q)\\
4:\quad & = PK_U \cdot V_U + \textstyle\sum m_i^t + \sum e_{1i} + C_U \cdot SK_U && (\mathrm{mod}\ q)\\
5:\quad & = \textstyle\sum pk_i \cdot V_U + \sum m_i^t + \sum e_{1i} + \sum c_i \cdot SK_U && (\mathrm{mod}\ q)\\
6:\quad & = \textstyle\sum(-A \cdot sk_i + e_i) \cdot V_U + \sum m_i^t + \sum e_{1i} + \sum (A \cdot v_i + e_{0i}) \cdot SK_U && (\mathrm{mod}\ q)\\
7:\quad & = -A \cdot \textstyle\sum sk_i \cdot V_U + \sum e_i \cdot V_U + \sum m_i^t + \sum e_{1i} + A \cdot \sum v_i \cdot SK_U + \sum e_{0i} \cdot SK_U && (\mathrm{mod}\ q)\\
8:\quad & = -A \cdot SK_U \cdot V_U + \textstyle\sum e_i \cdot V_U + \sum m_i^t + \sum e_{1i} + A \cdot V_U \cdot SK_U + \sum e_{0i} \cdot SK_U && (\mathrm{mod}\ q)\\
9:\quad & = \textstyle\sum m_i^t + \sum e_i \cdot V_U + \sum e_{1i} + \sum e_{0i} \cdot SK_U && (\mathrm{mod}\ q)\\
10:\quad & = \textstyle\sum \left(m_i^t + e_i \cdot V_U + e_{1i} + e_{0ii} \cdot SK_U\right) && (\mathrm{mod}\ q)\\
11:\quad & = \textstyle\sum Dec(C_i, SK_u) \text{ where } C_i = Enc\left(m_i^t, PK_U\right) \text{ in } CKKS\text{-}HE && (\mathrm{mod}\ q)
\end{aligned}
$$

Decryption involves removing the public ring vector A from the ciphertext, resulting in a plaintext in the form of $m + e$. By summing all ciphertexts, as demonstrated in the equation at line 4, $PK_U \cdot V_U$ and $C_U \cdot SK_U$ are computed. These are equivalent to $\sum(-A \cdot sk_i + e_i) \cdot V_U$ and $\sum (A \cdot v_i + e_{0i}) \cdot SK_U$ as shown in the equation at line 6, and then derived to $-A \cdot SK_U \cdot V_U$ and $A \cdot V_U \cdot SK_U$, respectively, as indicated in the equation at line 8. Consequently, the public ring vector is eliminated, leaving the sum of all plaintexts containing the error terms. Finally, as shown in the equation at line 10, the sum of E_i^t equals the sum of the values decrypted using CKKS' decryption algorithm $Dec(C_i,\ SK_u)$ for the ciphertext $C_i = Enc(\ m_i^t,\ PK_U)$, which was encrypted by CKKS' encryption algorithm for each node's plaintext m_i^t. Finally, it accurately computes the sum of all original M_i^t through the CKKS's decoding operation $Dcd\left(E_U^t;\ \Delta\right) = \sum M_i^t$.

We demonstrate that our protocol maintains the privacy of local model parameters.

Theorem 1. *The proposed MHESA-based federated learning model preserves the privacy of local model parameters of nodes, if and only if the total number of nodes participating in the Learning phase is 4 or more, and the number of currently available (or active) nodes is 3 or more when dropouts occurred, in each iteration of Learning phase.*

Proof. In our model, the number of active nodes participating in the *Learning* phase must be at least 2. For a given node set U, the group public key PK_U for U is generated using the public keys of all nodes in U. When ciphertexts from all nodes in U are summed, the partial decryption parts of the ciphertexts are also summed and the decryption with the group private key SK_U associated with PK_U is completed. Thus, the summed ciphertext produces the cumulative sum of actual mask values contributed by all nodes in U.

If $|U| = 1$, the group public-private key pair <PK_U, SK_U> is identical to the public-private key pair <*pk*, *sk*> of the sole node in U. In this case, a ciphertext containing its decryption with *sk* directly reveals the plaintext itself. Thus, the proposed encryption scheme is unsuitable for a single node scenario. Conversely, when $|U| \geq 2$, the decrypted value corresponds to the sum of plaintexts from all participating nodes in U. Because only the sum, and not individual plaintexts, is computed, the privacy of each node's local update is preserved. Therefore, the model inherently requires a minimum of 2 active nodes to operate securely.

In dropout scenarios, as discussed in Section 3.3, the model assumes the presence of at least two dropout nodes when a dropout occurs. This may include one active node treated as a dropout. Consequently, at least 3 active nodes are required when a single dropout happens. For scenarios involving 2 or more dropout nodes, at least 2 active nodes are necessary to maintain the protocol. Therefore, the proposed model requires at least 4 nodes, including at least 3 active nodes, to ensure privacy-preserving federated learning.

Next, we prove that neither the federated learning server nor any other nodes can infer the original mask value from the local updates. As our proposed MKHE is based on *CKKS-HE*, the security of MKHE depends on the robustness of *CKKS-HE*. The distinction lies in the ciphertext of our MKHE, which includes a partial decryption. Hence, we assess whether this alteration could risk revealing the original mask value. Initially, we prove that it is impossible to ascertain m_i^t from its corresponding ciphertext E_i^t without the secret values v_j and sk_j from another node u_j.

The equation $E_i^t = v_i \cdot PK_U + m_i^t + e_{1i} + C_U \cdot sk_i$ consists of two components: a partial encryption of m_i^t and a partial decryption for m_i^t. Specifically, the portion of $v_i \cdot PK_U + m_i^t + e_{1i}$ is u_i's partial encryption of m_i^t using the group public key PK_U and u_i's random secret v_i, while the section of $C_U \cdot sk_i$ is u_i's partial decryption for m_i^t with its secret key sk_i. Because m_i^t is encrypted with PK_U, it can only be decrypted by the respective group secret key SK_U. However, E_i^t includes only u_i's partial decryption $C_U \cdot sk_i$. To derive m_i^t from E_i^t, the server must calculate $C_U \cdot SK_U$ and $v_i \cdot PK_U$, requiring knowledge of u_i's v_i and $C_U \cdot sk_j$ for all other nodes u_j. Yet, without knowing sk_j and v_j of u_j, it is infeasible to compute $C_U \cdot sk_j$ and $v_j \cdot PK_U$, making it impossible to determine m_i^t from E_i^t.

Second, we demonstrate that it is also impractical to identify individual m_i^t from all ciphertexts E_i^t. In the preceding proof, we established that determining m_i^t, $v_i \cdot PK_U$, and $C_U \cdot sk_i$ from each E_i^t is impossible without knowledge of sk_i and v_i of u_i. Once E_j^t of all u_j are delivered to the server, it can perform homomorphic operations on these ciphertexts. As demonstrated in the correctness proof, by aggregating all E_j^t, $V_U \cdot PK_U$, and $C_U \cdot SK_U$, the result is computed from the aggregated ciphertext, enabling decryption and revealing the total sum of the plaintexts m_j^t from all nodes u_j. Here, the decrypted value is not an individual m_j^t but the sum of all m_j^t. The server cannot identify either $V_U \cdot PK_U$ or $C_U \cdot SK_U$ from the aggregated ciphertext because these values are automatically obscured after aggregation. In our protocol, each plaintext m_i is encrypted and decrypted using u_i's secret values v_i and sk_i, and the public group key is the resultant sum of these values from all nodes. Consequently, the server cannot decrypt a message m_i for u_i unless all other nodes u_j generate E_j for m_i using their v_j and sk_j and provide these to the server. Thus, the proposed protocol safeguards the privacy of local model parameters, as it is impractical to extract individual m_i^t from the provided updates under the honest but curious threat model. □

4.2. Simulated Performance

In this section, we analyze the accuracy and efficiency of the proposed aggregation model using the MNIST [30] database. We compare the accuracy of the proposed aggregation model to that of a single centralized learning model. Additionally, we evaluate the accuracy of the proposed federated learning model by examining the number of nodes under two scenarios: IID and non-IID data distributions. To demonstrate the efficiency of the proposed model, we examine the computation time each node takes to generate its local update, the computation time the server requires to aggregate all local updates, and the size of data communicated between the server and the client. Specifically, we aimed to demonstrate the increase in computation and communication overhead at each local node and the corresponding communication delay due to the use of homomorphic encryption.

4.2.1. Experimental Environment

We first describe our experimental environment in detail. Two systems were used to implement the server and clients. The server system is equipped with an Intel(R) Core(TM) i7-12700K CPU, Intel(R) UHD Graphics 770 GPU, NVIDIA GeForce RTX 3080 GPU, and 32 GB of RAM. The client system is configured with an Intel(R) Core(TM) i9-7920X CPU, two

NVIDIA GeForce GTX 1080Ti GPUs, and 32 GB of RAM. Multiple nodes were implemented by creating multiple threads on the client system, ensuring that all nodes have the same computing power and communication environment. Moreover, both systems are on the same network with a bandwidth of 1 Gbps, meaning that actual communication time, including latency, was not considered in the experiments.

For individual training of each node, a two-layer CNN model with 5×5 convolution layers (the first with sixteen channels, the second with thirty-two channels, each followed with 2×2 max pooling), ReLU activation, and a final softmax output layer was used. Local learning at the client side was implemented using the PyTorch framework while the federated learning at both sides was implemented in C++ with CKKS open codes. The MNIST database consists of a training set of 60,000 images along with a test set of 10,000 images. The data is distributed in two ways: IID and non-IID distributions, depending on the number of nodes participating in the federated learning. In the IID distribution, data is distributed evenly among the nodes, whereas in the non-IID distribution, the size of data allocated to each node varies. This leads to differences in execution times and, ultimately, varying response times to *FS*. This setup effectively simulates heterogeneous nodes with diverse computing environments. Table 1 shows the size of data allocated to nodes according to the number of nodes in both distributions.

Table 1. The size of data allocated to nodes.

Number of Nodes	Data Size per Node		
	IID Distribution	Non-IID Distribution	
		Minimum	Maximum
1	60,000	60,000	
10	6000	500	12,700
20	3000	400	4800
50	1200	100	2200
100	600	100	1200

To ensure 128 bits of security for *CKKS-HE*, we used a 16-bit n as the degree of the ring polynomial, an 800-bit q as the modulus for decryption, and a 1600-bit Q as the modulus for encryption. The server and clients completed 100 rounds of *Learning* phase (the global model updating phase) to finalize a new federated learning process. Additionally, we repeated the experiment 10 times to obtain an average result. Table 2 summarizes the experimental parameters and values used in our experiments.

Table 2. Experimental parameters and values.

Parameters	Values
The number of nodes	1, 10, 20, 50, 100
The number of rounds for federated learning	100
The number of weight parameters	21,840
The size of n as the degree of ring polynomial (bits)	16
The size of q as the modulus for decryption (bits)	800
Size of Q as the modulus for encryption (bits)	1600

4.2.2. Experimental Results

We first demonstrate the accuracy of our proposed model by analyzing the aggregated results using our MHESA compared to raw data aggregation. And then, we assess how

differences in accuracy relate to the number of nodes and data distribution methods. Table 3 compares the accuracy of our model with that of raw data aggregation by round. In this experiment, there are 10 nodes, and data is evenly distributed, with each node using 6000 images for local learning. We obtained the average accuracy of the two models through 100 repeated experiments.

Table 3. The average accuracy of the proposed model versus raw data aggregation.

Round	Accuracy of the Proposed Model (N = 10, IID) (%)	Accuracy of Raw Data Aggregation (%)
10	91.28 (0.10) *	91.27 (0.11) *
20	94.33 (0.08) *	94.32 (0.07) *
40	96.36 (0.05) *	96.36 (0.05) *
60	97.08 (0.05) *	97.09 (0.05) *
80	97.42 (0.05) *	97.43 (0.04) *
100	97.75 (0.04) *	97.74 (0.05) *

* The values in parentheses represent the standard deviation.

As Table 3 shows, the accuracy difference is very slight, approximately 0.02%, with multiple rounds where our model demonstrates higher accuracy. This variance is due to the random assignment of data to each node in every experiment, rather than the aggregation method itself. Over 100 rounds, the accuracy for both methods averages around 97.7%, indicating negligible accuracy loss due to our aggregation approach using masking and HE. Table 4 shows the average accuracy of our proposed model based on the number of nodes under both IID and non-IID distributions. N indicates the number of nodes.

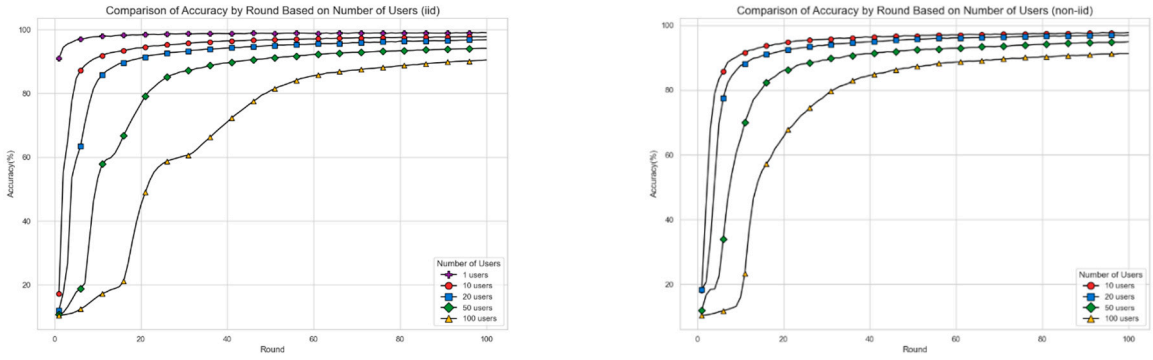
Table 4. The average accuracy of the proposed model according to the number of nodes.

	Average Accuracy (%)								
N	N = 1	N = 10		N = 20		N = 50		N = 100	
Round		IID	Non-IID	IID	Non-IID	IID	Non-IID	IID	Non-IID
10	97.69 (0.07) *	91.27 (0.19) *	87.96 (2.55) *	84.29 (0.15) *	87.47 (0.17) *	53.65 (0.46) *	66.98 (2.05) *	16.26 (0.2) *	35.55 (3.67) *
20	98.35 (0.08) *	94.27 (0.03) *	92.15 (3.03) *	91.19 (0.07) *	91.19 (0.18) *	76.70 (0.03) *	86.15 (0.32) *	47.21 (0.76) *	66.09 (1.62) *
40	98.78 (0.06) *	96.33 (0.04) *	94.72 (2.11) *	94.27 (0.04) *	95.05 (0.12) *	89.59 (0.12) *	91.27 (0.16) *	71.77 (0.34) *	84.70 (0.43) *
60	98.94 (0.02) *	97.08 (0.04) *	95.68 (1.81) *	95.54 (0.02) *	96.21 (0.06) *	92.10 (0.06) *	93.06 (0.14) *	85.77 (0.12) *	88.9 (0.19) *
80	99.02 (0.07) *	97.53 (0.05) *	96.28 (1.56) *	96.31 (0.03) *	96.67 (0.1) *	93.35 (0.04) *	94.13 (0.20) *	88.65 (0.05) *	90.74 (0.12) *
100	99.16 (0.05) *	97.74 (0.05) *	96.7 (1.32) *	96.73 (0.04) *	97.05 (0.1) *	94.13 (0.03) *	94.93 (0.22) *	90.39 (0.03) *	91.79 (0.15) *

* The values in parentheses represent the standard deviation.

As depicted in Table 4, the accuracy markedly depends on the number of nodes. With N = 10, the accuracy of federated learning is 97.74%, slightly lower than the 99.01% seen when N = 1, representing a centralized single model. As the number of participating nodes increases, each node receives less data, leading to decreased local learning accuracy. This is further analyzed to result in a reduction in the overall accuracy of federated learning. Particularly, as the node count grows, the accuracy for non-IID distributions is slightly higher than for IID. In non-IID scenarios, as Table 3 suggests, some nodes receive significantly more data than in the IID setup, enabling these nodes to achieve greater local learning accuracy, thereby enhancing the overall accuracy of federated learning. The local

accuracy of each node influences the overall performance, suggesting that if individual nodes develop effective local models with their independent data, federated learning with these nodes yields more accurate results. Figure 1 illustrates the accuracy of the proposed model according to the number of nodes per round in IID and non-IID distributions.



(**a**) (**b**)

Figure 1. Comparison of accuracy based on the number of nodes in IID and non-IID cases. (**a**) IID Accuracy. (**b**) Non-IID Accuracy.

We next evaluate the effectiveness of the proposed model by examining the size of communication data and computation time. The model encompasses *Setup* and *Learning* phases; the *Setup* phase occurs only once at the onset of federated learning. A critical task during the *Setup* phase involves each node generating its own public key $<pk_i, c_i>$ after the server has created the public ring A. Subsequently, the server crafts a group public key $<PK_U, C_U>$ using these keys.

The *Learning* phase is carried out repeatedly in each round of federated learning. During this phase, each node produces and forwards to the server D_i, which is masked with a randomly chosen mask M_i for the local parameters w_i along with the ciphertext E_i of M_i. The server aggregates them, computes D_U (the sum of D_i) and E_U (the sum of E_i), then calculates the global update w from these values, and distributes it to the nodes. The exact sizes of these parameters at the key stages are itemized in the ensuing Table 5.

Table 5. Data sizes of main parameters.

Parameters	Size (Byte)
A	13,107,200 (=12.5 MB)
pk_i, PK_U	13,631,488 (=13 MB)
c_i, C_U	6,815,744 (=6.5 MB)
w_i, D_i, D_U, w	174,720 (=0.17 MB)
E_i, E_U	13,107,200 (=12.5 MB)

In the *Setup* phase, the public ring parameter A transmitted by the server to each node is approximately 12.5 MB, and the combined size of PK_U and C_U is about 19.5 MB, independent of the node count. Similarly, the pair of pk_i and c_i transmitted by each node to the server is also roughly 19.5 MB. During the *Learning* phase, each node also sends approximately 12.5 MB of the ciphertext E_i along with about 0.17 MB of the masked local parameters D_i. The global update w, which the server sends back to the nodes, is approximately 0.17 MB. Compared to the masking-based aggregation model, the size of the data transmitted during the aggregation process in the proposed model is significantly larger. In the masking-based aggregation model, only the masked local parameters are transmitted, making the size of the transmitted data proportional to the size of the local

parameters, which is similar to the size of D_i (0.17 MB) in our experiments. In the proposed model, both the masked parameters D_i and the ciphertext of the mask E_i are transmitted. While D_i is proportional to the size of the local parameters, E_i is independent of the size of the local parameters and is instead determined by the ring polynomial used in HE. Therefore, even if the number of local parameters increases, the size of E_i remains constant. However, to ensure the security of the proposed HE, a 16-bit degree ring polynomial, along with 800-bit and 1600-bit moduli, is required, which inevitably increases the size of the transmitted data.

We further analyze the actual time required to perform the main operations associated with HE, which represent the most time-intensive computations in our aggregation model. During the *Setup* phase, we determined the average time for the server to generate the public ring polynomial A, the average time for each node to compute and send its own pk_i and c_i pair, and the average time for the server to compute and dispatch the group public key pair, PK_U and C_U. For the *Learning* phase, we assessed the average time for each node to calculate and transmit its local update pair, D_i and E_i, and the average time for the server to compute the global update by aggregating all D_is and E_is. Table 6 presents a summary of the actual average times required to execute these principal operations. These durations are the averages of all times recorded in individual experiments conducted with different numbers of nodes (N = 10, 20, 50, and 100) under both IID and non-IID conditions.

Table 6. The actual average time to perform main operations related to HE.

Phase	Actor	Operations	Average Time (ms)
Setup	Server	Creating a ring polynomial A	34.9399
		Sending A to each node	1568.971
	Node	Computing pk_i and c_i pair	114.8651
		Sending pk_i and c_i to the server	1117.554
	Server	T_{PKC}: adding a single pk_i to PK_U and adding a single c_i to C_U	4.763361
		Computing PK_U and C_U for all N nodes	$N \cdot T_{PKC}$
		Transmitting PK_U and C_U to each node	1263.832
Learning	Node	Computing the D_i and E_i pair	60.08073
		Transmitting D_i and E_i to the server	590.122
	Server	T_D: Adding a single D_i to D_U	0.026612
		Computing D_U for all N nodes	$N \cdot T_D$
		T_E: Adding a single E_i to E_U	7.099121
		Computing E_U for all N nodes	$N \cdot T_E$
		Computing w from D_U and E_U, including decoding E_U	0.15177

As indicated in Table 6, the computation time for generating ring parameters is approximately 35 ms, while generating individual public key pairs takes about 115 ms. Masking the local parameters and encrypting the mask value requires around 60 ms. All these computations are performed independently at each node, and the computation time is determined by the degree of the ring polynomial used in HE, regardless of the number of nodes. Therefore, even as the scale of federated learning grows, the execution time for each node remains the same. On the other hand, the time required for *FS* to aggregate all local updates scales proportionally with the number of nodes, with each addition operation lasting about 5 to 7 ms. While the total aggregation time increases with the number of nodes, the time required for a single addition operation is relatively short. As a result, even if the scale of federated learning grows to thousands of nodes, aggregation can still be completed within a few seconds.

Nonetheless, the data transmission time between the server and nodes is significantly longer, attributable to the large size of the transmitted data, ranging from 590 ms to 1569 ms. Although communication time is highly dependent on the specific networking environment, the experimental results obtained under the same local network conditions may not be broadly applicable. Nevertheless, it is clear that HE increases the communication load, leading to longer communication times. Excluding communication time, the duration required for each node to generate its local update, and the time for the server to aggregate all local updates during each *Learning* phase, is relatively short, demonstrating that the proposed model is feasible for real-world applications.

Figure 2 presents the total aggregation time based on the number of nodes under the non-IID distribution. The total aggregation time represents the duration required to complete 100 rounds of the Learning phase. This includes the time taken by each node to update its local model, generate local updates (including mask encryption), and transmit these updates, in each round. In the non-IID distribution, since the size of local dataset of each node is different, the execution time of each node is different, resulting in different response times to *FS*. We evaluated two scenarios: one with dropout occurrences and one without any dropouts. To handle dropouts, a waiting time threshold for each round must be defined. If a node failed to deliver updates to *FS* within the pre-defined waiting time, it was classified as a dropout. The waiting time per round was determined based on the maximum round time observed during our experimental measurements. In the absence of dropouts, the aggregation proceeded to the next step immediately after receiving updates from all nodes, even if the designated waiting time had not elapsed. Conversely, in the presence of dropouts, *FS* waited for the full pre-defined waiting time before initiating re-aggregation with the remaining active nodes. As the number of nodes increased, the time required for *FS* to receive and aggregate updates also grew, leading to a proportional increase in the total aggregation time. When no dropout occurred, for $n = 10$, the average total aggregation time was approximately 933 s; for $n = 20$, it was 2018 s; for $n = 50$, 7092 s; and for $n = 100$, 13,499 s.

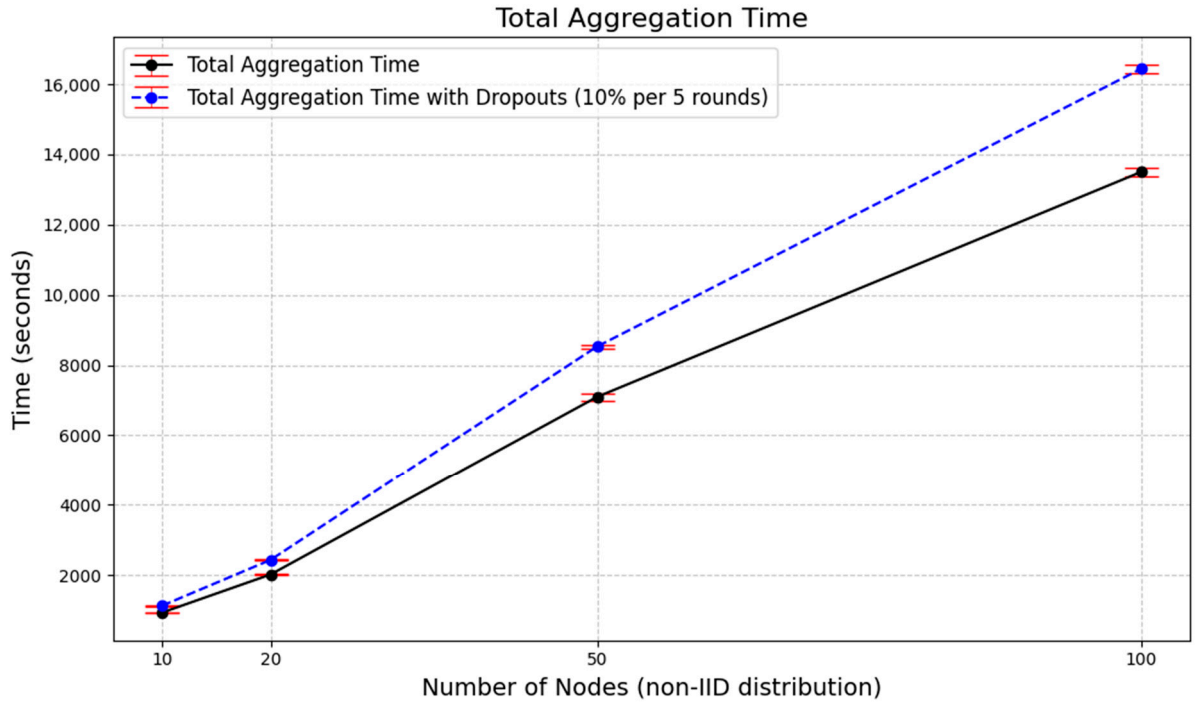


Figure 2. Total aggregation time by the number of nodes with non-IID distribution.

For a dropout scenario, when assumed a 10% dropout rate every 5 rounds, the average total aggregation time was as follows: n = 10, 1129 s; n = 20, 2434 s; n = 50, 8528 s; and n = 100, 16,436 s. These results demonstrate that both the number of nodes and the occurrence of dropouts significantly impact the total aggregation time. Dropouts, in particular, introduce additional waiting times and computational overhead, exacerbating the aggregation delays as the scale of the system grows.

Finally, we compare the key features of our proposed model with existing aggregation models. We employ the model by K. Bonawitz et al. [22], which represents a masking-based aggregation strategy, alongside two recently proposed HE-based aggregation strategies by J. Park and H. Lim [15], and W. Liu et al. [16]. These HE-based models enable the use of different encryption keys for nodes. Table 7 below summarizes this comparison.

Table 7. Comparison of the key features of the proposed model with existing aggregation models.

Features		**The Proposed Model**	**Masking-Based Aggregation by K. Bonawitz et al. [22]**	**HE-Based Aggregation by J. Park and H. Lim [15]**	**MKFHE-Based Aggregation by W. Liu et al. [16]**
Privacy-preserving strategies for local parameters		Masking and HE (based on *CKKS-HE*)	Masking	HE (based on a Distributed Homomorphic Cryptosystem)	HE (based on *CKKS-HE*)
Use of additional third parties		X	X	Computation Provider (CP)	X
The number of interactions required for aggregation (Assume that the set of nodes for aggregation has been determined and no dropouts occurred)	Number	2	4	4	4
	Interactions	N→S: send local updates S→N: send global update	N→S: send masks for other nodes S→N: distribute masks N→S: transmit masked local update S→N: transmit global update	N→S: transmit encrypted local update S→CP: transmit partially decrypted aggregation CP→S: transmit encrypted sum S→N: transmit encrypted global update	N→S: transmit encrypted local update S→N: transmit encrypted aggregation N→S: transmit partial decryption S→N: transmit global update
Type of global update retrieved by the server		Global update	Global update	Encrypted global update	Global update
Use of different keys for nodes		O	N/A	O	O
Update encryption key when dropout occurs		X	N/A	N/A	O
Decryption process		X	N/A	O (each node needs to decrypt the encrypted global update)	O (each node needs to partially decrypt the given aggregation)

In the masking-based aggregation method, each node must generate masks for all participating nodes, leading to computational and communication overhead that increases

proportionally with the total number of nodes. Additionally, mask sharing among all nodes must be completed before aggregation, making this approach unsuitable for self-adaptive models where the composition of participating nodes can dynamically change.

In the MKHE-based aggregation method, individual encryption keys are used by each node to enhance the security of local updates. However, the decryption process requires all nodes to provide partial decryptions for the aggregated ciphertext, adding complexity. Furthermore, when the aggregation group changes, the nodes in the new group must perform key refreshing processes, which can be inefficient in environments where node composition frequently changes.

Compared to existing aggregation models, the main advantage of our model is its ability to simplify and streamline the aggregation process. The proposed model significantly reduces the number of interactions between the server and nodes. Specifically, only two interactions are required to complete the global model update, provided there are no dropouts. While the proposed model simplifies the aggregation process, the use of homomorphic encryption results in an increase in data size, which may impact communication efficiency. However, the generation of local updates is performed independently on each node, and the size of the data depends solely on the security strength of the homomorphic encryption and the local model parameters. Therefore, the computational and communication overhead for each node is independent of the total number of participating nodes. Moreover, even when the aggregation group changes, the server only needs to update and distribute the group key for the new node group. This makes the proposed model particularly useful for self-adaptive models, where node configurations are dynamic and require flexible and efficient aggregation. Although the proposed model has the drawback of increased data size, it improves communication rounds and offers scalability for varying numbers of nodes and dynamic node groups.

5. Discussion and Conclusions

In this paper, we have proposed a novel and practical aggregation method for federated learning, based on masking and HE techniques. This scheme preserves the privacy of local model parameters by masking them with a user-chosen random value, and the masking values are securely encrypted and eliminated using the proposed MKHE technique. The scheme does not require communication or data sharing among nodes and minimizes interactions between the nodes and the server. It enhances the security of user-chosen masking values by allowing nodes to use different encryption keys, and simplifies the aggregation process by enabling the server to compute the aggregated result of the actual local parameters without needing to perform a separate decryption procedure. Even if dropouts occur, the remaining active nodes can independently generate new updates and re-aggregate them without changing their encryption keys. As a result, our model requires only two interactions to complete the global model update, provided there are no dropouts. We believe that the proposed model is highly significant as it minimizes the inefficiency caused by using MKHE techniques and maximizes practicality and security for secure aggregation in federated learning.

We conducted experiments to evaluate the aggregation accuracy and computational efficiency of the proposed model. Compared to raw data aggregation accuracy, no decrease in accuracy was observed with our masking and HE-based aggregation. In both cases, the accuracy remained at approximately 97.7% under identical conditions. The model's accuracy is significantly influenced by the number of nodes, achieving a peak accuracy of approximately 97.74% when $N = 10$ under IID distribution. This accuracy is about 1.27% lower than that of a centralized single model when $N = 1$. The overall accuracy of federated learning from our experiments depends on the accuracy of each node's local model. We

anticipate improvements in federated learning overall accuracy if each node can develop a robust local model using their independent training data.

The time required to complete a single round of our aggregation protocol has been assessed. On average, computing main parameters related to HE, such as a ring parameter, key pair, and local update of each node, took between 35 ms and 115 ms. However, the average transmission time of these parameters between the server and nodes was notably longer, ranging from 590 ms to 1580 ms when performed within the same local network environment. This delay is attributed to the increased data size due to HE. In terms of security, our experiments utilized a 16-bit ring polynomial, an 800-bit decryption modulus, and a 1600-bit encryption modulus. The sizes of the ring parameter A, the public key pk_i, and the encrypted mask E_i were approximately 13 MB, whereas the commitment c_i was about 6.5 MB and the masked local parameter D_i was roughly 0.17 MB. Given the significant data volumes involved, communication delays are likely as the communication load increases; however, these delays are effectively mitigated by minimizing the number of communication rounds. Compared to existing aggregation models, our proposed model significantly reduces the number of interactions between the server and nodes, requiring only two interactions to complete the global model update, provided there are no dropouts. We streamlined the aggregation process to its optimal point. and believe our model considerably enhances the efficiency and real-world applicability of the overall aggregation process.

We aim to establish a secure and practical federated learning model for medical diagnosis. As a potential solution, we proposed the MHESA-based federated learning model. Given the critical importance of privacy protection in medical data, federated learning is highly appropriate for building comprehensive medical diagnostic models using the distinct datasets held by individual hospitals. Hospitals often possess substantial amounts of differentiated medical data and sufficient computational infrastructure to perform local learning independently. Furthermore, federated learning for medical data does not require frequent or real-time processing, making it relatively resilient to communication delays. Therefore, hospitals can periodically engage in federated learning using the proposed model. Each hospital updates its local model and transmits it to the federated server (FS), which promptly updates the global model by aggregating the data from all participating hospitals and sharing the updated global model with them. While the use of homomorphic encryption may result in increased computational and transmission costs, these are manageable within the scope of hospital-level systems. Communication delays can also be mitigated by allowing adequate waiting times. Most importantly, the proposed model eliminates the need for collaborative operations between hospitals and minimizes communication rounds between hospitals and the federated learning server, thereby enhancing the overall efficiency of the federated learning for medical data.

The drawback of the proposed method is that, if only one node is determined as dropout, one other active node must be also treated as dropout to ensure the privacy of the dropout node. This issue needs to be improved so that all active nodes can participate in the aggregation under any disruption. In addition, the experiments in this paper were conducted on 100 nodes within the same network, which imposes limitations on the analysis of communication delays in real federated learning environments. In each round of federated learning, a waiting time must be set to receive data from all nodes and to identify dropout nodes. Determining an appropriate waiting time requires consideration of various factors, including the computational workload of each local node, computing power, data size for transmission, and the communication environment. Based on these factors, an analysis of communication delays in actual network conditions is necessary.

In future work, we plan to address the issue of determining an appropriate waiting time, considering real-world communication environments, as part of improving the pro-

posed model. Also, we will continue to develop methods to improve dropout management, reduce communication overhead and validate the aggregation result.

Author Contributions: Conceptualization, S.P.; methodology, S.P., J.L. and K.H.; software, J.L. and K.H.; validation, S.P., J.L., K.H. and J.C.; formal analysis, S.P.; data curation, J.C.; writing—original draft preparation, S.P., J.L. and K.H.; writing—review and editing, S.P. and J.C.; supervision, S.P.; project administration, S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (NRF-2021R1F1A1063172).

Data Availability Statement: The original data presented in the study are openly available in http://yann.lecun.com/exdb/mnist (accessed on 20 November 2024).

Conflicts of Interest: The authors declare no conflict of interest in this paper.

References

1. McMahan, H.B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA, 9–11 May 2017; Volume 54.
2. Zhu, L.; Liu, Z.; Han, S. Deep leakage from gradients. *arXiv* **2019**, arXiv:1906.08935.
3. Wang, Z.; Song, M.; Zhang, Z.; Song, Y.; Wang, Q.; Qi, H. Beyond inferring class representatives: User-level privacy leakage from federated learning. In Proceedings of the IEEE INFOCOM, Paris, France, 29 April–2 May 2019; pp. 2512–2520.
4. Geiping, J.; Bauermeister, H.; Dröge, H.; Moeller, M. Inverting gradients—How easy is it to break privacy in federated learning? In Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Online, 6–12 December 2020.
5. Yao, A.C. Protocols for secure computations. In Proceedings of the 23rd IEEE Annual Symposium on Foundations of Computer Sciecne (sfcs 1982), Chicago, IL, USA, 3–5 November 1982; pp. 160–164.
6. Shamir, A. How to share a secret. Commun. *ACM* **1979**, *22*, 612–613. [CrossRef]
7. Geyer, R.C.; Klein, T.; Nabi, M. Differentially private federated learning: A client level perspective. In Proceedings of the NIPS 2017 Workshop: Machine Learning on the Phone and other Consumer Devices, Long Beach, CA, USA, 8 December 2017.
8. Wei, K.; Li, J.; Ding, M.; Ma, C.; Yang, H.H.; Farokhi, F.; Jin, S.; Quek, T.Q.S.; Poor, H.V. Federated Learning with Differential Privacy: Algorithms and Performance Analysis. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3454–3469. [CrossRef]
9. Leontiadis, I.; Elkhiyaoui, K.; Molva, R. Private and dynamic timeseries data aggregation with trust relaxation. In Proceedings of the International Conferences on Cryptology and Network Security (CANS 2014), Seoul, Republic of Korea, 1–3 December 2010; Springer: Berlin/Heidelberg, Germany, 2014; pp. 305–320.
10. Rastogi, V.; Nath, S. Differentially private aggregation of distributed time-series with transformation and encryption. In Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 10), Indianapolis, IN, USA, 6–10 June 2010; pp. 735–746.
11. Halevi, S.; Lindell, Y.; Pinkas, B. Secure computation on the Web: Computing without simultaneous interaction. In *Advances in Cryptology—CRYPTO 2011*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 132–150.
12. Leontiadis, I.; Elkhiyaoui, K.; Önen, M.; Molva, R. PUDA—Privacy and Unforgeability for Data Aggregation. In *Cryptology and Network Security. CANS 2015*; Springer: Cham, Switzerland, 2015; pp. 3–18.
13. Aono, Y.; Hayashi, T.; Wang, L.; Moriai, S. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Trans. Inf. Forensics Secur.* **2017**, *13*, 1333–1345.
14. Fang, H.; Qian, Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. *Future Internet* **2021**, *13*, 94. [CrossRef]
15. Park, J.; Lim, H. Privacy-Preserving Federated Learning Using Homomorphic Encryption. *Appl. Sci.* **2022**, *12*, 734. [CrossRef]
16. Liu, W.; Zhou, T.; Chen, L.; Yang, H.; Han, J.; Yang, X. Round efficient privacy-preserving federated learning based on MKFHE. *Comput. Stand. Interfaces* **2024**, *87*, 103773. [CrossRef]
17. Jin, W.; Yao, Y.; Han, S.; Gu, J.; Joe-Wong, C.; Ravi, S.; Avestimehr, S.; He, C. FedML-HE: An Efficient Homomorphic-Encryption-Based Privacy-Preserving Federated Learning System. *arXiv* **2023**, arXiv:2303.10837.
18. Wibawa, F.; Catak, F.O.; Kuzlu, M.; Sarp, S.; Cali, U. Homomorphic Encryption and Federated Learning based Privacy-Preserving CNN Training: COVID-19 Detection Use-Case. In Proceedings of the 2022 European Interdisciplinary Cybersecurity Conference (EICC'22), Barcelona, Spain, 15–16 June 2022; Association for Computing Machinery: New York, NY, USA, 2022; pp. 85–90. [CrossRef]

19. Hijazi, N.M.; Aloqaily, M.; Guizani, M.; Ouni, B.; Karray, F. Secure Federated Learning With Fully Homomorphic Encryption for IoT Communications. *IEEE Internet Things J.* **2024**, *11*, 4289–4300. [CrossRef]
20. Sanon, S.P.; Reddy, R.; Lipps, C.; Schotten, H.D. Secure Federated Learning: An Evaluation of Homomorphic Encrypted Network Traffic Prediction. In Proceedings of the IEEE 20th Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 8–11 January 2023; pp. 1–6. [CrossRef]
21. Cheon, J.H.; Kim, A.; Kim, M.; Song, Y. Homomorphic Encryption for Arithmetic of Approximate Numbers. In *Advances in Cryptology-ASIACRYPT 2017*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2017; Volume 10624. [CrossRef]
22. Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedoney, A.; McMahan, H.; Patel, S.; Ramage, D.; Segal, A.; Seth, K. Practical Secure Aggregation for Federated Learning on User-Held Data. *arXiv* **2016**, arXiv:1611.04482.
23. Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H.B.; Patel, S.; Ramage, D.; Segal, A.; Seth, K. Practical secure aggregation for privacy-preserving machine learning. In Proceedings of the ACM SIGSAC Conferences on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 1175–1191.
24. Ács, G.; Castelluccia, C. I have a DREAM! (DiffeRentially privatE smArt Metering). In *Information Hiding. IH 2011*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 118–132.
25. Goryczka, S.; Xiong, L. A comprehensive comparison of multiparty secure additions with differential privacy. *IEEE Trans. Dependable Secur. Comput.* **2017**, *14*, 463–477. [CrossRef] [PubMed]
26. Elahi, T.; Danezis, G.; Goldberg, I. Privex: Private collection of traffic statistics for anonymous communication networks. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, 3–7 November 2014; pp. 1068–1079.
27. Jansen, R.; Johnson, A. Safely Measuring Tor. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 1553–1567.
28. So, J.; Güler, B.; Avestimehr, A.S. Turbo-Aggregate: Breaking the Quadratic Aggregation Barrier in Secure Federated Learning. *arXiv* **2020**, arXiv:2002.04156. [CrossRef]
29. Kim, J.; Park, G.; Kim, M.; Park, S. Cluster-Based Secure Aggregation for Federated Learning. *Electronics* **2023**, *12*, 870. [CrossRef]
30. LeCun, Y.; Cortes, C.; Burges, C.J. MNIST Handwritten Digit Database. 2010. Available online: http://yann.lecun.com/exdb/mnist (accessed on 4 October 2024).

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 May 2024, accepted 15 June 2024, date of publication 24 June 2024, date of current version 1 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3418016

RESEARCH ARTICLE

Enhancing Efficiency in Privacy-Preserving Federated Learning for Healthcare: Adaptive Gaussian Clipping With DFT Aggregator

MUHAMMAD AYAT HIDAYAT, (Graduate Student Member, IEEE), YUGO NAKAMURA, (Member, IEEE), AND YUTAKA ARAKAWA, (Member, IEEE)
Department of Information Science and Technology, ISEE, Kyushu University, Fukuoka 819-0385, Japan
Corresponding author: Muhammad Ayat Hidayat (muhammad.971@s.kyushu-u.ac.jp)

This work was supported by Japan Science and Technology Agency (JST), PRESTO, Japan, under Grant JPMJPR21P7.

ABSTRACT Machine learning's exponential growth has transformed healthcare, with Federated Learning (FL) playing a pivotal role. Despite its significance, FL is vulnerable to privacy attacks. In response, researchers have integrated differential privacy (DP) into FL. Nevertheless, incorporating DP introduces challenges such as increased total communication costs and computational overheads due to the introduction of noise. This drawback renders FL with DP less viable for healthcare systems, characterized by numerous low-resource devices and network bandwidth constraints. To overcome this limitation, we propose integrating a Discrete Fourier Transform (DFT) aggregator post-noise addition to transform the gradient generated by local training before sending it to the central server. This process reduces the gradient size and provides rudimentary encryption. The evaluation results reveal the superior performance of our proposed method, demonstrating an enhanced accuracy ranging from 0.2% to 2% compared to existing differential privacy techniques, including RDP, DP-SGD, ZcDP, LDP-Fed, and DP-AdapClip. Our approach substantially reduces the total communication costs (ranging from 6% to 43% across different privacy budgets) with faster training times in healthcare datasets such as the PIMA Indian database and Breast Cancer Histopathology Images.

INDEX TERMS Federated learning, differential privacy, adaptive Gaussian clipping, healthcare.

I. INTRODUCTION

The field of machine learning (ML) has undergone a remarkable surge in growth in the past decade, resulting in substantial progress across a multitude of technologies and applications. Particularly in healthcare systems, ML has played a pivotal role in transforming the landscape of patient care and medical research. Within the realm of ML, one noteworthy approach is Federated Learning (FL), which has emerged as a significant paradigm. FL empowers the training of machine learning models across numerous distributed devices without centralizing sensitive data. This innovation is particularly crucial in healthcare, where the privacy and security of medical data are paramount. In the FL framework, a collaborative network of clinicians develops their own local learning model using their respective medical data. These individual models generate incremental updates before being subsequently transmitted to a central coordinating node. This decentralized approach ensures that each clinician retains control over their data governance and privacy requirements, aligning with the stringent regulations governing healthcare information [1]. While FL advocates for data privacy, it remains susceptible to attacks that can diminish or destabilize model accuracy. A prevalent attack

The associate editor coordinating the review of this manuscript and approving it for publication was Ganesh Naik.

is known as a ''poisoning attack'' to distort the model by introducing corrupted training data. This action renders the server incapable of distinguishing data sources from authorized clients [2]. Furthermore, other types of attacks can be executed within the FL framework. The ''model reconstruction attack'' [3] involves monitoring and recording all communication between clients and the server, which is then used to reconstruct the client's local model. The ''GAN-based inference attack'' identifies user inputs on the server and selectively sends global updates to an isolated client [4]. Another attack, the ''inferring membership attack,'' seeks to exploit gradients from the SGD algorithm to access information related to the training process [5]. Given the multitude and diversity of these attacks, FL alone may not be sufficient to safeguard the data used in model training under certain circumstances. In these cases, many researchers have applied the principle of differential privacy to improve privacy measures in Federated Learning. This involves introducing random noise into the model gradient to modify the original value before transmitting it to a central server for aggregation.

Simply adding noise to a model without optimization inevitably reduces its accuracy. The Adaptive Gaussian Clipping Differential Privacy (AGC-DP) method, discussed in our previous research [6], offers a solution to improve the trade-off between privacy and model effectiveness. By incorporating Privacy Loss Distribution (PLD), dynamically adjusting privacy parameters based on data complexity, and optimizing data sampling through Poisson sub-sampling, AGC-DP outperforms other DP algorithms in terms of model accuracy. However, we noted that AGC-DP incurs elevated total communication costs, which often lead to higher latency. This latency can be detrimental in critical healthcare scenarios where timely data processing is essential.

Lowering communication costs is particularly beneficial for healthcare facilities, as it enables real-time synchronization of patient records across different locations. This ensures that healthcare providers have the most current information for better decision-making. Therefore, in this paper, we extend our previous work on AGC-DP by introducing a Discrete Fourier Transform (DFT) to encrypt and reduce the size of the gradient generated from local training before adding noise to it. This compressed noisy gradient is then sent to a central server for aggregation, reducing the gradient size sent to the central server and thereby lowering total communication costs. We also evaluate the system using healthcare datasets, such as the PIMA Indian diabetes dataset and Breast Cancer Histopathology CT Images, which contain sensitive medical information. This evaluation aligns with the system's purpose of safeguarding such sensitive data. Our key contributions can be summarized as follows:

1) **Enhanced Gradient Compression**: We introduced a Discrete Fourier Transform (DFT) aggregator to reduce the gradient size and implement basic encryption on AGC-DP. We applied DFT after local training and then added noise to the compressed gradient before sending it to the central server. This approach significantly decreases the total communication cost while enhancing security by protecting the gradient before transmission to the central server.
2) **Comprehensive Evaluation**: We conducted in-depth simulations of Enhanced AGC-DP using two distinct healthcare datasets. Through rigorous comparison with other DP methodologies, we assessed accuracy, total communication cost, and CPU time to demonstrate the effectiveness of our enhancements.

The structure of this article is outlined as follows. Section II introduces the related work on implementing secure FL in healthcare. In Section III, we describe the features related to our proposed method, including a system overview and threat model. Section IV presents our proposed method, including the description of the detailed process sequentially. Section V provides a detailed description of the datasets used for the learning process. Section VI describes the model detail used, including layers and activation type. Section VII explains the simulation parameters, specifically the environment and simulation scenario. Section VIII presents the simulation results and discussion. In Section IX, conclusions and future work are presented.

II. RELATED WORK

A. PRIVACY-PRESERVING FEDERATED LEARNING

In FL, client devices refrain from sending raw data to the central server, opting instead to exchange only the model's gradients. This protocol protects the raw data, ensuring it remains exclusively stored on local devices, thus bolstering the security and privacy of FL. However, despite clients transmitting only model gradients to the central server, research, such as [7], has demonstrated that these gradients can still be utilized to reconstruct the local model and deduce information about the training dataset. To tackle this issue, various strategies have emerged.

One approach, Homomorphic Encryption (HE) [8], utilizes the ElGamal algorithm to encrypt local gradients before sending them to the central server. However, the application of HE may face problems, especially with the complex models often encountered in deep learning, as encrypting many layers used in such models can introduce significant computational overhead. Conversely, another strategy employs blockchain technology [9], leveraging blockchain and secure global aggregation methodologies to safeguard against potential attacks from malicious edge devices and servers. Nonetheless, storing data continuously on the blockchain could lead to increased storage demands, while integrating complex blockchain technology could increase computational overhead.

Differential Privacy (DP) offers another solution by injecting random noise into the released information to distort the original values. In the realm of FL, several studies have explored DP implementation methods, including [10], which

utilizes Renyi differential privacy to prevent privacy leaks from the FL-MAC as discussed in the paper. DP-SGD [11], which adds noise based on the proportion of the clipping norm to the gradient updates, and ZcDP [12], which employs zero-concentrated differential privacy (ZcDP) to achieve a tighter privacy guarantee, thereby reducing the amount of noise added to the model compared to other methods with the same DP assurance.

There are other approaches, such as LDP−Fed [13], which use local differential privacy (LDP) based on the local device privacy budget and using utility-aware privacy perturbation to prevent uncontrolled noise from overwhelming the FL training algorithm in the presence of large, complex model parameter updates. The evaluation of the work shows that it achieved improved accuracy but did not show how efficient the method was in terms of communication and computational overhead, and DP-AdapClip [14] which is an approach that simplifies the training of neural networks in federated learning through user-level differential privacy. Instead of using a fixed clipping norm for each user's model update, this work proposes setting it dynamically based on a quantile of the update norm distribution, adapting to different tasks. This eliminates the need to fine-tune the fixed clipping hyperparameters, making the process much more efficient.

Another approach is presented in [15], which is also an extension of [6], as well as our previous work. This approach utilizes DCT-pruned weights to compress gradients using a dynamic compression ratio. The compression ratio is determined based on the device's resource availability, including processor and memory capacity. The primary distinction between our proposed method and [15] lies in how to manipulate the model weights. In the DCT-pruned weight approach, insignificant coefficients produced from the Discrete Cosine Transform (DCT) of the model weights are first pruned or removed and then masked before adding noise. This process involves identifying and discarding coefficients that contribute minimally to the representation of the model, thus reducing the overall size of the weight matrix. On the other hand, in this work, we employ Discrete Fourier Transform (DFT) and rotation techniques to manipulate the model weights. Specifically, we utilize the DFT to transform the model weights into the frequency domain, where rotations are applied before noise is added to enhance the privacy and security of the gradients. Another difference is that we implement [15] in a cloud environment and resource-constrained device (edge device) like Raspberry Pi. So, we specifically designed the method to fit the lower resource in an edge device, different from the work presented in this article, which focuses on healthcare applications that may use high image resolution, which is not really the kind of data used in edge devices.

B. PRIVACY-PRESEVING FEDERATED LEARNING IN HEALTHCARE

The implementation of privacy-preserving federated learning in the healthcare system is commonly utilized, given the sensitive nature of healthcare data, which includes confidential information such as patient records and medical diagnoses. Data protection represents one of the main challenges in the healthcare system. Several studies have implemented privacy-preserving federated learning, such as the work by [16], which employs traditional homomorphic encryption to encrypt model updates from local IoT devices. This data is then collected in a fog node and sent to the central server for aggregation and decryption. However, the use of encryption, as demonstrated in works like [17] and [18], incurs a high computational overhead. Particularly when using longer encryption keys, even if encryption is only performed once during the learning process.

Another work is [19], which combines differential privacy to ensure the privacy of the data and blockchain to record all user activity in the system to ensure transparency. Even though the evaluation results show that the method has better resource consumption (memory and CPU), it is unclear how much total communication cost is generated from this method. Even though network latency is mentioned, it solely depends on the network speed and environment. There is also [20], which combines differential privacy, Secure Multi-Party Computation (SMPC), and Homomorphic Encryption (HE) for model updates to provide maximum privacy, mitigating potential data leakage risks. However, the combination of these features can have a very high impact on the computational overhead and total communication costs. Even though the method is 10% more efficient in terms of computational overhead than the state-of-the-art method, it is not clear how much of the total communication costs is affected by this method, which is one of the weaknesses of differential privacy in this context. The summary of related work and the feature comparison of our proposed method can be seen in Table 1.

TABLE 1. Feature comparison between related work and proposed method.

Method	HA	Low CO	Low TCC	Secure
Homorphic Encryption [8]	✓	✗	✗	✓
Blockchain [8]	✓	✓	✗	✓
RDP [10]	✓	✗	✗	✓
DP-SGD [11]	✓	✗	✗	✓
ZcDP [12]	✓	✗	✗	✓
LDP-Fed [13]	✓	✗	✗	✓
DP-AdapClip [14]	✓	✗	✗	✓
HE-Med [16]	✓	✗	✗	✓
PPFLHE [17]	✓	✗	✗	✓
HE-Covid19 [18]	✓	✗	✗	✓
Block-DP [19]	✓	✓	✗	✓
DP-HE-SMPC [20]	✓	✓	✗	✓
Proposed Method	✓	✓	✓	✓

* **HA** : High Accuracy, **TCC** : Total Communication Cost, **CO** :Computational Overhead

III. PREELIMINARIES

In this section, we describe the general features that constitute our proposed method. We begin by discussing the features of

our previous work (AGC-DP) and the infrastructure of our system, including an overview of the system and the threat model.

A. ADAPTIVE GAUSSIAN CLIPPING DP(AGC-DP)

AGC-DP is a strategy designed to enhance the security of Federated Learning by incorporating varying levels of noise into the model's gradient by perturbing its original values. The variable noise levels in AGC-DP are created through the following components:

1) PRIVACY LOSS DISTRIBUTION (PLD)

In AGC-DP, the Privacy Loss Distribution (PLD) establishes a stringent privacy guarantee. The PLD measures the logarithmic ratio between the likelihood of observing an outcome given different inputs. For two discrete distributions, μ_{up} and μ_{lo}, the privacy loss at an outcome $o \in \sup(\mu_{up})$ (meaning o is within the support of μ_{up}, the set of all possible values with non-zero probability) is defined as follows:

Definition 1: The PLD of involving μ_{up} and μ_{lo}, referred to as PLD μ_{up}/μ_{lo}, represents a distribution on $RU\infty$, where $y \sim PLD_{\mu_{up}/\mu_{lo}}$ is obtained by sampling $o \sim \mu_{up}$ and setting $y = \mathcal{L}_{\mu_{up}/\mu_{lo}}(o)$. Mathematically, this is expressed as:

$$\mathcal{L}_{\mu_{up}/\mu_{lo}}(o) := \ln\left(\frac{\mu_{up}(o)}{\mu_{lo}(o)}\right) \tag{1}$$

AGC-DP utilizes the Gaussian mechanism to generate noise. This mechanism is characterized by a Gaussian (or normal) random variable with a mean (μ) and a standard deviation (σ). The likelihood of a Gaussian random variable assuming a specific value is described by its probability density function (PDF). The PDF for a Gaussian random variable at a point x is given by: $x = \frac{1}{\sigma\sqrt{2\pi}} \times e^{\frac{(x-\mu)^2}{2\sigma^2}}$.

Where μ denotes the mean of the distribution, indicating its central tendency, and σ represents the standard deviation, which measures the spread or dispersion of the distribution. The function exp denotes the exponential function. In the context of the Gaussian mechanism used in differential privacy (DP), noise is added to the output of a function to ensure the privacy of individuals in the dataset is preserved. The amount of noise is typically scaled according to the sensitivity of the function. To introduce the concept of scaled sensitivity, a new variable, $\tilde{\Delta}$, is defined as $\tilde{\Delta} = \delta(f)/\sigma$. Here, $\Delta(f)$ represents the sensitivity of the function f, indicating the maximum change in f due to the modification of a single data point. σ represents the standard deviation of the Gaussian noise being added. The noise generated by the Gaussian mechanism follows a normal distribution, often standardized as $N(0, 1)$, signifying it has a mean of 0 and a standard deviation of 1. This standardization is critical as any Gaussian distribution can be transformed into the standard normal distribution through normalization. Mathematically, this is represented as:

$$\ln\left(\frac{\frac{1}{\sigma\sqrt{2\pi}} \times e^{-x^2/2}}{\frac{1}{\sigma\sqrt{2\pi}} \times e^{-(x-\tilde{\Delta})^2/2}}\right) = \frac{\tilde{\Delta}}{2}.(\tilde{\Delta} - 2(x)) \tag{2}$$

Here, the natural logarithm of a ratio compares the probability density function (PDF) of a standard normal distribution $N(0, 1)$ at point x with and without an additional noise term ($\tilde{\Delta}$). ($\tilde{\Delta} - 2(x)$) represents half of the scaled sensitivity ($\tilde{\Delta}$), indicating how much the function's sensitivity contributes to the noise level. ($\tilde{\Delta} - 2(x)$) represents the difference between the scaled sensitivity ($\tilde{\Delta}$) and twice the observed value ($2(x)$). This difference captures the discrepancy between the scaled sensitivity and the observed value and their influence on the distribution. By utilizing the above formula, AGC-DP can analyze the impact of noise addition on the Privacy Loss Distribution, which is crucial for determining the amount of noise to be added to the gradient.

2) SUB-SAMPLING

AGC-DP also incorporates a subsampling method to enhance privacy protection, employing the Poisson sub-sampled technique with a sampling probability of q. This method randomly selects individual data points to be included in a sub-sampled dataset independently, each chosen with a probability of q. The mechanism's output is subsequently generated based on this sub-sampled dataset. In this process, the Privacy Loss Distribution (PLD) is computed for each mechanism (including noise addition through Gaussian distribution and data subsampling) employed at AGC-DP, and their convolutions are analyzed alongside an evaluation of the divergence of the ϵ-hockey stick. This divergence metric indicates how closely the mechanism aligns with the desired privacy level, measuring the discrepancy between the actual and ideal privacy loss distributions in DP.

In the context of DP, mechanisms often operate within datasets where data points can be added or removed, such as during data collection or when individuals request data deletion. The addition of a data point to the dataset can potentially increase privacy loss by introducing additional sensitive information, while the removal of a data point may also impact privacy loss by altering the dataset's composition. Taking into account both addition $\mathcal{L}_{\mu/\nu'}$ and removal $\mathcal{L}_{\mu'/\nu}$ adjacency, the privacy loss functions for the Poisson sub-sampled mechanism are represented as:

$$\mathcal{L}_{\mu/\nu'} = \log(1 - q + q \cdot e^{-\mathcal{L}_{\mu/\nu(o)}}) \tag{3}$$

$$\mathcal{L}_{\mu'/\nu} = -\log(1 - q + q \cdot e^{\mathcal{L}_{\mu/\nu(o)}}) \tag{4}$$

3) ADAPTIVE CLIPPING

AGC-DP integrates adaptive clipping as part of its operations. In this regard, AGC-DP employs the initial clipping threshold, referred to as C^0, based on the target unclipped quantile UC. This threshold establishes the limit beyond which noisy data points are clipped to safeguard privacy

while upholding data utility. During the training phase, UC gradually decreases over time. This decay in UC affects C by influencing its value. Specifically, as UC decreases, indicating a lower tolerance for unclipped data points, C is adjusted accordingly to maintain the balance between privacy protection and model utility. This process can be defined as:

$$C_j = UC_{t-1} \cdot (1 - r)^t \tag{5}$$

(C_j) represents a clipping threshold, where $(1 - r)$ represents the decay factor determining how quickly the threshold decreases over time t. If the threshold is initially high, meaning r is small, then less noise is added, leading to a slower decrease in the threshold. Conversely, if the threshold is initially low, meaning r is large, more noise is added, leading to a faster decrease in the threshold. The decay of the target unclipped quantile every 20 iterations corresponds to the term UC_{t-1}, where the $t - 1$ indicates that the value of the unclipped quantile decays over time.

B. SYSTEM OVERVIEW

For the system's architecture, we consider a general Federated Learning (FL) system, illustrated in Fig.1, comprising a server and K clients. Each client is denoted by k_i, where i ranges from 1 to K, and each client possesses a local database $\mathcal{D}_{ki}$. The server aims to iteratively train a model using data from K-associated clients over T steps.

In our proposed system, the process starts with the central server's program setting up the global model as w_0. The next step involves determining the number of participating clients and establishing the noise level (S) to be injected into the gradient. Each client's program then updates its local model (w_t) and preprocesses the dataset for training purposes. Once ready, the client's program initiates the training process to generate its local model. Following this, the system applies the Discrete Fourier Transform (DFT) aggregator to the model gradient, converting it into a 1D rank tensor and rotating it to increase randomness. This produces c_t^k, which is subsequently added with noise determined by the previously calculated value of S. Following this, the client's program transmits the gradients with added noise, represented as $\hat{c}_t^k$, to the server. The server's program aggregates these gradients, which are subsequently subjected to an inverse Discrete Fourier Transform (DFT) to create an updated parameter for adjusting the global model. After completing the update process, the program shares the global model that has been updated with the clients, initiating the subsequent iteration round, which continues until $t = T$.

C. THREAT MODEL

For the threat model, we assume that the central server operates honestly. It accurately aggregates all uploaded gradients of the local model, ensuring no intentional omissions. Furthermore, clients show no interest in the private data of other clients and refrain from seeking further insights from the collaborative models. Our threat model can be defined as follows:

1) **Privacy leakage**. Local clients and central servers cannot conduct inference attacks directed towards individual clients for exchanging gradients in order to reconstruct their local model or training datasets.
2) **External adversaries**. External adversaries cannot manipulate data within the local client to introduce false information that could disrupt the model's learning process. However, they can eavesdrop on the transmitted data or gradients sent from the client to the central server, potentially enabling them to reconstruct training datasets.

IV. PROPOSED METHOD

In this section, we explain the steps of our proposed method in detail. As shown in Fig.1, our proposed method consists of nine steps: 1) Initialization, 2) Preparation for local training, 3) Privacy loss calculation, 4) Local training, 5) Implementation of the Discrete Fourier Transform (DFT) aggregator, 6) Noise addition, 7) Aggregation, 8) Inverse DFT aggregator, and 9) Global model update. In this section, we provide a detailed explanation of each process.

A. INITIALIZATION

In this phase, the server program will set up a global model, referred to as w_0. The central server and all clients collaborate to determine an initial weight range based on prior knowledge, which helps improve the model's convergence during training. Using this range, the server initializes w_0. The central server then randomly distributes w_0 to the participating clients for local training. This random distribution is crucial to avoid bias or favoritism towards specific clients by the central server during the training process.

B. PREPARATION FOR LOCAL TRAINING

During this phase, we set up all the necessary parameters for the training process. The first parameter defined was the base noise multiplier (bnm), which influences the amount of noise added to the local model on each client. Additionally, we established the number of base clients per round ($bcpr$). Furthermore, we determined the values for target epsilon ϵ_t and target delta δ_t. These values have a significant impact on the privacy parameters and play a role in controlling the level of privacy protection in the subsequent stages of the process.

C. PRIVACY LOSS CALCULATION

Like [15], we also used the Privacy Loss Distribution (PLD) defined in AGC-DP [6] to compute the privacy loss (ϵ), as outlined in Equations (3) and (4). The PLD generates PL(ϵ), clients per round (cpr), and noise multiplier (nm), as shown in lines 1 to 5 of Algorithm 1. The computed ϵ from PLD is then compared to the predefined target privacy (ϵ_t). This target privacy determines the desired level of privacy protection for the system (lower values indicate higher privacy, while higher values imply lower privacy). The values of cpr and nm will be adjusted based on ϵ until it

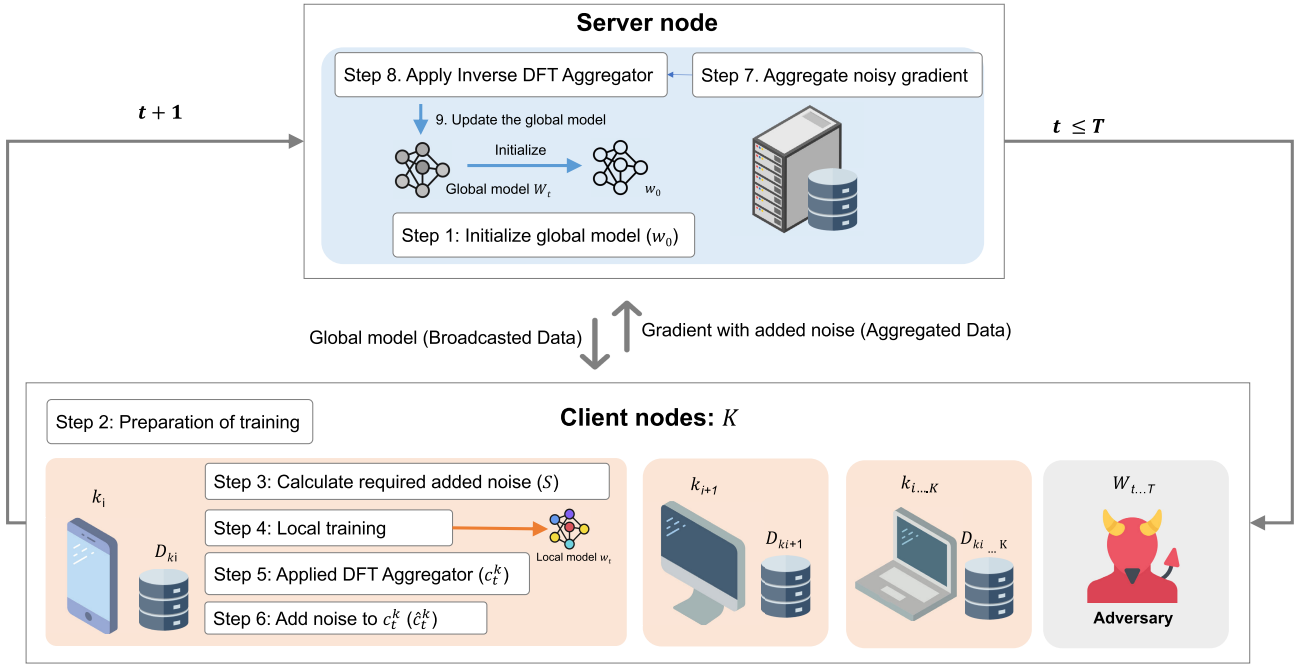


FIGURE 1. Overview of the proposed system.

Algorithm 1 Privacy Loss Calculation

Data: ϵ_t, δ_t, $bcpr = 50$, $bnm = 0.5$
Result: nm
1 **Function** `GetEps(`cpr`)`**:**
2 $nm \leftarrow bnm$
3 $nm\times = cpr/bcpr$
`// Calculate privacy loss`
4 $\epsilon = PLD_{accnt}(nm)$
5 **return** cpr, ϵ, nm
6 **Function** `FindClientNeeded()`**:**
7 $ep \leftarrow$ `GetEps(`$bcpr$`)`
8 **if** $ep[1] < \epsilon_t$ **then**
9 **return** ep;
10 **end**
11 **while** True **do**
12 $ls \leftarrow ep$
13 $ep \leftarrow GetEps(2 * ls[0])$
14 **end**
15 **return** ep

matches ϵ_t. Once achieved, the appropriate cpr and nm for ϵ_t are determined as a reference in lines 7 to 14 of Algorithm 1. These parameters are then utilized in local training, where cpr dictates the number of client participation in one iteration, and nm governs the amount of random noise added to the gradient after local training.

D. LOCAL TRAINING

In the initial stage of local training, the clipping process was initiated. During this phase, the target unclipped quantile gradually diminished, as specified in lines 2 of Algorithm 3. Alterations to the target unclipped quantile have a direct impact on the threshold value employed for gradient clipping. This threshold, in turn, indirectly affects the introduction of noise into the model gradient at regular intervals, typically every 20 iterations. Subsequently, the local clients commenced their training process, utilizing the parameters generated in stages 1-3. Each client conducted local model training, denoted as w_{kt}, for a single local epoch to ensure comprehensive local training. During each local epoch, the client meticulously processed their entire local dataset. To streamline and expedite the local training procedure while avoiding the substantial time delays associated with multiple local epochs, the client divided the local dataset into smaller batches. Following this segmentation, the client applied training to one randomly selected mini-batch from the collection of batches, one at a time.

E. APPLY DFT AGGREGATOR

In this phase, the gradient from local training was applied to the randomized discrete Fourier transform or DFT. The details of this process are described below.

1) **Flattens the gradients**: This process converts a multi-dimensional gradient w_t^k into a one-dimensional vector. This operation essentially "unwraps" the gradient, transforming it into a sequence of elements in a specific order, as seen in Algorithm 2 line 2-3.
2) **Pads the flattened gradients**: This process involves adding zero values to the flattened gradient flt so then the resulting gradient has an even number of elements.

Algorithm 2 DFT Aggregator

Data: w_t^k
Result: c_t^k
1 **Function** `ClientTrans` (w_t^k) **:**
  Data: $flt, padW, cw_t^k, rw_t^k, dw_t^k, uw_t^k$
  Result: c_t^k
2 **for** *each element x in* w_t^k **do**
3 $flt \leftarrow \text{Reshape}(x, [-1])$
4 **end**
5 **foreach** *element y in flt* **do**
6 **if** $size(y)\%2 = 0$ **then**
7 $num\_zeros \leftarrow 0$
8 **end**
9 **else**
10 $num\_zeros \leftarrow 1$
11 **end**
12 $padW \leftarrow \text{Concat}([y, num\_zeros])$
13 **end**
14 $cw_t^k \leftarrow \text{Complex}(rl = padW[0], im = padW[1])$
15 $rw_t^k \leftarrow \text{Rotate}(cw_t^k)$
16 $dw_t^k \leftarrow \text{FFT}(rw_t^k)$
17 $uw_t^k \leftarrow \text{Concat}(real(dw_t^k), imag(dw_t^k))$
18 **foreach** *element j in* uw_t^k **do**
19 $c_t^k \leftarrow \sqrt{\frac{\text{size}(j)}{2}}$
20 **end**
21 **return** c_t^k;

3) **Packs the gradient**: This operation transforms the gradient padded with zero $padW$ into a complex-valued gradient by creating complex numbers. The resulting complex gradient cw_t^k will have half as many elements as the original real gradient. The transformation is achieved by pairing the real rl and imaginary im values from the original gradient to form complex numbers collectively. The complex tensor will have half as many elements as the original gradient, indicated by $d/2$. In other words, if the original gradient had d elements, the resulting complex gradient would have $d/2$ complex numbers. This reduces the size of the gradient. This process can be seen in Algorithm 2 line 14.
4) **Rotating complex gradient coordinates**: This process applies a random phase shift to the individual complex numbers within the gradients cw_t^k as defined in Algorithm 2 line 15. Each complex number consists of a real part and an imaginary part, and an angle effectively rotates these two components randomly.
5) **Applies the discrete Fourier transform**: This process applies DFT to the cw_t^k to convert it from the spatial domain into the frequency domain as seen in Algorithm 2 line 16. In this context, the DFT is applied to the complex tensor. It reveals the frequency components present in the data after the previous transformations.
6) **Unpacks the complex gradient**: Unpacks the complex gradient: This step reverses the process of step 3. The complex gradient dw_t^k is converted back into a real gradient by separating the real and imaginary components of the complex tensor and concatenating them back into a real tensor with the length of d. This process can be seen in Algorithm 2 line 19.
7) **Normalizes the gradient**: This process divides the un- pack gradients uw_t^k as defined in Algorithm 2 line 18-20. This process ensures that the gradient remains within a desired range or that the magnitude of the gradient is consistent for the inverse process on the server. This process will generate the transform gradient c_t^k that will be added noise on the next process.

F. NOISE ADDITION

By incorporating adaptive Gaussian clipping, as detailed in [6], we determined a clipping threshold C by referencing a target unclipped quantile denoted as UC. This process involved gradually reducing UC every 20 iterations. Adjusting UC directly impacted the clipping threshold value, consequently affecting the amount of noise introduced to the gradient (S) as illustrated in Algorithm 3 in line 2.

Algorithm 3 Noise Applied to the Gradient Locally

Data: c_t^k
Result: $\hat{c}_t^k$
1 – *Decrease the desired unclipped quantile*;
2 $C_j = UC_{t-1} \cdot (1 - r)^t$;
3 – *Injecting noise into the models gradient*;
4 $\hat{c}_t^k = c_t^k / max(1, \frac{||c_t^k||2}{S})$;
5 –*Send the noisy gradient to the central server.*;

Subsequently, we introduced noise to the compressed gradient denoted as c_t^k, using the computed noise factor S. To implement the noise mechanism, we utilized a Gaussian mechanism. This procedure yielded a compressed gradient with the incorporated noise, represented as $\hat{c}_t^k$, as illustrated in Algorithm 3 in line 4. Following this, the gradient was transmitted to the server for further steps, including aggregation, inversion, and the updating of the global model.

G. AGGREGATION

During this phase, the server received the compressed noisy gradient $\hat{c}_t^k$. Subsequently, the server conducted gradient aggregation, following the procedure outlined in lines 14-16 of Algorithm 4. The aggregation process produces the updated parameter, denoted as w_t, representing the parameter values at a specific iteration or time step during the optimization process. These updated parameters are obtained by aggregating $\hat{c}_t^k$ from participating clients. The parameter w_t will be employed for the inverse Discrete Fourier Transform (DFT) aggregator.

H. INVERSE DFT AGGREGATOR

This process was the inverse of the DFT aggregator; the step aimed to restore the original gradient transformed in the DFT aggregator process. The detailed steps of this process can be described as follows:

Algorithm 4 Inverse DFT and Model Update

Data: w_t
Result: o_w
1 **Function** `ServerTrans` (w_t) **:**
  Data: o_c, o_i, o_{ir}
  Result: o_w
2 **foreach** *element* o *in* w_t **do**
3 $o_s \leftarrow o * \sqrt{\frac{2}{d}}$
4 **end**
5 **foreach** *element* g *in* o_s **do**
6 $o_r \leftarrow$ Reshape$(g, [-1])$
7 **end**
8 $o_c \leftarrow$ Complex$(real = o_r[0], imag = o_r[1])$
9 $o_i \leftarrow$ IFFT(o_c)
10 $o_{ir} \leftarrow$ InverseR(o_i)
11 $o_w \leftarrow$ Concat$(real(o_{ir}), imag(o_{ir}))$ **return** o_w;
12
13 – Server Aggregation
14 **for** $k \in k_1, k_2, \ldots, k_n, K$ **do**
15 $w_t \leftarrow \sum_{t=1}^{T} \frac{n_k}{n} c_t^k$;
16 **end**
17 – Inverse transformation
18 $o_w \leftarrow$ `ServerTrans` (w_t) ;
19 – Global model update
20 $W_t \leftarrow W_{t-1} - n\nabla g(o_w)$;

1) **Scaling**: This step starts by reversing the scaling step, which involves multiplying each element in the gradient w_t by a scaling factor. In this case, it multiplies the tensor by $\frac{1}{\sqrt{\frac{d}{2}}}$ to normalize it. This process can be seen in the Algorithm 4 line 2-4.
2) **Reshaping**: This step reshapes the scaled gradient o_s into a two-row matrix with a flexible number of columns as seen in Algorithm 4 line 5-7. This process effectively reverses the flattening operation in the DFT aggregator process.
3) **Complex Packing**: this step repacks the real-valued gradient o_r into a complex data type, reversing the operation of unpacking it into real and imaginary components.
4) **Inverse Fourier Transform (IFFT)**: This applies the inverse Fourier transform (IFFT) to convert the complex gradient o_c from the frequency domain back to the time or spatial domain, effectively reversing the forward Fourier transform. This process can be seen in Algorithm 4 line 9.
5) **Random Inverse Rotation**: This process reverses the random rotations introduced in the DFT aggregator process in the client by applying the inverse rotation operation to o_i. This random inverse rotation has the same parameter as the previous one used in random rotation. To reverse the random rotations, it typically needs access to the same random rotation parameters used in the forward transformation. Without knowledge of these parameters, accurately inverting the rotations is challenging or impossible. Implementing random rotation ensures that sensitive information remains protected. This process can be seen in Algorithm 4 line 10.
6) **Real-Imaginary Concatenation**: This process concatenates the real and imaginary components into a complex gradient o_w as defined in Algorithm 4 line 11, effectively reversing the separation of real and imaginary parts.

I. UPDATE GLOBAL MODEL

This phase commences following the inverse operation, which calculates the reverse gradient of o_w. This inverted gradient is then utilized to update the global model W, as indicated in Algorithm 4 at line 20. This process generated an updated global model W_t. Subsequently, this updated global model W_t is distributed to all participating clients and will be used for the next iteration in the learning process. The detailed algorithm of our proposed method can be found in Algorithm 5.

Algorithm 5 Overall Algorithm

1 **Program** *SERVER*
2 - Initialization of global model w_0
3 **for** $t|t_1|t_2|t_n \ldots..T$ **do**
4 - Inverse Transform and Model Update
5 $W_t \leftarrow$ **Algorithm 4**;
6 **end**
7 **return** cpr, nm, w_0, W_t;
8 **end**
9 **Program** *CLIENT*
10 – Privacy Loss Calculation
11 $S \leftarrow$ **Algorithm 1**
12 $w_t^k \leftarrow$ Local Training
13 – Gradient Transformation with DFT Aggregator
14 $c_t^k \leftarrow$ **Algorithm 2**
15 – Adaptive Gaussian Clipping DP
16 $\hat{c}_t^k \leftarrow$ **Algorithm 3**
17 **return** $\hat{c}_t^k$
18 **end**

V. DATASETS

We tested our proposed framework on two datasets: the PIMA Indian Diabetes dataset and Breast Cancer Histopathology images. These two datasets are of different sizes and have distinct characteristics, which are described below:

1) **PIMA Indian Diabetes Dataset**[1]. The Pima Indian Diabetes Dataset, originally obtained from the National Institute of Diabetes and Digestive and Kidney Diseases, provides information about 768 women living near Phoenix, Arizona, USA. This dataset includes data about eight factors believed to influence diabetes, along with the corresponding classifications. It is organized into 9 columns and 768 rows, with 500 entries representing non-diabetic individuals and 268 representing diabetic individuals. The classification outcome variable is binary, using 0 (indicating a negative diabetes test) and 1 (indicating a positive diabetes test).
2) **Breast Cancer Histopathology (BCH)**[2]. The dataset comprises 162 whole-mount slide images of Breast Cancer (BCa) specimens scanned at 40x magnification. From these images, a total of 277,524 patches measuring 50 × 50 were extracted, with 198,738 being IDC negative and 78,786 being IDC positive.

VI. MODELS

For each client, we used a local convolutional neural network model. We also applied a convolutional neural network for the Global model. Different models are used in the PIMA Indian Diabetes Dataset and Breast Cancer Histopathology. Below is the description of the model used in this paper.

1) **Model for PIMA Indian Diabetes Dataset**: The model used for this type of dataset was a feedforward neural network comprised of an input layer and two hidden layers with 6 and 3 neurons, respectively, all using sigmoid activation functions, as well as an output layer with a single neuron and sigmoid activation. The Glorot Normal weight initialization method was employed for all layers. Notably, the output layer incorporates L1 and L2 regularization with specific strengths ($l1 = 0.0001$ and $l2 = 0.01$).
2) **Model for Breast Cancer Histopathology Dataset**: The model for this dataset consisted of four convolutional layers with 32, 64, 128, and 128 filters of size 3×3, respectively, each followed by a rectified linear unit (ReLU) activation function and max-pooling with a 2×2 pool size. Dropout layers with a rate of 0.25 were inserted after each max-pooling layer. Subsequently, a flattening layer converted the 2D feature maps into a 1D vector, followed by a fully connected dense layer with 128 units and ReLU activation. Another dropout layer with a rate of 0.5 was added to reduce overfitting further, and the final output layer consisted of two neurons with sigmoid activation.

We used the above model for the different evaluation scenarios and implemented it for all the state-of-the-art methods used as comparators, including the proposed method.

[1]https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database
[2]https://www.kaggle.com/datasets/paultimothymooney/breast-histopathology-images

TABLE 2. Detail of simulation hyperparameter.

Scenario	Parameter	Value
1	Client learning rate	0.1
	Server learning rate	1
	Server momentum	0.9
	bcpr	20
	Total Client	538
	bnm	1
2	Client learning rate	0.005
	Server learning rate	0.5
	Server momentum	0.5
	bcpr	50
	Total Client	2500
	bnm	0.5

VII. SIMULATIONS

A. SIMULATION ENVIRONMENT

The simulation was conducted on a computer equipped with an 11th Generation Intel Core i9-11900K CPU, 128 GB of RAM, and an NVIDIA GTX 3070 graphics card with 8 GB of VRAM. The computer ran the Ubuntu 20.04 LTS operating system. The simulation platform included Python 3.8 and Tensorflow 2.8.3, with Tensorflow Federated 0.22.0 serving as the framework for Federated Learning (FL), and Tensorflow Privacy 0.8.0 being utilized as the privacy library.

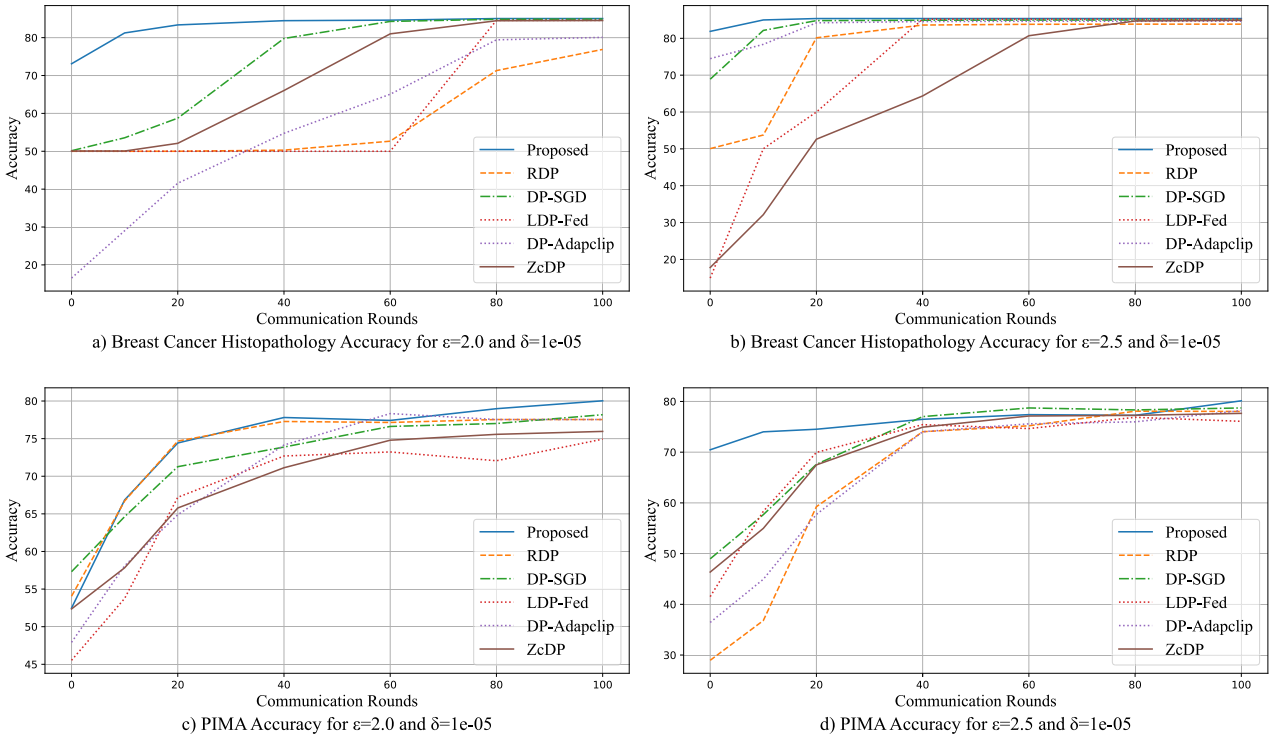
B. SIMULATION SCENARIOS

In our evaluation, we utilized a simulation to compare our proposed method with the other algorithms previously employed in Federated Learning (FL), including DP-SGD, RDP, ZcDP, LDP-Fed, and DP-AdapClip. We assessed the differences among the five algorithms and our proposed approach in terms of model accuracy, total communication costs, and computational complexity. Our simulation process encompassed two distinct scenarios to provide a comprehensive analysis.

- **Scenario 1**. We evaluate our proposed method and five other algorithms using different datasets. For the privacy budget, we set $\epsilon = 2.0$ and $\delta =$ 1e-05. For the dataset, we used the PIMA Indian Diabetes dataset and Breast Cancer Histopathology datasets. We iterated the scenario for 100 communication rounds.
- **Scenario 2**. We evaluate our proposed method and five other algorithms using the same datasets as scenario 1. The difference lay in the privacy budget used. In this scenario, we used $\epsilon = 2.5$ and $\delta =$ 1e-05. We also iterated this scenario using 100 communication rounds. The detailed simulation hyperparameters are listed in Table.2.

VIII. RESULTS AND DISCUSSION

In this section, we present and discuss the results obtained from the simulation process. In particular, we have engaged in an analysis of the results in terms of accuracy, total communication costs, and computational complexity.



a) Breast Cancer Histopathology Accuracy for ε=2.0 and δ=1e-05

b) Breast Cancer Histopathology Accuracy for ε=2.5 and δ=1e-05

c) PIMA Accuracy for ε=2.0 and δ=1e-05

d) PIMA Accuracy for ε=2.5 and δ=1e-05

FIGURE 2. Global model accuracy of proposed method compare to other DP algorithm.

TABLE 3. Detail of simulation results on model's accuracy.

No	Dataset	Eps	Delta	Proposed	RDP	ZcDP	LDP-Fed	DP-AdapClip	DP-SGD
1	PIMA Indian Diabetes	2	1e-05	**80.03**	77.54	75.96	74.93	77.54	78.18
		2.5	1e-05	**80.13**	78.03	77.65	76.08	78.2	78.72
2	Breast Cancer Histophatology	2	1e-05	**85.04**	76.89	84.52	84.56	80.07	84.9
		2.5	1e-05	**85.36**	83.84	84.88	85.2	84.64	84.96

A. MODEL ACCURACY

In Fig.2, we can see that our method achieves higher model accuracy compared to other methods in both scenarios 1 and 2. While the accuracy improvement in the Breast Cancer Histopathology datasets may seem modest, it is notable that our approach outperforms all state-of-the-art methods in the PIMA Indian Diabetes datasets. Furthermore, the findings indicate that our method achieves faster model convergence than all state-of-the-art methods in the Breast Cancer Histopathology datasets, regardless of whether the value of ϵ is higher or lower. Specifically, our method reaches convergence within 20 communication rounds. In contrast, the other state-of-the-art methods require 40 communication rounds to converge, except for DP-SGD at $\epsilon = 2.5$, which converges in the same time frame as our proposed method.

B. TOTAL COMMUNICATION COSTS

We also evaluated the total communication costs incurred by each method. To assess this, we introduced the Aggregated Data (AD) variable, which represents the tensors returned by each client to the server (upstream data) after local training. The size of this parameter was measured in Gigabytes (GB) for the Breast Cancer Histopathology dataset and Kilobytes (KB) for the PIMA Indian Diabetes dataset. Fig.4 shows that our proposed method results in significantly lower total communication costs compared to other state-of-the-art methods in both scenarios for the Breast Cancer Histopathology and PIMA Indian Diabetes datasets. Additionally, for the PIMA Indian Diabetes datasets, our method's total communication costs do not increase substantially beyond 80 communication rounds. In contrast, methods like RDP, DP-SGD, and ZcDP consistently show increasing communication costs with each round. Although LDP-Fed and DP-AdapClip initially have lower total communication costs than our method, they exhibit exponential increases beyond 80 rounds, eventually surpassing our proposed method.

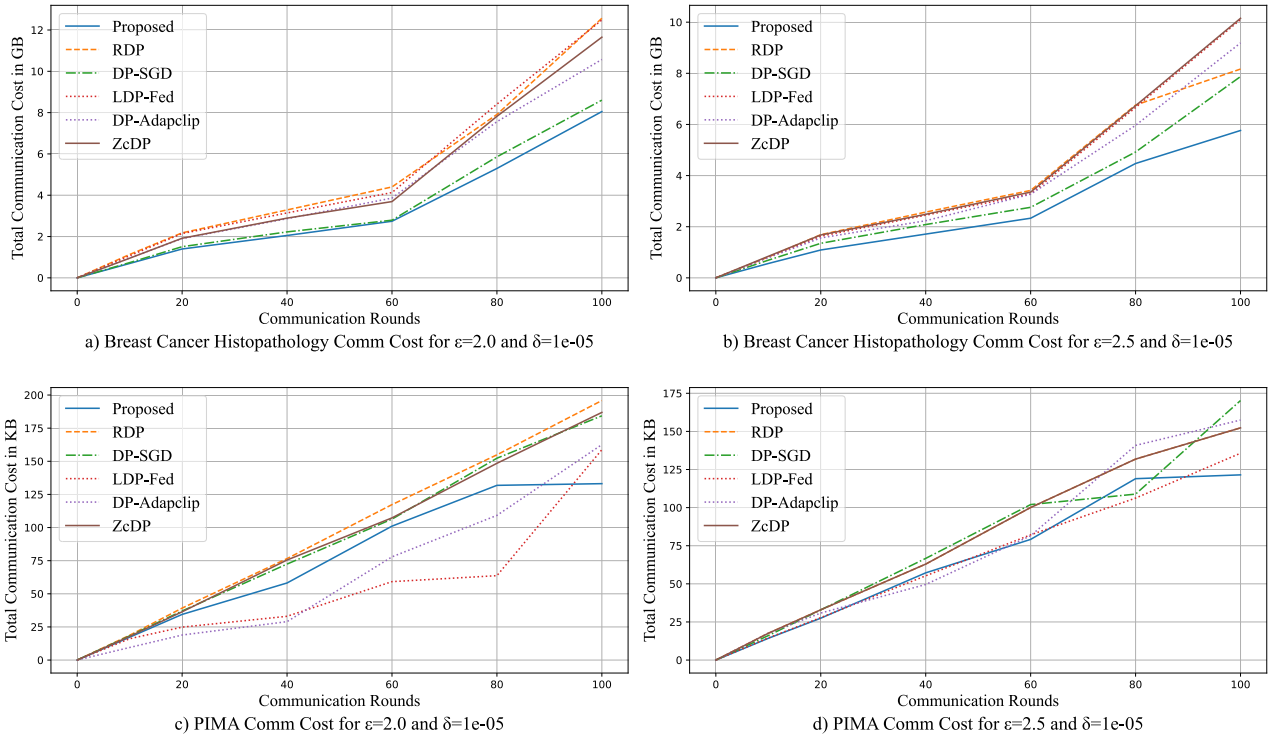
Based on Table 4, our proposed method has a communication efficiency advantage compared to another state-of-the-art method, even when evaluated using more extensive datasets and complex models in additional relation to different privacy parameters. We also observed an increase in total

TABLE 4. Detail of simulation results on total communication cost.

No	Dataset	Eps	Delta	Proposed	RDP	ZcDP	LDP-Fed	DP-AdapClip	DP-SGD
1	PIMA Indian Diabetes (In KB)	2	1e-05	**133.12**	195.84	186.88	158.72	162.56	184.32
		2.5	1e-05	**121.49**	152.32	152.32	135.68	157.44	170.24
2	Breast Cancer Histopathology (In GB)	2	1e-05	**8.05**	12.57	11.65	12.48	10.57	8.60
		2.5	1e-05	**5.76**	8.17	10.15	10.10	9.19	7.87

TABLE 5. Detail of simulation results on total training time in (Seconds).

No	Dataset	Eps	Delta	Proposed	RDP	ZcDP	LDP-Fed	DP-AdapClip	DP-SGD
1	PIMA Indian Diabetes	2	1e-05	**72**	452	475	310	509	468
		2.5	1e-05	**67**	364	305	240	413	420
2	Breast Cancer Histophatology	2	1e-05	**1971**	4683	4080	4045	3936	3803
		2.5	1e-05	**1889**	4353	3573	3515	3173	2739



a) Breast Cancer Histopathology Comm Cost for ε=2.0 and δ=1e-05

b) Breast Cancer Histopathology Comm Cost for ε=2.5 and δ=1e-05

c) PIMA Comm Cost for ε=2.0 and δ=1e-05

d) PIMA Comm Cost for ε=2.5 and δ=1e-05

FIGURE 3. Total communication cost of proposed method compared to other DP algorithms.

communication costs when there was a decrease in ϵ. This increase occurred because as ϵ decreases, more noise is added to the model gradient, resulting in a more biased model. To address this challenge, as noted in [6], each algorithm increases the number of clients participating in the learning process, which in turn raises total communication costs. Notably, DP-SGD had the lowest increase in total communication costs, indicating that it did not significantly increase the number of participating clients in the learning process in order to meet a lower privacy budget of ϵ and δ, unlike other methods.

C. COMPUTATIONAL COMPLEXITY

We measured computational complexity based on the observed training time for all methods, with lower training times indicating lower computational complexity and faster convergence. This has significant real-world implications for resource efficiency, responsiveness, and feasibility in privacy-preserving federated learning applications. Our proposed method demonstrated consistently lower training times than other state-of-the-art methods for both the Breast Cancer Histopathology and PIMA Indian Diabetes datasets in Scenarios 1 and 2, as shown in Fig.3. For the PIMA Diabetes dataset at $\epsilon = 2.0$, our method achieved substantial reductions in training time compared to other methods. Even at $\epsilon = 2.5$, our method maintained its efficiency advantage. Similarly, for the Breast Cancer Histopathology dataset, our method exhibited significant training time efficiency for both ϵ values. Our proposed method achieved this efficiency, even with additional time added, because of the DFT transform

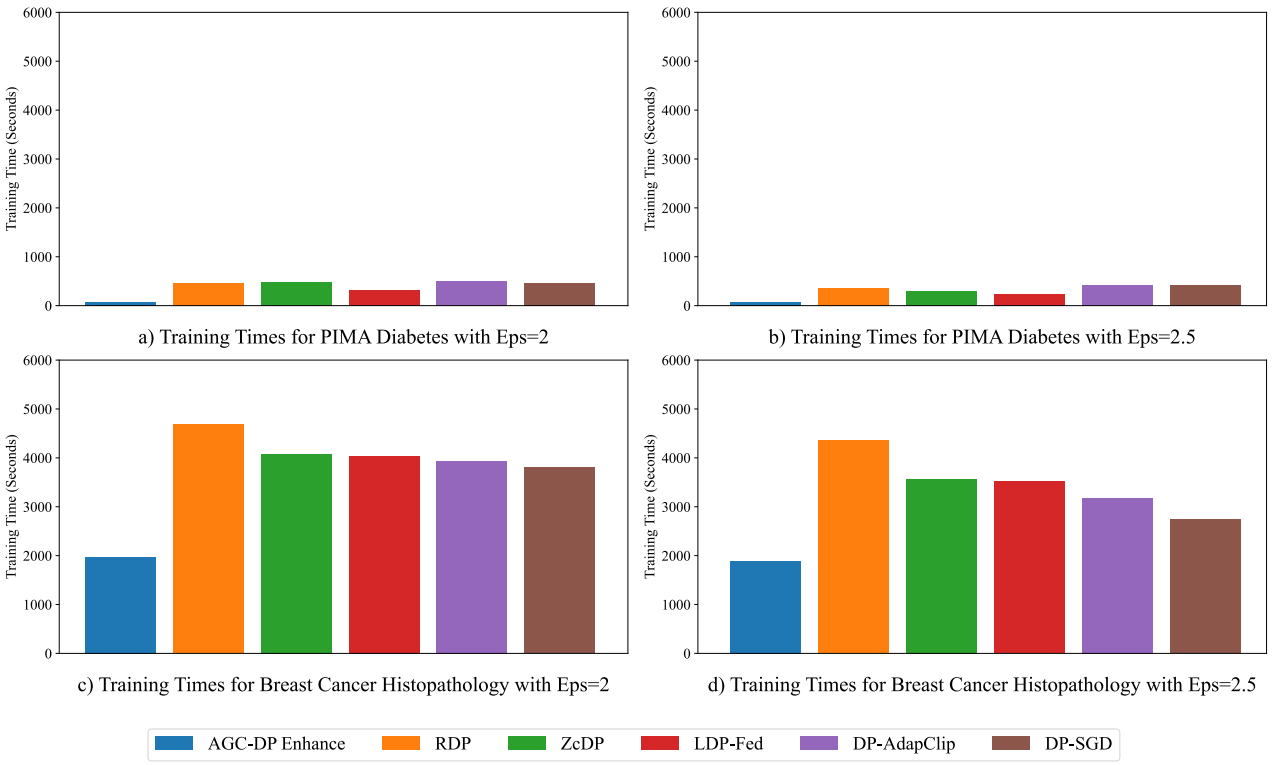


FIGURE 4. Total training time of proposed method compared to other DP algorithms.

aggregator in the client and inverse process in the server. The detailed results of the total training time can be seen in Table 5.

D. DISCUSSION

The simulation results show that accuracy decreases as the epsilon value decreases from 2.5 to 2.0 across both datasets due to increased noise making the model more biased. Despite this, our proposed method demonstrates remarkable resilience to stricter privacy constraints, exhibiting only minimal decreases in accuracy. In contrast, other methods show more pronounced accuracy decreases. For example, the RDP method shows a substantial drop in accuracy in both datasets. Similarly, ZcDP, LDP-Fed, and DP-AdapClip also experience significant decreases, indicating varying vulnerability to stricter privacy budgets. Our proposed method and DP-SGD exhibit minimal decreases in accuracy as epsilon decreases. While DP-SGD performs slightly better in one dataset, our method performs better in the other. Overall, DP-SGD demonstrates greater resilience to accuracy decreases with lower epsilon values in larger and more complex datasets, such as CT scan images, compared to all other methods, including our proposed method.

In our study, we carefully selected two privacy budgets, $\epsilon = 2.0$ and $\epsilon = 2.5$, with a fixed $\delta =$ 1e-05, as the foundation for our simulations. This deliberate choice allowed us to assess the performance of each algorithm under different privacy budgets, demonstrating the adaptability and flexibility of our proposed method across various settings and environments. Furthermore, we extended our evaluation to include practical healthcare datasets. This step was essential to showcase the real-world applicability of our proposed system, particularly within the healthcare sector. Given the prevalence of resource-constrained devices in healthcare applications, in terms of both computational power and communication capabilities, we recognized the need to evaluate the total communication costs generated by our method and previous approaches. Our evaluation results provide valuable insights into the communication burdens imposed by our proposed method and its impact on network bandwidth in real-world environments compared to prior methods. This insight is invaluable for understanding the practical implications and scalability of our approach within the broader context of privacy-preserving federated learning in healthcare.

IX. CONCLUSION AND FUTURE WORKS

This paper outlines a novel technique aimed at enhancing the privacy measures of Differential Privacy (DP) within the context of Federated Learning (FL) for healthcare. Our approach integrates a DFT Aggregator and leverages adaptive Gaussian clipping. To gauge the effectiveness of our proposed system, we conducted evaluations using two healthcare datasets: PIMA Indian Diabetes and Breast Cancer Histopathology, each operating under different privacy budget constraints.

Our simulation results demonstrate that our method surpasses the traditional DP algorithms typically employed in FL, achieving heightened accuracy. Notably, this enhanced accuracy is accomplished while reducing total communication costs and training time. Through our proposed method, we enhance patient confidentiality and data security, which are crucial in modern healthcare. Additionally, our method's increased efficiency can streamline processes, improving patient outcomes and resource allocation. In future work, we aim to optimize hyperparameters and refine our algorithm to enable implementation on real devices and for use with more complex and larger datasets.

REFERENCES

[1] Z. A. E. Houda, A. S. Hafid, L. Khoukhi, and B. Brik, "When collaborative federated learning meets blockchain to preserve privacy in healthcare," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 5, pp. 2455–2465, Nov. 2023.

[2] R. Gosselin, L. Vieu, F. Loukil, and A. Benoit, "Privacy and security in federated learning: A survey," *Appl. Sci.*, vol. 12, no. 19, p. 9901, Oct. 2022.

[3] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*. San Francisco, CA, USA: IEEE, May 2019, pp. 691–706.

[4] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1333–1345, May 2018.

[5] O. Zari, C. Xu, and G. Neglia, "Efficient passive membership inference attack in federated learning," 2021, *arXiv:2111.00430*.

[6] M. A. Hidayat, Y. Nakamura, B. Dawton, and Y. Arakawa, "AGC-DP: Differential privacy with adaptive Gaussian clipping for federated learning," in *Proc. 24th IEEE Int. Conf. Mobile Data Manage. (MDM)*, Jul. 2023, pp. 199–208.

[7] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Gener. Comput. Syst.*, vol. 115, pp. 619–640, Feb. 2021.

[8] L. Zhang, J. Xu, P. Vijayakumar, P. K. Sharma, and U. Ghosh, "Homomorphic encryption-based privacy-preserving federated learning in IoT-enabled healthcare system," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 5, pp. 1–17, 2022.

[9] Z. Yang, Y. Shi, Y. Zhou, Z. Wang, and K. Yang, "Trustworthy federated learning via blockchain," *IEEE Internet Things J.*, vol. 10, no. 1, pp. 92–109, Jan. 2023.

[10] S. Wu, M. Yu, M. A. M. Ahmed, Y. Qian, and Y. Tao, "FL-MAC-RDP: Federated learning over multiple access channels with Rényi differential privacy," *Int. J. Theor. Phys.*, vol. 60, no. 7, pp. 2668–2682, Jul. 2021.

[11] M. Abadi, "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, Vienna Austria, 2016, pp. 308–318.

[12] R. Hu, Y. Guo, and Y. Gong, "Concentrated differentially private federated learning with performance analysis," *IEEE Open J. Comput. Soc.*, vol. 2, pp. 276–289, 2021.

[13] S. Truex, L. Liu, K.-H. Chow, M. E. Gursoy, and W. Wei, "LDP-fed: Federated learning with local differential privacy," in *Proc. 3rd ACM Int. Workshop Edge Syst., Analytics Netw.* New York, NY, USA: Association for Computing Machinery, Apr. 2020, pp. 61–66.

[14] G. Andrew, O. Thakkar, B. McMahan, and S. Ramaswamy, "Differentially private learning with adaptive clipping," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 17455–17466.

[15] M. A. Hidayat, Y. Nakamura, and Y. Arakawa, "Privacy-preserving federated learning with resource-adaptive compression for edge devices," *IEEE Internet Things J.*, vol. 11, no. 8, pp. 13180–13198, Apr. 2024.

[16] H. Ku, W. Susilo, Y. Zhang, W. Liu, and M. Zhang, "Privacy-preserving federated learning in medical diagnosis with homomorphic re-encryption," *Comput. Standards Interfaces*, vol. 80, Mar. 2022, Art. no. 103583.

[17] B. Wang, H. Li, Y. Guo, and J. Wang, "PPFLHE: A privacy-preserving federated learning scheme with homomorphic encryption for healthcare data," *Appl. Soft Comput.*, vol. 146, Oct. 2023, Art. no. 110677.

[18] F. Wibawa, F. O. Catak, M. Kuzlu, S. Sarp, and U. Cali, "Homomorphic encryption and federated learning based privacy-preserving CNN training: COVID-19 detection use-case," in *Proc. Eur. Interdiscipl. Cybersecurity Conf.* New York, NY, USA: Association for Computing Machinery, Jun. 2022, pp. 85–90.

[19] L. Ngan Van, A. H. Tuan, D. Phan The, T.-K. Vo, and V.-H. Pham, "A privacy-preserving approach for building learning models in smart healthcare using blockchain and federated learning," in *Proc. 11th Int. Symp. Inf. Commun. Technol.* New York, NY, USA: Association for Computing Machinery, Dec. 2022, pp. 435–441.

[20] M. Abaoud, M. A. Almuqrin, and M. F. Khan, "Advancing federated learning through novel mechanism for privacy preservation in healthcare applications," *IEEE Access*, vol. 11, pp. 83562–83579, 2023.

MUHAMMAD AYAT HIDAYAT (Graduate Student Member, IEEE) was born in 1991. He received the bachelor's degree in informatics engineering from Pasundan University, in 2015, and the M.Eng. degree in computer engineering from Bandung Institute of Technology, in 2018. He is currently pursuing the Ph.D. degree with the Graduate School of Information Science and Electrical Engineering, Kyushu University, Japan. His research interests include security and privacy in distributed learning and sensors and embedded systems

YUGO NAKAMURA (Member, IEEE) was born in 1992. He received the B.E. degree from the Advanced Course of Production System Engineering, National Institute of Technology, Hakodate College, Japan, in 2015, and the M.E. and Ph.D. degrees from the Graduate School of Information Science, Nara Institute of Science and Technology, Japan, in 2017 and 2020, respectively. He is currently an Assistant Professor with the Graduate School and the Faculty of Information Science and Electrical Engineering, Kyushu University. His current research interests include the Internet of Things, ubiquitous computing, and human–computer interaction. He is currently a member of ACM and IPSJ.

YUTAKA ARAKAWA (Member, IEEE) received the B.E., M.E., and Ph.D. degrees from Keio University, Japan, in 2001, 2003, and 2006, respectively. He is currently a Professor with the Graduate School and the Faculty of Information Science and Electrical Engineering, Kyushu University. He also an Invited Professor with Osaka University. His current research interests include human activity recognition, behavior change support systems, and location-based information systems. He has received the 2023 Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science and Technology, PerCom 2019 Best Demonstration Award, the IPSJ/IEEE-CS Young Scientist Award, the Ubicomp/ISWC 2016 Best Demo Award, and more than 35 other awards.

Privacy-Preserving Federated Learning in Healthcare, E-Commerce, and Finance: A Taxonomy of Security Threats and Mitigation Strategies

Rahul Kumar[1, 2*], Chin-Shiuh Shieh[2], *Prasun Chakrabarti[1] ,Ashok Kumar[3], Jhankar Moolchandani[4] and Raj Sinha[5]*

[1]Sir Padampat Singhania University, Udaipur, Rajasthan, India
[2]Department of Electronic Engineering, National Kaohsiung University of Science and Technology, Taiwan
[3,4]Department of Computer Science and Engineering, Amity School of Engineering and Technology, Amity University, Madhya Pradesh, Gwalior, India
[5]Faculty of IT & CS (FITCS),Parul University, Vadodara,Gujarat ,India

Abstract. Federated Learning (FL) transformed decentralized machine learning by allowing joint model training without mutually sharing raw data, hence being especially useful in privacy-sensitive applications like healthcare, e-commerce, and finance. Even with its privacy-focused architecture, FL is vulnerable to a range of security attacks such as data poisoning, model inversion, membership inference attacks, and communication interception. These attacks compromise the confidentiality of patients in healthcare, consumer data privacy in e-commerce, and financial safety in banking, thus necessitating effective privacy-preserving mechanisms. This survey presents a classification of security threats in FL, grouping them by their source, effect, and attack mode. We review state-of-the-art countermeasures, such as differential privacy, secure multi-party computation, homomorphic encryption, and resilient aggregation methods, their effectiveness, trade-offs, and real-world applicability to FL. In medicine, FL enables joint disease diagnosis without compromising patient confidentiality; in online shopping, it provides personalized suggestions without revealing customer tastes; and in banking, it improves fraud detection without violating regulatory requirements. In addition, we discuss future horizons in privacy-preserving FL, including adversarial robustness, blockchain-protected models, and tailored FL architectures, improving security and resiliency in these domains. We also discuss the balancing problems between security, accuracy, and computational efficiency with possible trade-offs in scaling privacy-preserving FL By analyzing threats and mitigation strategies systematically, this paper will provide direction to future research on designing secure, scalable, and privacy-preserving FL frameworks for the changing healthcare, e-commerce, and finance needs.

1 Introduction

The tremendous leap of artificial intelligence (AI) and machine learning (ML) [1] has been applied ubiquitously across diversified domains such as healthcare, finance, and IoT-enabled environments. Traditional centralized approaches to ML require collecting copious amounts of data obtained from distributed sources into a central server, raising an avalanche of

* Corresponding author: rahul.cse397@gmail.com

concerns on data privacy, security, and regulatory compliance. Federated learning has emerged as one of the promising decentralized paradigms that allow multiple clients, such as mobile devices, edge nodes, and institutions, to collaboratively train a shared global model without sharing raw data directly [1]. By keeping the data localized and only transmitting model updates, federated learning mitigates much privacy risks while also allowing organizations to apply distributed data for better learning outcomes. Such a privacy-preserving design has made FL an essential technology for conducting AI applications in highly privacy-sensitive domains [2].

Although it has many advantages, FL is not inherently secure and continues to suffer from various privacy and security threats [2]. Malicious participants or adversaries exploit FL's decentralization to attack it with data poisoning attacks, model inversion attacks, and membership inference attacks on sensitive information. Furthermore, since communication between clients and the central server is possible, FL suffers from eavesdropping attacks, model tampering, among other adversarial manipulations. Therefore, the issues related to privacy and robustness can only be guaranteed through more sophisticated security mechanisms such as differential privacy, secure multi-party computation, and homomorphic encryption. But still, achieving an optimal balance between security, computational efficiency, and model accuracy remains a significant challenge [2].

Privacy-preserving FL [3] focuses on securing the data to provide improved safety along with an effective collaborative learning procedure. While the FL protocol gains momentum with widespread applications in reality, dealing with security-related threats is quite vital for fostering confidence and guaranteeing its implementation [3]. Current works involve the establishment of privacy-preserving mechanisms with limited threat but non-significant degradations to the models' performance. However, the varied nature of security threats, from data leakage to adversarial model manipulations, calls for a thorough understanding of FL's attack surface and corresponding countermeasures.

In addition to that, this review paper presents cutting-edge privacy-preserving strategies and attempts to give a systematic overview of security issues in federated learning [4]. The paper will study new developments in safe architectures for FL, evaluate mitigation techniques of existing works, and classify different attacks according to their impact and attack paths. This paper further goes on to talk about the primary challenges, trade-offs, and future research direction in privacy-preserving FL. This work shall be a handy resource for security threats and their countermeasures, which should be of prime interest to those researchers and practitioners who are willing to enhance privacy, security, and reliability within federated learning systems [4].

2 Federated Learning Concept

Federated learning [5] is a type of decentralized machine learning where many devices or organizations can train a global shared model without revealing the raw data they use. This is different from other paradigms of machine learning because most of them have approaches that usually require centralizing data in a single repository before training. However, in federated learning, the data stays on local devices. Rather than sending data back to a central server, this model shares with the central aggregator only the locally computed model updates, such as gradient changes or model parameters. The central server takes up all these updates, combines them together, and hence improves the global model. The process iterates over multiple rounds until convergence of the model. FL has attracted a great deal of attention within privacy-sensitive applications mainly because it has provided support to security-related concerns about data security, confidentiality, and regulation compliance with the capability for massive-scale collaborative learning [5].

The basic operating principles of FL [6] are those of distributed training models and secure communication between the clients participating in it and the central server. The protocol usually starts by initializing the global model by the central server and then distributing the same to all the participants. Each client uses its private dataset to train the model locally for a certain number of iterations. After the local training, the updates of the model instead of the actual data are transferred to the central server by the clients [6]. These updates are aggregated by the server; in the most common instance, these aggregate operations are handled by algorithms of the type that averages the received model parameters based on the local computations of distributed clients. An aggregated model thus again gets relayed to these clients for local training for an additional round before such cycles again lead to distributed learning with only private data used at the various clients [7].

The three crucial constituents of the FL system are a central server, a communication mechanism, and a central client. Any Client can be a node of the FL network dispersed through an IoT device, a medical organization, or any other smartphone. The central server orchestrates the whole FL process in the sense of model initialization and collection of the local updates represented as gradients; it also delivers the resulting global model throughout the network. As FL is mostly decentralized about storage, the above conception of central server questions issues of trust and security since now this server would prove to be an attractive target for adversarial attacks. [8]

Communication mechanism aids in mutual transmission of model updates both from clients and the server. The role of effective and secure communication, therefore, is significant in FL to reduce the bandwidth consumption of the network with the preservation of model integrity. A large number of optimization techniques, such as update compression, quantization, and secure aggregation, are applied to reduce communication overhead with preserved model performance and security [9].

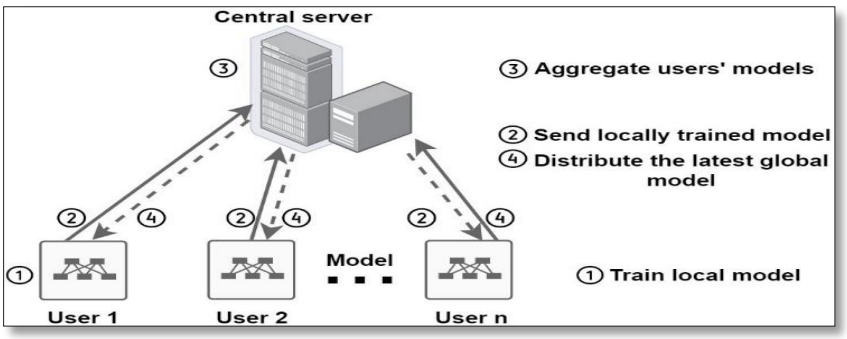


Fig. 1. Federate Learning system work flow[10]

As shown in Figure 1, A typical Federated Learning (FL) scheme operates through an iterative process involving the following steps, repeated until training is stopped:

- **Local Model Training**: Each FL user u_i trains its model M_i using its local dataset D_i
- **Model Uploading**: Each FL user ui uploads its locally trained model M_i to the central server S.
- **Model Aggregation**: The central server S collects and aggregates users' models to update the global model M.
- **Model Updating**: The central server S updates the global model M and distributes it to all FL users [10].

3 Security Threats Taxonomy of Federated Learning

Federated Learning has the advantages of protecting the data privacy in the decentralized model. However, new risks of security have also occurred. Summarily, four categories of attacks can be distinguished: privacy attacks, integrity threats, communication vulnerabilities, and governance issues. Each type counts to a different threat for confidentiality, model integrity, and hence for the trust within the system [11].

3.1 Privacy attacks

Federated Learning preserves decentralization control but keeps native privacy risk as exchange model updates continues. This means that attackers will analyze gradients so that they can infer sensitive training data, reconstruct images, text, or numerical records [12].

3.2 Integrity Threats

Integrity threats violate the correctness of FL models. The data poisoning attack includes label flipping, where training samples are mislabeled and backdoor attacks, where triggers are embedded that cause misclassification. It involves manipulating training data to deceive the learning process of the model[12].

3.3 Communication and Network Threats

FL relies on frequent communication between clients and servers, making it vulnerable to network-based threats. Eavesdropping attacks occur when the adversary intercepts model updates sent during transmission while extracting useful information about the training data. More advanced forms of adversaries implement MITM attacks, where attackers alter transmitted updates before forwarding them to the server and degrade model performance. The introduction of small strategic perturbations in updates leads to biased or inaccurate model output due to adversarial attacks in model transmission [12].

3.4 Trust and Governance Issues

Trust and governance challenges impact the fairness and reliability of FL systems. Free-riding occurs when clients participate without contributing meaningful updates, benefiting from the trained model without expending computational resources. Client-side adversarial behavior involves manipulating updates to gain an unfair advantage, such as biasing a recommendation system or bypassing fraud detection. Addressing these issues requires robust client selection mechanisms and trust validation strategies [13].

4 Methodology

This literature review is systematically structured with respect to recent breakthroughs about privacy-preserving federated learning (FL), specifically its security threats and mitigations. Methodology includes identifying literature, applying a multistage selection process, and performing an in-depth review of papers selected for the study.

4.1 Databases and Resources

Access to quality relevance in research was obtained through credible source retrieval of articles from:

- IEEE Xplore, SpringerLink , Elsevier (ScienceDirect) etc..

4.2 Inclusion/Exclusion criteria

The literature list was prepared on strict inclusion and exclusion criteria as follows:

Time Horizon: Only papers in the last ten years, 2019–2024, were considered since this will consider the latest trends. Old papers were only referenced if they provide the context or the foundation.

Relevance to Federated Learning: Include the paper only if it focuses on FL models, architectures, or applications. However, exclude a paper narrowly scoped on the centralized learning method.

Security and Privacy Focus. In this review, focus was placed on research that addresses security issues of leakage of data, adversarial attacks, model poisoning, and communication vulnerabilities.

Experimental Validation: Searches over abstract discussions that have been carrying out empirical analyses, simulations, or case studies on the feasibility of security solutions in FL.

4.3 Keywords and Search Strategy

Keywords along with Boolean operators were used in order to narrow the search through cross searching across various databases.

Keywords Searched for

- Federated Learning , Private Machine Learning, Secure Aggregation ,Differential Privacy in FL, Adversarial Attacks in FL , Privacy Preserving FL in Ecommerce , Privacy Preserving FL in Finance, Privacy Preserving FL in Healthcare

Then the keywords were combined with Boolean operators (AND, OR) in order to make the results come only from papers that are extremely relevant.

4.4 Selection Process

Selected literature was subject to a three-step process of progressive selection so as to get a structured filtering of the gathered literature in table 1 & figurer 2 :-

Stage One Collection: Through the found keywords, a comprehensive search gave 90 papers from 30 papers in each category of FL in Ecommerce, Finance and Healthcare.

Shortlisting: Titles and abstracts of all 90 papers were screened, resulting in the shortlisting of 55 papers. Papers that do not have proper discussions on FL security or those that are out of date are excluded.

Final Selection: Based on a closer analysis of the methodology, experimental results, and contribution, 29 papers were shortlisted for the detailed analysis.

Table 1. Selection Process of Literature Review

Stage	Number of Papers	Description

Initial Collection	90	Papers identified through database searches and citation tracking based on predefined keywords related to federated learning, security threats, and mitigation techniques.
Shortlisting	65	Papers reviewed based on inclusion criteria such as security aspects in FL, recent advancements, and proposed mitigation strategies.
Final Survey	29	In-depth review of the final selected papers, focusing on security threats, privacy-preserving techniques, and practical implementations in FL in Ecommerce, Finance and Healthcare.

In applying a funnel diagram, as exhibited in Figure 2, will show stepwise filtering to give the review sound structure and only focus on very impactful research relating to privacy-preservation federated learning.

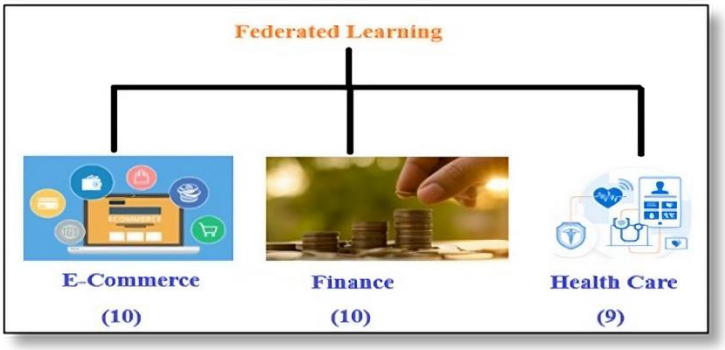


Fig. 2. Flow Diagram for Literature Review

5 Literature Review

5.1 Privacy Preservation Work in E-Commerce

Protecting online consumer privacy remains essential to security in e-commerce space because it builds customer trust while ensuring data protection. The protection of privacy in e-commerce requires two main approaches: minimizing collected data to essential information and providing clear consent procedures to users. The implementation of strong encryption protocols SSL/TLS together with secure payment methods based on tokenization delivers protection to sensitive data throughout its transfer process and storage phase. Organizations that implement 'Privacy by Design' design privacy mechanisms through all development stages to meet GDPR compliance requirements. Security audits and employee awareness programs along with routine assessments help build a safe digital shopping space which reinforces data protection for customers in figure 3 shows the architecture of FL for E-Commerce

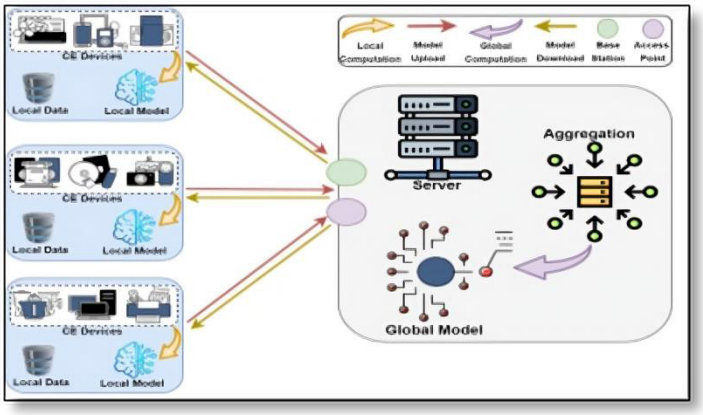


Fig. 3. Architecture of FL for E-Commerce [16]

Table 2. Findings in Privacy Preservation works in Ecommerce

Author Name (Year) [Ref]	**Main Concept**	**Findings**	**Limitations**
Ali, W., et al. (2025) [14]	Privacy-preserved and accountable recommender systems using FL, blockchain, and differential privacy	Categorized privacy threats, industrial demands, and technical solutions while providing an open-source benchmarking repository	Lacks real-time implementation details
Chen, S., & Huang, Y. (2025) [15]	FL-based airline upgrade optimization model	Enhanced prediction accuracy for upgrade invitations while preserving customer data privacy and mitigating data silos in the airline sector	Computational overhead compared to single-source models
Wu, J., et al. (2024) [16]	Federated deep learning-based recommender system for consumer electronics (FRS-CE)	Leveraged feature fusion, convolution operations, and adaptive aggregation for improved privacy, scalability, and recommendation accuracy	Needs further validation in different recommendation domains
Alqhatani, A., & Khan, S. B. (2024) [17]	IoT-based Hybrid Deep Collaborative Transformer (HDCT) with FL for e-commerce recommendations	Enhanced recommendation accuracy and error reduction using the Myntra fashion dataset	Optimization issues for larger datasets
Wei, P., et al. (2023) [18]	FedAds benchmark for privacy-preserving CVR estimation using vertical FL (vFL)	Provided standardized datasets and evaluation methods to improve privacy and performance in ad conversion prediction	Primarily applicable to advertising and not generalized for other domains

Liu, Z., et al. (2022) [19]	Survey on privacy-preserving aggregation (PPAgg) in FL	Analyzed PPAgg protocols, their advantages, limitations, and future research directions for improving privacy in FL systems	Lacks deployment in real-world FL applications
Wang, L.-e., et al. (2021) [20]	POI recommendation framework with FL and privacy preservation	Improved recommendation quality using auxiliary domain data and encrypted latent feature distribution	Needs large-scale dataset evaluation
Li, J., et al. (2021) [21]	Horizontal FL and ConvLSTM-based demand forecasting for e-commerce	Improved forecasting accuracy, reduced bullwhip effect, and ensured data privacy, promoting sustainable e-commerce growth	Computationally intensive compared to traditional models
Cheng, Y., et al. (2020) [22]	Federated learning for privacy-preserving AI	Addressed data silos and privacy regulations, demonstrating FL's potential for distributed model training	Did not assess FL efficiency on large-scale datasets
Kanagavelu, R., et al. (2020) [23]	Two-phase multi-party computation (MPC) for privacy-preserving FL	Improved scalability using an elected committee for model aggregation and implemented in IoT-based smart manufacturing	High communication overhead in peer-to-peer interactions

In table 2 gives the different research paper in E-Commerce and Finally, future research must focus on improving computational efficiency, correcting security vulnerabilities, designing standardized evaluation procedures, and using real-world applications of FL in different industries.

5.2 Privacy Preservation Work in Finance

Financial entities must preserve privacy because it protects customer-sensitive data and meets regulatory requirements. Financial institutions protect data during rest and movement with complex encryption technologies which minimize unauthorized access risks. Organizations achieve secure data analysis through techniques that both mask data and make information anonymous so private details remain protected. The emerging technologies of homomorphic encryption and secure multi-party computation allow protected data processing through encrypted data which enables collaborative analytics while preserving secret information. The secure environments enabled by confidential computing offer additional protection for processing sensitive information which improves overall data protection systems. in figure 4 shows the architecture of FL Finance with three basic key points.

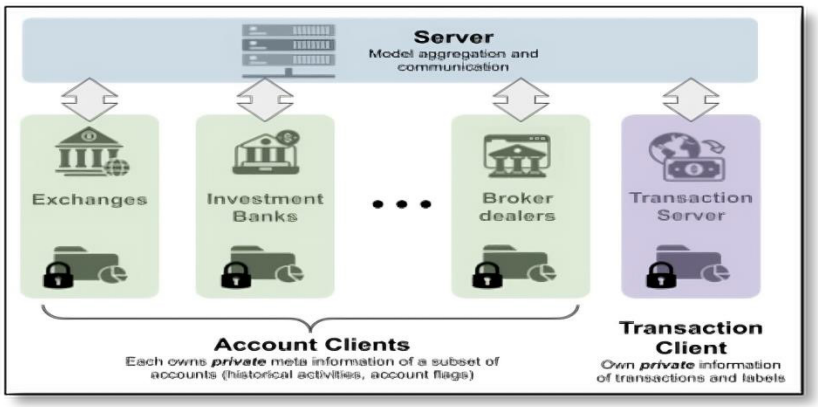


Fig. 4. The federated setup and three key participants of the proposed HyFL framework 1) A server responsible for information aggregation; 2) account clients: a set of financial institutions owning meta information of accounts; 3) A transaction client that owns private transactions and their labels (abnormal or not).[30]

Table 3. Findings in Privacy Preservation works in Finance

Author Name (Year)	Main Concept	Findings	Limitations
Abadi, A., et al. (2024) [24]	Starlit: Scalable FL mechanism for financial fraud detection	Improves scalability, accuracy, and security; addresses identity alignment and computational efficiency	Evaluated only on synthetic data; real-world applicability not tested
Haseeb, A., et al. (2024, November) [25]	FL framework with additive encryption for fraud detection	MLP achieves high accuracy (90-98%) on encrypted data	Focused only on MLP; lacks comparisons with other models
He, P., et al. (2024) [26]	DPFedBank: FL framework using Local Differential Privacy (LDP)	Enhances security through adaptive LDP, cryptographic methods, and authentication protocols	Does not address computational overhead introduced by LDP
Khan, M. S. I., et al. (2024) [27]	Fed-RD: FL algorithm for vertically and horizontally partitioned financial datasets	Ensures high model accuracy while preserving privacy with differential privacy and SMPC	Lacks real-world deployment validation
Kadhe, S. R., et al. (2023) [28]	PV4FAD: FL with homomorphic encryption (HE), SMPC, and differential privacy	Minimizes per-bank noise while maintaining high accuracy with a random forest ensemble model	Computationally expensive due to HE and SMPC

Arora, S., et al. (2023) [29]	FL with SMPC and differential privacy-driven noise aggregation	Improves anomaly detection AUPRC from 0.6 to 0.7 without data privacy loss	Lacks evaluation against adversarial attacks
Zhang, H., et al. (2023) [30]	Hybrid FL system for secure financial crime detection	Validated against privacy and security attacks in collaborative learning	Needs further real-world testing across multiple institutions
Kanamori, S., et al. (2022) [31]	Deepprotect: FL protocol for fraud detection in Japanese banks	Detects fraud beyond individual bank datasets while ensuring privacy compliance	Limited to five banks in Japan; generalizability is uncertain
Byrd, D., & Polychroniadou, A. (2020, October) [32]	FL combining differential privacy and SMPC for financial applications	Prevents data leakage while retaining model precision in credit card fraud detection	Tested only on logistic regression; lacks scalability to deep learning models
Li, Z., et al. (2020) [33]	Challenges and solutions in FL for privacy-preserving learning	Highlights mitigation techniques for privacy attacks and future research directions	Theoretical discussion; lacks experimental validation

In table 3 gives the different research paper in Finance and find the gap . Bridging such gaps by extensive real-world trials, effective encryption methods, adversarial robustness, and unified benchmarks will be crucial for making privacy-preserving FL more impactful in financial fraud detection.

5.3 Privacy Preservation Work in Healthcare

Healthcare must protect sensitive patient data as a top priority because it allows innovation to proceed in medical research and care delivery while preserving patient privacy. By applying federated learning and differential privacy methods institutions can jointly train AI models while maintaining data confidentiality through their raw data stays in private networks. The secure manipulation of encrypted data becomes possible with homomorphic encryption methods which boost privacy throughout the processing stage. Blockchain technology serves to prove data integrity while providing secure access control in decentralized healthcare structures. The combination of these methods provides healthcare systems a way to fulfill both data utility requirements and strict privacy standards.Iin figure 5 shows the architecture & application of FL for Healthcare.

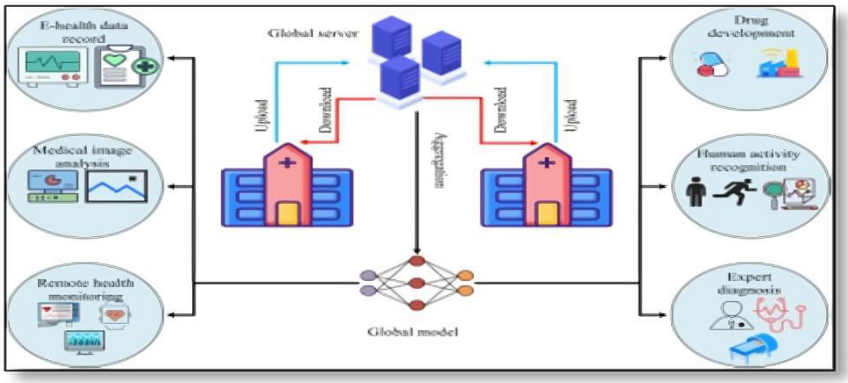


Fig. 5. Examples of federated learning applications in healthcare, including drug development, human activity recognition, remote health monitoring, electronic health data recording, medical image analysis, and assisted expert diagnosis. These are some of the most common use cases for FL in the healthcare field.[40]

Table 4. Findings in Privacy Preservation works in Healthcare

Author Name (Year) [Citation]	**Main Concept**	**Findings**	**Limitations**
Chaddad, Y., et al. (2024) [34]	FL in healthcare, comparing FL and centralized models.	Highlights FL's role in medical image analysis and behavior recognition, discusses evaluation metrics and future directions.	Does not address real-world implementation challenges in FL adoption.
Moon, S., & Lee, W. H. (2023) [35]	FL applications in healthcare, including COVID-19, tumor segmentation, and smart healthcare.	Identifies key FL use cases in healthcare and discusses privacy concerns and enhancement techniques.	Lacks empirical analysis on FL's effectiveness in these domains.
Ali, M., et al. (2023) [36]	Privacy preservation in IoMT healthcare networks using FL.	Introduces advanced FL architectures integrating deep reinforcement learning, digital twins, and GANs for detecting threats.	Does not provide real-world validation or implementation case studies.
Aouedi, O., et al. (2023) [37]	FL for medical data protection and its security vulnerabilities.	Reviews FL's role in protecting medical data, discusses security risks and privacy trade-offs.	Lacks experimental validation and practical implementation examples.
Firdaus, M., & Rhee, K.-H. (2023) [38]	Cross-Silo Federated Learning with Blockchain and Differential Privacy (CSFL-BDP).	Proposes a decentralized FL framework using blockchain and differential privacy for secure healthcare data sharing.	The scalability and computational cost of CSFL-BDP are not analyzed.
Islam, T. U., et al. (2022) [39]	Privacy-preserving FL with differential	Introduces a privacy-preserving FL model to	Needs evaluation on larger, more diverse

	privacy and feature selection.	predict heart failure and cancer diseases securely.	datasets for generalizability.
Narmadha, K., & Varalakshmi, P. (2022) [40]	Enhancing FL privacy in healthcare applications.	Explores FL's significance in preserving sensitive patient data and proposes an advanced privacy-preserving approach.	Does not address FL's communication overhead and efficiency.
Chowdhury, A., et al. (2021) [41]	FL applications in oncology and cancer research.	Reviews FL's use in decentralized cancer research while ensuring privacy and security.	Limited discussion on integrating FL with other AI models for cancer diagnosis.
Pfitzner, B., et al. (2021) [42]	Systematic review of FL in healthcare.	Evaluates FL's ability to maintain patient confidentiality while enabling collaborative training.	Lacks detailed discussion on regulatory challenges and compliance.

In table 4 gives the different research paper in Healthcare and find the gap. Another area that has not been thoroughly explored is interoperability between different healthcare systems and adherence to global privacy laws like GDPR and HIPAA. Computational overhead of FL is another crucial gap, as it creates difficulties for resource-limited medical IoT devices and edge computing use cases. In addition, the majority of the research focuses on medical imaging applications, and other domains like personalized treatment advice, genomics, and orphan disease studies remain comparatively underrepresented. Real-world pilot studies, privacy-preserving aggregation methods, and optimizing FL for heterogeneous and resource-constrained environments are the areas where future work needs to be directed to improve its use in real-world healthcare settings.

5.4 Roadmap for Future Research on Federated Learning in Healthcare, E-commerce, and Finance

Future federated learning (FL) research in healthcare, e-commerce, and finance must aim to surmount computational inefficiencies, privacy loopholes, and real-world scalability issues while maintaining cross-domain interoperability. In healthcare, FL can be integrated with blockchain, homomorphic encryption, and differential privacy to increase security while facilitating decentralized learning in hospitals and research institutions. For e-commerce, FL optimization for recommendation, fraud detection, and personal marketing needs to address adversarial robustness and dynamic user behavior. For finance, attention is needed for building scalable privacy-preserving FL frameworks that are able to defend against model inversion and poisoning attacks efficiently while complying with regulatory policies like GDPR and HIPAA. A common benchmarking framework for FL models across these domains, and real-world pilot experiments on heterogeneous datasets, will be critical to assess performance, standardize security measures, and maximize computational efficiency.

5.5 Methodology of proposed Future Research on Federated Learning in Healthcare, E-commerce, and Finance

Step	Action
1. Create Local Datasets	Random datasets simulating hospital, bank, e-commerce
2. Define Local Training	Train logistic regression separately at each client
3. Train Local Models	Each client trains its own model

4. Add Noise	(Optional) Apply Differential Privacy (tiny noise to coefficients)
5. Federated Averaging	Server averages coefficients/intercepts from all clients
6. Build Global Model	Assemble a global model from aggregated weights
7. Test & Evaluate	Test on new unseen data and print accuracy

Table 5:-Methodology of Future Research on Federated Learning in Healthcare, E-commerce, and Finance

- The system is based on Federated Learning (FL), a revolutionary machine learning approach where data never leaves the device or organization (like a hospital, bank, or e-commerce platform).
- Instead of sending sensitive raw data to a central server, each client (data holder) trains a local model using its private dataset and only sends encrypted or privacy-preserved model updates (like gradients or weights) to a central server.
- The server then aggregates these updates (e.g., by Federated Averaging) to create a global model that benefits from the knowledge of all participants — without ever seeing their private data.
- Additionally, privacy-enhancing technologies such as Homomorphic Encryption (HE), Differential Privacy (DP), and Secure Multi-Party Computation (SMPC) are used to further protect sensitive information.
- After multiple rounds of training and aggregation, a powerful and privacy-respecting global model is produced and deployed for real-world applications.

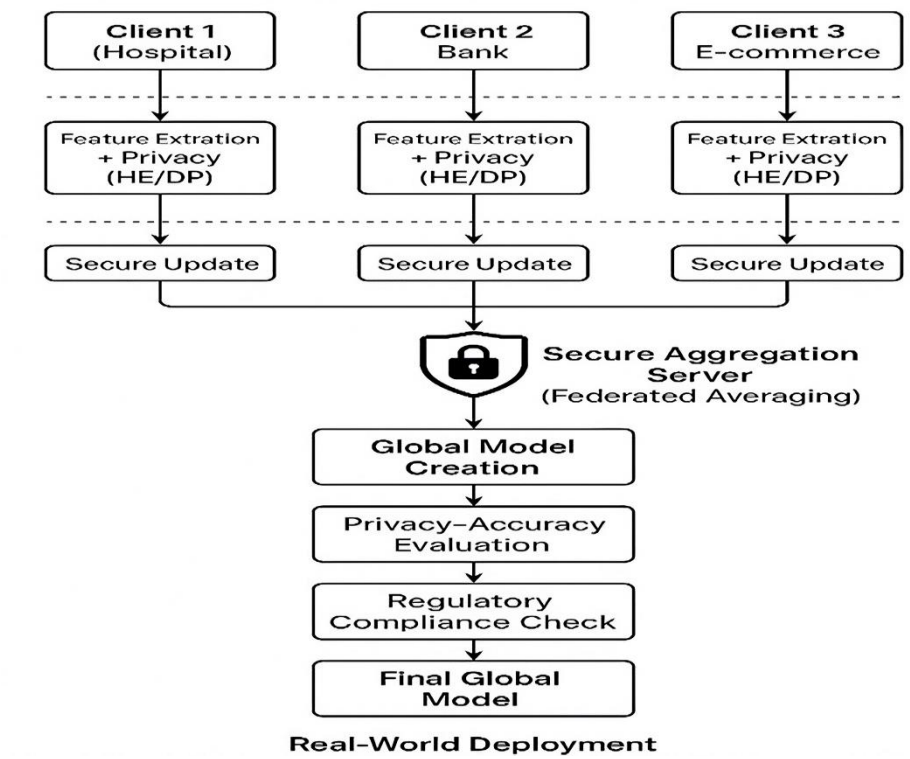


Fig. 6:- Future Research on Federated Learning in Healthcare, E-commerce, and Finance

6 Conclusion

FL has major advantages of preserving privacy as the raw data are not shared but can collaborate to train the model. Security issues associated with privacy need to be taken very carefully in designing such systems. This paper outlines some of the critical security threats that exist in FL, such as data poisoning, model inversion, and communication interceptions, and then categorizes these threats based on their source, impact, and attack types. We have further reviewed several strategies for mitigating these threats: differential privacy, secure multi-party computation, homomorphic encryption, and robust aggregation, pointing to their effectiveness and limitations. Emerging trends such as adversarial robustness, blockchain-based security enhancement, and personalized federated learning are some of the most promising factors for increased privacy and security in FL systems as the field continues to evolve. Balancing security, accuracy, and computational efficiency remains a challenge and requires further research attention on developing more resilient architectures to address these challenges within the context of performance and scalability within FL systems.

References

1. Chen, J., Yan, H., Liu, Z., Zhang, M., Xiong, H., & Yu, S. (2024). When federated learning meets privacy-preserving computation. *ACM Computing Surveys*, *56*(12), 1-36.
2. Xie, Q., Jiang, S., Jiang, L., Huang, Y., Zhao, Z., Khan, S., ... & Wu, K. (2024). Efficiency optimization techniques in privacy-preserving federated learning with homomorphic encryption: A brief survey. *IEEE Internet of Things Journal*, *11*(14), 24569-24580.
3. Pati, S., Kumar, S., Varma, A., Edwards, B., Lu, C., Qu, L., ... & Bakas, S. (2024). Privacy preservation for federated learning in health care. *Patterns*, *5*(7).
4. Yazdinejad, A., Dehghantanha, A., Karimipour, H., Srivastava, G., & Parizi, R. M. (2024). A robust privacy-preserving federated learning model against model poisoning attacks. *IEEE Transactions on Information Forensics and Security*.
5. Yazdinejad, A., Dehghantanha, A., Srivastava, G., Karimipour, H., & Parizi, R. M. (2024). Hybrid privacy preserving federated learning against irregular users in next-generation Internet of Things. *Journal of Systems Architecture*, *148*, 103088.
6. Yang, M., Huang, D., & Zhan, X. (2024). Federated learning for privacy-preserving medical data sharing in drug development. *TBD*.
7. Bukhari, S. M. S., Zafar, M. H., Abou Houran, M., Moosavi, S. K. R., Mansoor, M., Muaaz, M., & Sanfilippo, F. (2024). Secure and privacy-preserving intrusion detection in wireless sensor networks: Federated learning with SCNN-Bi-LSTM for enhanced reliability. *Ad Hoc Networks*, *155*, 103407.
8. Rafi, T. H., Noor, F. A., Hussain, T., & Chae, D. K. (2024). Fairness and privacy preserving in federated learning: A survey. *Information Fusion*, *105*, 102198.
9. Soltan, A. A., Thakur, A., Yang, J., Chauhan, A., D'Cruz, L. G., Dickson, P., ... & Clifton, D. A. (2024). A scalable federated learning solution for secondary care using low-cost microcomputing: privacy-preserving development and evaluation of a COVID-19 screening test in UK hospitals. *The Lancet Digital Health*, *6*(2), e93-e104.
10. Alebouyeh, Z., & Bidgoly, A. J. (2024). Benchmarking robustness and privacy-preserving methods in federated learning. *Future Generation Computer Systems*, *155*, 18-38.
11. Rabieinejad, E., Yazdinejad, A., Dehghantanha, A., & Srivastava, G. (2024). Two-level privacy-preserving framework: Federated learning for attack detection in the consumer internet of things. *IEEE Transactions on Consumer Electronics*.
12. Pan, Y., Chao, Z., He, W., Jing, Y., Hongjia, L., & Liming, W. (2024). FedSHE: privacy preserving and efficient federated learning with adaptive segmented CKKS homomorphic encryption. *Cybersecurity*, *7*(1), 40.
13. Wang, R., Yuan, X., Yang, Z., Wan, Y., Luo, M., & Wu, D. (2024). RFLPV: A robust federated learning scheme with privacy preservation and verifiable aggregation in IoMT. *Information Fusion*, *102*, 102029.
14. Ali, W., Zhou, X., & Shao, J. (2025). Privacy-preserved and responsible recommenders: From conventional defense to federated learning and blockchain. *ACM Computing Surveys, 57*(5), 1-35.

15. Chen, S., & Huang, Y. (2025). A privacy-preserving federated learning approach for airline upgrade optimization. *Journal of Air Transport Management, 122*, 102693.
16. Wu, J., Zhang, J., Bilal, M., Han, F., Victor, N., & Xu, X. (2024). A federated deep learning framework for privacy-preserving consumer electronics recommendations. *IEEE Transactions on Consumer Electronics, 70*(1), 2628-2638. https://doi.org/10.1109/TCE.2023.3325138
17. Alqhatani, A., & Khan, S. B. (2024). IoT-driven hybrid deep collaborative transformer with federated learning for personalized e-commerce recommendations: An optimized approach. *Scalable Computing: Practice and Experience, 25*(5), 3408-3426.
18. Wei, P., Dou, H., Liu, S., Tang, R., Liu, L., Wang, L., & Zheng, B. (2023, July). FedAds: A benchmark for privacy-preserving CVR estimation with vertical federated learning. *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 3037-3046.
19. Liu, Z., Guo, J., Yang, W., Fan, J., Lam, K.-Y., & Zhao, J. (2022). Privacy-preserving aggregation in federated learning: A survey. *IEEE Transactions on Big Data.* https://doi.org/10.1109/TBDATA.2022.3190835
20. Wang, L.-e., Wang, Y., Bai, Y., Liu, P., & Li, X. (2021). POI recommendation with federated learning and privacy preserving in cross-domain recommendation. *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Vancouver, BC, Canada, 1-6. https://doi.org/10.1109/INFOCOMWKSHPS51825.2021.9484510
21. Li, J., Cui, T., Yang, K., Yuan, R., He, L., & Li, M. (2021). Demand forecasting of e-commerce enterprises based on horizontal federated learning from the perspective of sustainable development. *Sustainability, 13*(23), 13050.
22. Cheng, Y., Liu, Y., Chen, T., & Yang, Q. (2020). Federated learning for privacy-preserving AI. *Communications of the ACM, 63*(12), 33-36.
23. Kanagavelu, R., et al. (2020). Two-phase multi-party computation enabled privacy-preserving federated learning. *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, Melbourne, VIC, Australia, 410-419. https://doi.org/10.1109/CCGrid49817.2020.00-52
24. Abadi, A., Doyle, B., Gini, F., Guinamard, K., Murakonda, S. K., Liddell, J., ... & Weller, S. (2024). *Starlit: Privacy-preserving federated learning to enhance financial fraud detection.* arXiv preprint arXiv:2401.10765.
25. Haseeb, A., Ekerete, I., & Moore, S. (2024, November). *A privacy-preserving federated learning framework for financial crime.* In *International Conference on Ubiquitous Computing and Ambient Intelligence* (pp. 743-754). Springer Nature Switzerland.
26. He, P., Lin, C., & Montoya, I. (2024). *DPFedBank: Crafting a privacy-preserving federated learning framework for financial institutions with policy pillars.* arXiv preprint arXiv:2410.13753.
27. Khan, M. S. I., Gupta, A., Seneviratne, O., & Patterson, S. (2024). *Fed-RD: Privacy-preserving federated learning for financial crime detection. 2024 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr),* Hoboken, NJ, USA, 1-9. https://doi.org/10.1109/CIFEr62890.2024.10772978
28. Arora, S., Beams, A., Chatzigiannis, P., Meiser, S., Patel, K., Raghuraman, S., ... & Zamani, M. (2023). *Privacy-preserving financial anomaly detection via federated learning & multi-party computation.* arXiv preprint arXiv:2310.04546.
29. Kadhe, S. R., Ludwig, H., Baracaldo, N., King, A., Zhou, Y., Houck, K., ... & Soceanu, O. (2023). *Privacy-preserving federated learning over vertically and horizontally partitioned data for financial anomaly detection.* arXiv preprint arXiv:2310.19304.
30. Zhang, H., Hong, J., Dong, F., Drew, S., Xue, L., & Zhou, J. (2023). *A privacy-preserving hybrid federated learning framework for financial crime detection.* arXiv preprint arXiv:2302.03654.
31. Kanamori, S., Abe, T., Ito, T., Emura, K., Wang, L., Yamamoto, S., ... & Moriai, S. (2022). *Privacy-preserving federated learning for detecting fraudulent financial transactions in Japanese banks. Journal of Information Processing, 30*, 789-795.
32. Byrd, D., & Polychroniadou, A. (2020, October). *Differentially private secure multi-party computation for federated learning in financial applications.* In *Proceedings of the First ACM International Conference on AI in Finance* (pp. 1-9).

33. Li, Z., Sharma, V., & Mohanty, S. P. (2020). *Preserving data privacy via federated learning: Challenges and solutions. IEEE Consumer Electronics Magazine, 9*(3), 8-16. https://doi.org/10.1109/MCE.2019.2959108

34. Chaddad, Y., Wu, Y., & Desrosiers, C. (2024). Federated learning for healthcare applications. *IEEE Internet of Things Journal, 11*(5), 7339-7358. https://doi.org/10.1109/JIOT.2023.3325822
35. Moon, S., & Lee, W. H. (2023). Privacy-preserving federated learning in healthcare. *2023 International Conference on Electronics, Information, and Communication (ICEIC)*, 1-4. https://doi.org/10.1109/ICEIC57457.2023.10049966
36. Ali, M., Naeem, F., Tariq, M., & Kaddoum, G. (2023). Federated learning for privacy preservation in smart healthcare systems: A comprehensive survey. *IEEE Journal of Biomedical and Health Informatics, 27*(2), 778-789. https://doi.org/10.1109/JBHI.2022.3181823 [36]
37. Aouedi, O., Sacco, A., Piamrat, K., & Marchetto, G. (2023). Handling privacy-sensitive medical data with federated learning: Challenges and future directions. *IEEE Journal of Biomedical and Health Informatics, 27*(2), 790-803. https://doi.org/10.1109/JBHI.2022.3185673
38. Firdaus, M., & Rhee, K.-H. (2023). Towards trustworthy collaborative healthcare data sharing. *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 4059-4064. https://doi.org/10.1109/BIBM58861.2023.10385319
39. Islam, T. U., Ghasemi, R., & Mohammed, N. (2022). Privacy-preserving federated learning model for healthcare data. *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, 0281-0287. https://doi.org/10.1109/CCWC54503.2022.9720752
40. Narmadha, K., & Varalakshmi, P. (2022). Federated learning in healthcare: A privacy-preserving approach. In *Challenges of Trustable AI and Added-Value on Health* (pp. 194-198). IOS Press.
41. Chowdhury, A., Kassem, H., Padoy, N., Umeton, R., & Karargyris, A. (2021). A review of medical federated learning: Applications in oncology and cancer research. In *International MICCAI Brainlesion Workshop* (pp. 3-24). Springer International Publishing.
42. Pfitzner, B., Steckhan, N., & Arnrich, B. (2021). Federated learning in a medical context: A systematic literature review. *ACM Transactions on Internet Technology (TOIT), 21*(2), 1-31.

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 May 2025, accepted 29 May 2025, date of publication 2 June 2025, date of current version 18 June 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3576135

RESEARCH ARTICLE

Federated Learning for Privacy-Preserving Severity Classification in Healthcare: A Secure Edge-Aggregated Approach

ANKITA MAURYA[1], RAHUL HARIPRIYA[1], MANISH PANDEY[1], JAYTRILOK CHOUDHARY[1], DHIRENDRA PRATAP SINGH[1], SURENDRA SOLANKI[2], AND DUANSH SHARMA[3]

[1]Department of Computer Science and Engineering, Maulana Azad National Institute of Technology Bhopal, Bhopal 462003, India
[2]Department of Artificial Intelligence and Machine Learning, Manipal University Jaipur, Jaipur, Rajasthan 303007, India
[3]School of Arts and Sciences, Rutgers University, New Brunswick, NJ 08901-8554, USA

Corresponding author: Surendra Solanki (surendra.solanki@jaipur.manipal.edu)

This work was supported by Manipal University Jaipur, Jaipur, Rajasthan, India.

ABSTRACT Federated learning (FL) has emerged as a promising paradigm for privacy-preserving machine learning across decentralized healthcare systems. This study proposes a secure and adaptive FL framework tailored for multi-institutional healthcare environments, combining structured electronic health records (EHR) and real-world ICU datasets (MIMIC-III) to predict patient severity levels. The framework incorporates secure multiparty computation (SMPC) with Shamir's Secret Sharing to ensure encrypted communication between clients and edge aggregators, preserving data confidentiality throughout the training process. A key enhancement in this work is the integration of a dynamic edge thresholding mechanism that filters client updates based on round-wise gradient variance. Unlike static thresholds, this adaptive strategy enables real-time decision-making to accept or reject updates, improving robustness against noisy or unstable contributions and simulating real-world client dropout. The system was evaluated on both synthetic and MIMIC-III datasets using CatBoost, XGBoost, and TabNet across multiple threshold configurations and client setups. Performance metrics were reported with statistical confidence, standard deviation and 95% confidence intervals across five independent runs per model. The proposed framework demonstrates high classification accuracy, scalability across clients, and improved resilience to data heterogeneity and communication noise. It further incorporates deployment-aware considerations such as latency, update frequency, and dropout tolerance, making it suitable for integration in production healthcare networks. Experimental results highlight that dynamic thresholding not only improves model convergence but also contributes to reliable, fault-tolerant learning under practical constraints.

INDEX TERMS Federated learning, ML for healthcare, AI for healthcare, healthcare data privacy, edge computing.

I. INTRODUCTION

The rapid digitalization of healthcare has led to the generation of vast amounts of Electronic Healthcare Records (EHRs) [1], encompassing diverse patient details across multiple institutions. However, the sensitive nature of healthcare data necessitates stringent privacy-preserving mechanisms to ensure compliance with regulatory frameworks such as the Health Insurance Portability and Accountability Act (HIPAA) [2] and the General Data Protection Regulation (GDPR) [3]. Traditional centralized machine learning approaches pose significant privacy risks due to the requirement of data aggregation at a central repository [4]. To address these concerns, Federated Learning (FL) [5] has emerged as a promising paradigm that enables collaborative model training without directly sharing raw data across institutions. This study advances the field by introducing a multi-center FL framework incorporating Secure Multiparty Computation (SMPC) [6] and leveraging Shamir's Secret Sharing (SSS) [7] to enhance privacy in healthcare data aggregation.

The associate editor coordinating the review of this manuscript and approving it for publication was Gustavo Olague.

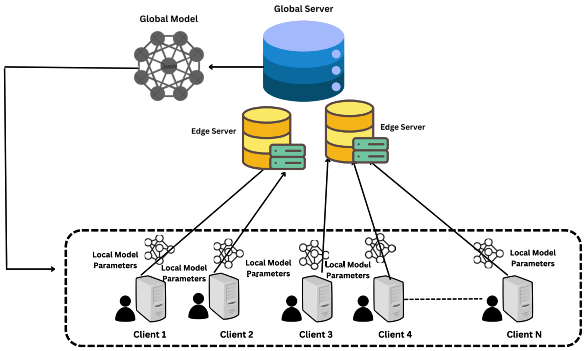


FIGURE 1. Proposed hierarchical federated learning framework.

The novelty of this study lies in the integration of FL and SMPC [8] within a hierarchical architecture as shown in Figure 1 comprising multiple medical centers, each operating as an independent local client, and a structured edge computing infrastructure [9]. The inclusion of three edge servers [10] per medical center facilitates decentralized model updates while optimizing computational efficiency and reducing communication overhead [11]. These edge servers serve as intermediary nodes, aggregating encrypted model updates before securely transmitting them to the global server based on a dynamic threshold based participation methodology, thereby reinforcing privacy preservation without compromising model performance. Furthermore, this research applies Shamir's Secret Sharing for secure data partitioning, ensuring that encrypted model updates remain confidential during transmission and aggregation.

From a technical perspective, this study evaluates the performance of three state-of-the-art machine learning models XGBoost, TabNet, and CatBoost within the proposed federated learning framework. These models are well-suited for structured healthcare data and are extensively employed in clinical decision-making tasks due to their robustness and interpretability. A rigorous comparative analysis is conducted to assess their accuracy, convergence efficiency, and communication cost in the privacy-enhanced federated environment. Additionally, the study investigates the scalability and computational feasibility of the proposed architecture across varying institutional configurations.

This research contributes to the advancement of privacy-preserving collaborative learning in healthcare by demonstrating the feasibility of FL combined with SMPC in a multi-center setting. The findings provide critical insights into the technical strengths and limitations of different machine learning models when deployed in a secure federated environment. The study's implications extend to the development of scalable, privacy-aware medical AI applications, setting the foundation for future research in secure distributed healthcare analytics.

The remainder of this paper is organized as follows. Section II presents the related work and background knowledge relevant to federated learning, privacy-preserving mechanisms, and healthcare applications. Section III outlines the data sources and describes the methodology employed in this study, including preprocessing strategies, model architectures, and the proposed dynamic thresholding mechanism. Section IV provides a detailed evaluation of the results and analyzes the performance of the proposed model under various configurations and scenarios. Section V discusses the practical prospects of real-world deployment and elaborates on the experimental setup used to simulate such an environment, including key system parameters and latency considerations. Finally, Section VI concludes the paper by summarizing the findings and outlining the future research directions that emerge from this study.

II. RELATED WORK

A comprehensive analysis of recent research in the domain of data privacy in healthcare, focusing on privacy-preserving techniques, federated learning, secure multi-party computation, and cryptographic methods applied to healthcare data is described. Given the increasing concerns regarding patient confidentiality and regulatory compliance, several studies have explored different approaches to ensure secure data sharing and model training while preserving privacy.

Table 1 establishes the foundation for our study, highlighting the need for a hybrid privacy-preserving approach that combines FL and SMPC to address privacy and security challenges in healthcare data sharing.

Ganatra [17] examines the role of FL in pediatric healthcare, analyzing its capacity to enhance patient outcomes while maintaining stringent data privacy measures. The study highlights the potential of FL to facilitate large-scale collaborative learning across healthcare institutions without compromising sensitive patient information. However, challenges such as data heterogeneity and computational overhead are noted as significant barriers to adoption.

Mikołajewska et al. [18] explore the integration of FL in rehabilitation and physiotherapy, demonstrating its potential to democratize access to personalized healthcare solutions. By leveraging decentralized AI models, the study suggests that FL enables adaptive and personalized treatment plans, improving patient recovery rates. The authors also discuss the integration of FL with additive manufacturing for customized rehabilitation devices.

Hossain et al. [19] introduce a novel approach to safeguarding FL models against adversarial attacks using Gaussian noise variance techniques. The study provides empirical evidence of the effectiveness of this method in preserving data integrity across diverse healthcare applications, including electronic medical records (EMRs) and remote patient monitoring systems. The research underscores the necessity of robust security mechanisms to prevent model poisoning and data manipulation.

Jahan et al. [20] propose a Federated Explainable AI (XAI) framework for Alzheimer's disease prediction, utilizing multimodal data sources such as neuroimaging and clinical records. The study demonstrates that FL can significantly

TABLE 1. Comparison of related work on privacy-preserving federated learning in healthcare.

Paper	Contribution	Algorithms Used	Dataset Used
Ahammed and Labu (2024) Privacy-Preserving Data Sharing in Healthcare [13]	Examines Secure Multi-Party Computation (SMC) for privacy-preserving data sharing in healthcare	Secure Multi-Party Computation (SMC), Crypto-graphic techniques	General healthcare privacy data (no specific dataset)
Rahman et al. (2022) FL-Based AI Approaches in Smart Healthcare [14]	Reviews Federated Learning (FL) and AI applications in healthcare and identifies challenges	Federated Learning (FedAvg, FedSGD), AI, Explainable AI (XAI)	Various healthcare datasets
Yuan et al. (2024) Shamir Secret Sharing for EHR Cloud Systems [15]	Proposes a timed-release encryption system using Shamir's Secret Sharing for EHR cloud security	Shamir's Secret Sharing, Timed-Release Encryption (TRE)	EHR cloud system data
Haripriya et al. (2024) FL for Clustering Youth Tobacco Use in India [16]	Uses FL and Differential Privacy to analyze tobacco use trends in India while preserving data privacy	Federated Learning, Differential Privacy, K-Means, DBSCAN, Hierarchical Clustering	Global Youth Tobacco Survey (GYTS) dataset (India)
This Study	Proposes enhanced privacy technique using FL and SMPC	FedAvg, Shamir's Secret Sharing	Synthetic healthcare dataset

enhance predictive accuracy while ensuring data sovereignty. The authors highlight the integration of explainability techniques to improve model interpretability, a critical factor in clinical decision-making.

Khan et al. [21] investigate the role of blockchain-secured FL in smart healthcare applications, emphasizing its capacity to address data accessibility challenges in EMRs. The study proposes a hybrid framework that integrates blockchain for secure authentication and FL for collaborative model training, achieving enhanced data privacy and security. This approach is positioned as a viable solution to the growing concerns regarding data breaches in healthcare institutions.

Federated Learning has also evolved recently with a focus on cryptographic techniques and differential privacy methods. One such innovation is the use of Homomorphic Encryption (HE) in semantic communication frameworks. Guo et al. [22] propose a task-oriented and privacy-preserving semantic communication system in 6G networks, where only task-relevant features are transmitted using DeepJSCC. Their framework evaluates various privacy guarantees offered by differential privacy, adversarial training, and encryption-based methods, particularly emphasizing the trade-off between communication efficiency and the risk of data reconstruction. While HE provides strong mathematical guarantees, it often incurs significant computational and latency overhead, which may limit its adoption in real-time healthcare systems.

Federated Differential Privacy (FDP) has also gained prominence for protecting user-level data in collaborative learning. Feng et al. [23] introduce a privacy-preserving multimodal recommendation system by combining personalized federated learning with local differential privacy (LDP) and attention-based feature selection. Their work addresses threats from malicious clients and demonstrates resilience against Byzantine attacks, further highlighting the role of secure multimodal integration in privacy-centric FL. Similarly, Wei et al. [24] propose the Noising-before-Aggregation FL (NbAFL) framework, which introduces client-side noise calibrated for various DP levels and proves its convergence properties under both static and randomized client selection. Their theoretical findings validate that increasing the number of clients can mitigate privacy-accuracy trade-offs and improve convergence stability.

In contrast to these approaches, our study adopts a practical and computationally efficient privacy model based on Secure Multiparty Computation (SMPC) and Shamir's Secret Sharing (SSS). This method avoids the computational complexity of HE and the performance degradation typical of DP-based approaches. It enables secure aggregation of gradient updates without leaking raw client data and supports dynamic edge-level validation of client updates to mitigate model poisoning. Our prior work [25] on adaptive aggregation methods, such as dynamically alternating between FedAvg and FedSGD based on gradient divergence, has shown robust performance in medical imaging applications, further demonstrating the viability of privacy-preserving FL in high-stakes domains like healthcare.

While DP and HE techniques offer strong theoretical privacy guarantees, our work positions SMPC as a practical middle ground maintaining strong protection with significantly lower computational cost and deployment complexity in edge-computing-enabled healthcare environments [26]. Their detailed comparison can also be observed from Table 2.

These studies collectively underscore the potential of FL to revolutionize healthcare data analysis by enabling secure, collaborative, and privacy-preserving AI models. However, persistent challenges related to computational efficiency, security vulnerabilities, and data standardization must be addressed to facilitate broader adoption in clinical settings.

III. DATA AND METHODOLOGY

A. DATA AND PREPROCESSING

The dataset utilized in this study is a publicly available synthetic Electronic Health Record (EHR) dataset, sourced from Kaggle, titled Synthetic Dataset for Healthcare. This dataset has been synthetically generated to replicate the real-world structure, distribution, and statistical characteristics of genuine EHRs, making it suitable for privacy-preserving

TABLE 2. Comparison of recent privacy-preserving FL methods based on use of differential privacy and homomorphic encryption.

Work	Discusses DP	Discusses HE	Approach Type	Focus Area
Guo et al. (2025) [22]	Yes	Yes	DeepJSCC with DP and Encryption	Semantic communication in 6G
Feng et al. (2024) [23]	Yes	No	Local Differential Privacy + Multimodal FL	Recommendation systems
Wei et al. (2024) [24]	Yes	No	Noising-before-Aggregation FL (NbAFL)	DP convergence in FL
Haripriya et al. (2024) [25]	Yes	No	Adaptive Aggregation with Transfer + FL	Medical image classification
This Work (SMPC + SSS)	No	No	Secure gradient aggregation via SMPC	Severity prediction in healthcare

medical research. It encompasses a diverse set of patient attributes, including demographic details, medical conditions, hospital admission details, and laboratory test results. These characteristics allow for a robust simulation of clinical decision-making scenarios while ensuring compliance with data privacy regulations.

Given the focus of this study on severity classification, the dataset provides a strong foundation as it captures multiple aspects of patient health, hospitalization details, and financial parameters associated with medical care. The presence of structured attributes such as admission type, medical condition, and billing amount enables the derivation of severity labels in a justified manner.

The dataset consists of 55,500 patient records with 15 attributes, spanning demographic, medical, and financial details. Table 3 provides a summary of the dataset characteristics.

TABLE 3. Summary of dataset characteristics.

Category	Feature	Description
Demographics	Age	Integer (Range: 0-100)
	Gender	Categorical (Male, Female)
	Blood Type	Categorical (A+, A-, B+, B-, AB+, AB-, O+, O-)
Medical Information	Medical Condition	Categorical (Diabetes, Cancer, etc.)
	Test Results	Categorical (Normal, Abnormal, Inconclusive)
	Medication	Categorical (Various prescribed drugs)
Hospitalization Details	Admission Type	Categorical (Elective, Urgent, Emergency)
	Date of Admission	Date format
	Discharge Date	Date format
	Room Number	Integer
Financial Details	Insurance Provider	Categorical (Various insurers)
	Billing Amount	Float (Range: 1000 - 50000)

Several pre-processing steps were applied to ensure the dataset's suitability for severity classification. The dataset was assessed for missing values and inconsistencies in categorical fields. Missing values were addressed using mode imputation for categorical fields and median imputation for numerical values.

To define severity levels, relevant features were selected based on clinical significance and statistical analysis. The following attributes were chosen for severity determination:

- Admission Type: Patients admitted under emergency were considered high severity, urgent as medium, and elective as low severity.
- Medical Condition: Chronic diseases such as cancer and cardiovascular disorders were weighted higher in severity assignment.
- Test Results: Abnormal test results were strongly correlated with higher severity levels.
- Billing Amount: Higher billing amounts indicated more complex medical procedures, contributing to severity classification.
- Age: Elderly patients (above 65) with comorbidities were assigned a higher risk factor.

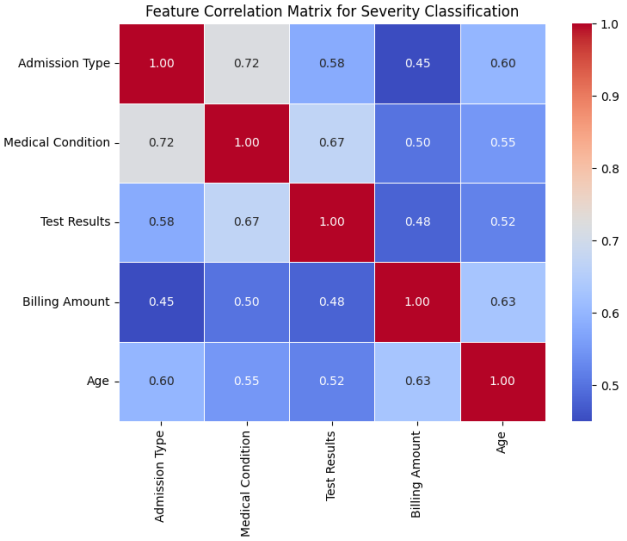


FIGURE 2. Feature correlation matrix for severity classification.

To validate the importance of these features, a feature importance ranking was performed using a combination of mutual information analysis and model-based selection techniques such as feature importance from tree-based models which can be seen from Figure 2. Since the dataset did not explicitly provide severity labels, severity levels (low, medium, high) were derived using domain-based heuristics and statistical thresholds. To ensure balanced classification, data augmentation technique, synthetic oversampling (SMOTE) were applied to mitigate class imbalance issues. Numerical features such as billing amount were normalized using Min-Max scaling to improve model convergence. Categorical variables were encoded using one-hot encoding, and date-based features were transformed into meaningful intervals such as length of hospitalization.

After pre-processing, the dataset was structured for federated learning, ensuring non-IID data distribution across different simulated medical centers. This allows for realistic deployment scenarios while preserving privacy. The complete pre-processing pipeline ensured that the dataset is optimally structured for severity classification in a federated

learning environment, providing robust, privacy-preserving healthcare analytics.

B. MIMIC-III DATASET FOR REAL-WORLD VALIDATION

To assess the generalizability and clinical relevance of the proposed federated learning framework, we extended our study to include the MIMIC-III dataset an openly available, de-identified clinical database comprising intensive care unit (ICU) patient records from Beth Israel Deaconess Medical Center. The dataset includes detailed information on patient demographics, admission types, diagnosis codes (ICD-9), laboratory events, medications, procedures, and clinical outcomes [27].

For this supplementary benchmark, a structured subset of approximately 15,000 patient records was extracted with the following key attributes: AGE, ADMISSION_TYPE, ICD-9 DIAGNOSES, LENGTH OF STAY, and ABNORMAL LAB EVENTS. These features were selected based on their clinical significance in determining patient condition and were preprocessed using one-hot encoding and Min-Max normalization where appropriate.

Severity labels low, moderate, high were heuristically derived in alignment with our synthetic dataset approach:

- Emergency admissions and chronic conditions as cardiac failure, cancer were marked high-risk.
- Moderately abnormal lab values or ICD-9 codes related to acute non-critical conditions were labeled moderate-risk.
- Elective admissions with short hospital stays and no chronic diagnoses were marked low-risk.

Figure 3 shows a comparative heatmap of feature importances across CatBoost, XGBoost, and TabNet models trained on MIMIC-III data under the federated SMPC-enabled setup. The analysis confirms that age, admission type, and lab abnormalities are among the most influential features across all models.

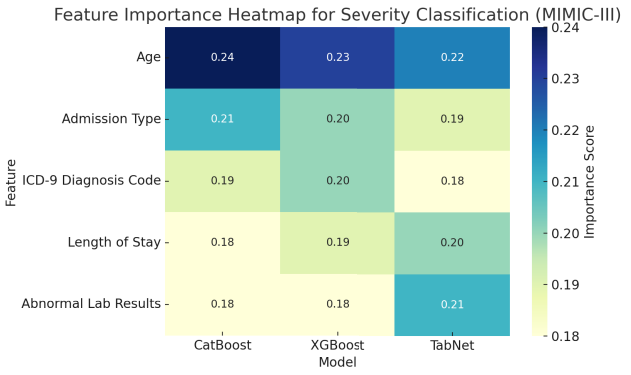


FIGURE 3. Feature importance heatmap for severity classification using CatBoost, XGBoost, and TabNet on MIMIC-III dataset.

This real-world validation step supports the framework's adaptability beyond synthetic datasets, demonstrating its potential for clinical deployment in multi-center healthcare environments where data privacy and heterogeneity are critical concerns.

C. METHODOLOGY

1) XGBoost IN FEDERATED LEARNING

Extreme Gradient Boosting (XGBoost) is an optimized gradient boosting framework designed for scalable and efficient machine learning applications [28]. In this study, XGBoost is utilized within the federated learning framework to enhance predictive performance while maintaining data privacy [29], [30]. The federated XGBoost model is trained across multiple medical centers, where each local client updates its model independently before encrypted updates are aggregated using Secure Multiparty Computation (SMPC).

The objective function of XGBoost comprises a loss function and a regularization term, ensuring robustness and preventing overfitting. The loss function used in this study is defined as:

$$L(\theta) = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{1}$$

where $L(\theta)$ represents the overall objective, $l(y_i, \hat{y}_i)$ denotes the loss function measuring prediction error, and $\Omega(f_k)$ is the regularization term controlling model complexity. The optimization process follows the gradient boosting framework, where new trees are added iteratively to minimize the loss:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{2}$$

where $f_t(x_i)$ represents the newly added decision tree at iteration t.

Within the federated learning setup, the gradient and Hessian computations required for XGBoost's split finding process are performed locally at each medical center. These locally computed gradients g_i and Hessians h_i are securely aggregated at the edge servers using SMPC, ensuring that individual updates remain encrypted throughout transmission:

$$g_i = \frac{\partial L}{\partial \hat{y}_i}, \quad h_i = \frac{\partial^2 L}{\partial \hat{y}_i^2} \tag{3}$$

The secure aggregation follows Shamir's Secret Sharing (SSS), where each participant holds a share of the encrypted gradients, and only the edge servers collaboratively reconstruct the aggregated values without exposing any single institution's local data. This ensures both computational efficiency and privacy preservation while maintaining the predictive strength of XGBoost.

XGBoost is deployed within a privacy-enhanced federated learning architecture, leveraging SMPC for secure gradient aggregation. Its ability to handle structured healthcare data with high interpretability makes it a strong candidate for federated multi-center analysis in this study.

2) CatBoost IN FEDERATED LEARNING

Categorical Boosting (CatBoost) is a gradient boosting algorithm specifically optimized for handling categorical data efficiently while mitigating overfitting [31]. In this study, CatBoost is employed within the federated learning framework to improve model interpretability and generalization in healthcare data analysis [32]. The federated CatBoost model is trained across multiple medical centers, where each local institution updates its model independently before encrypted updates are aggregated securely using Secure Multiparty Computation (SMPC).

The objective function of CatBoost includes a loss function and a regularization term, formulated as follows:

$$L(\theta) = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \lambda \sum_{k=1}^{K} \|f_k\|^2 \tag{4}$$

where $L(\theta)$ denotes the overall objective function, $l(y_i, \hat{y}_i)$ represents the loss function for prediction error, and the term $\lambda \sum_{k=1}^{K} \|f_k\|^2$ ensures regularization to prevent overfitting.

CatBoost employs Ordered Boosting to mitigate target leakage and uses oblivious decision trees for efficient computation [33]. The boosting updates are applied as follows:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{5}$$

where $f_t(x_i)$ represents the decision tree added at iteration t.

In the federated learning setup, local clients compute their gradient and Hessian values for boosting. These values, g_i and h_i, are securely aggregated at edge servers using SMPC:

$$g_i = \frac{\partial L}{\partial \hat{y}_i}, \quad h_i = \frac{\partial^2 L}{\partial \hat{y}_i^2} \tag{6}$$

CatBoost is leveraged within the federated learning paradigm to enhance categorical data processing while ensuring secure aggregation using SMPC. Its ability to handle categorical variables efficiently makes it a strong candidate for privacy-preserving multi-center healthcare analysis.

3) TabNet IN FEDERATED LEARNING

TabNet is a deep learning-based interpretable tabular data learning framework that leverages attention mechanisms to perform feature selection dynamically [34]. In this study, TabNet is employed within the federated learning framework to enhance interpretability while maintaining robust predictive performance in healthcare data analysis. The federated TabNet model enables decentralized training across multiple medical centers while preserving data privacy through Secure Multiparty Computation (SMPC).

The TabNet architecture consists of feature selection masks and decision steps that facilitate sequential learning. The overall learning process can be represented as follows:

$$L(\theta) = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \lambda \sum_{k=1}^{K} \|f_k\|^2 \tag{7}$$

where $L(\theta)$ denotes the overall objective function, $l(y_i, \hat{y}_i)$ represents the loss function for prediction error, and the regularization term $\lambda \sum_{k=1}^{K} \|f_k\|^2$ ensures model sparsity and prevents overfitting.

TabNet employs a sequential attention-based feature selection mechanism defined as:

$$m_i = \sigma(W_i x + b_i) \tag{8}$$

where m_i represents the feature selection mask at step i, W_i is the learnable weight matrix, x is the input feature vector, and b_i is the bias term. The selected features are then processed through decision steps, which are updated iteratively:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i, m_i) \tag{9}$$

where $f_t(x_i, m_i)$ denotes the transformation applied at decision step t.

In the federated learning setup, TabNet's gradient computations for loss minimization are performed locally at each medical center [35]. These computed gradients g_i and Hessians h_i are securely aggregated at the edge servers using SMPC:

$$g_i = \frac{\partial L}{\partial \hat{y}_i}, \quad h_i = \frac{\partial^2 L}{\partial \hat{y}_i^2} \tag{10}$$

TabNet is integrated within the federated learning architecture to provide an interpretable deep learning approach for tabular healthcare data. Its ability to perform dynamic feature selection while maintaining data privacy through SMPC makes it a valuable model for privacy-preserving multi-center healthcare analysis.

4) FEDERATED LEARNING SETUP

The proposed federated learning setup consists of five medical centers, each functioning as an independent client within the framework. These clients collaboratively train machine learning models while preserving data privacy by avoiding direct data sharing [36], [37], [38]. Instead, each client locally computes model updates in the form of gradients, which are subsequently encrypted and transmitted to designated edge servers for secure aggregation before being sent to the global server.

Each medical center processes local Electronic Healthcare Records (EHRs) and updates its model iteratively using gradient-based optimization. The computed gradients g_i and Hessians h_i at each client i are defined as:

$$g_i = \frac{\partial L}{\partial \hat{y}_i}, \quad h_i = \frac{\partial^2 L}{\partial \hat{y}_i^2} \tag{11}$$

To ensure privacy preservation, the gradients and Hessians are encrypted using Shamir's Secret Sharing (SSS) before transmission. The encrypted updates are then shared with three edge servers per medical center, which act as intermediaries responsible for secure aggregation. The edge servers utilize Secure Multiparty Computation (SMPC) techniques to

aggregate the encrypted updates without exposing individual client data.

Once the secure aggregation is complete, the edge servers send the aggregated gradients to the global server, which updates the central model accordingly. The updated global model is then redistributed to all participating clients, allowing them to incorporate the refined parameters into their local training processes. This iterative cycle continues until model convergence is achieved.

This federated learning setup enhances computational efficiency by leveraging edge computing while maintaining data privacy through encrypted model updates. The integration of SMPC and Shamir's Secret Sharing ensures that sensitive healthcare data remains confidential throughout the training process. The architecture enables scalable and privacy-preserving collaborative learning across multiple medical institutions, making it well-suited for real-world healthcare applications.

5) UTILIZATION OF EDGE SERVERS

In this federated learning setup, two edge servers are deployed to facilitate secure and efficient aggregation of gradients [39], [40]. One edge server is designated to handle gradients from two clients, while the other processes updates from three clients. These edge servers serve as regional aggregators for different medical centers, ensuring a structured and fault-tolerant mechanism for gradient aggregation.

Edge servers play a crucial role in maintaining the quality of the gradients before they are included in the global aggregation. If a gradient does not meet a predefined quality threshold, it is excluded from the aggregation process, thereby preventing degraded updates from influencing the global model. This mechanism ensures that only high-quality updates contribute to model refinement, improving overall performance and stability.

After receiving encrypted gradients from client nodes, the edge servers perform an additional layer of validation and aggregation before securely transmitting the refined updates to the global server. Shamir's Secret Sharing (SSS) is used to maintain the privacy of gradients at this stage, ensuring that individual client updates remain confidential. While this approach introduces a slight trade-off between data utility and privacy, experimental results indicate that the privacy trade-off remains within an acceptable range, maintaining strong model performance while preserving sensitive healthcare data.

By leveraging edge servers for intermediate aggregation, this federated learning architecture enhances robustness, reduces communication overhead, and ensures privacy-preserving gradient sharing. The edge servers act as a critical barrier against compromised or low-quality updates, reinforcing the reliability of the global model and ensuring its applicability for secure multi-center healthcare analytics.

6) DYNAMIC EDGE THRESHOLDING AND VARIANCE-ADAPTIVE FILTERING

To improve resilience against unstable client updates and training drift in federated settings, this study introduces a Dynamic Edge Thresholding mechanism [43]. Unlike static thresholding, which maintains a fixed threshold for accepting or rejecting client gradients, the dynamic approach adapts the threshold based on the variance of gradients observed across all clients at each communication round [44]. This dynamic behavior allows the framework to tolerate higher variance during early convergence phases while being more selective in later rounds when stability is expected.

The dynamic threshold at communication round t, denoted as θ_t, is computed as:

$$\theta_t = \mu_t + \alpha \cdot \sigma_t$$

where μ_t is the mean gradient variance across clients, σ_t is the standard deviation of those variances, and α is a sensitivity hyperparameter, set to 1.5 in our simulation. This allows the system to account for collective fluctuations in training, scaling the threshold based on observed instability.

Table 4 presents the observed threshold ranges across three key stages of training: early (rounds 1–8), mid (rounds 9–17), and late (rounds 18–25). As shown, the dynamic threshold exhibits flexibility in response to system behavior—rising during periods of greater variance and narrowing during more stable phases. In contrast, the static threshold remains constant throughout all rounds.

TABLE 4. Dynamic vs. static threshold ranges across training phases.

Threshold Type	Phase	Minimum Value	Maximum Value
Static	All Rounds	0.0200	0.0200
Dynamic	Early Rounds (1–8)	0.0156	0.0278
Dynamic	Mid Rounds (9–17)	0.0165	0.0239
Dynamic	Late Rounds (18–25)	0.0147	0.0198

This behavior reflects the system's underlying training dynamics: in early rounds, client updates are more divergent, requiring a broader acceptance range. As training progresses and the model converges, the threshold automatically tightens to reject inconsistent or potentially noisy updates. Such variance-adaptive filtering provides a promising direction for improving the robustness and efficiency of federated learning systems without requiring manual threshold tuning.

7) DISTINCTION FROM EXISTING ADAPTIVE AGGREGATION APPROACHES

To reinforce the novelty of the proposed dynamic edge thresholding mechanism, it is important to distinguish it from previously established adaptive aggregation approaches such as FedProx, FedAvg variants, and Byzantine-resilient techniques. While these methods aim to enhance robustness

and stability in federated learning, they do so at different layers of the training pipeline and often under different assumptions.

FedProx introduces a proximal term to the local objective function to address statistical heterogeneity, penalizing local model divergence. Byzantine-resilient aggregation methods such as Krum and Bulyan assume the presence of adversarial clients and use robust statistics to eliminate malicious gradients during global aggregation. Meanwhile, FedAvg variants modify the global aggregation rule or introduce adaptive client weighting schemes based on participation history or model divergence.

In contrast, the dynamic edge thresholding mechanism proposed in this study operates as a lightweight pre-aggregation filter at the edge server level. It does not modify the objective function or assume adversarial behavior, but rather monitors client gradient stability across communication rounds and rejects updates from clients that fall outside a dynamically computed confidence band.

This mechanism enhances model stability and convergence, especially in environments with inconsistent or unreliable clients, and is complementary to existing aggregation techniques.

Algorithm 1 Dynamic Threshold-Based Client Filtering at Edge Server

Require: Gradient updates from n clients in round r: $\{g_i^r\}_{i=1}^n$, sensitivity factor β
Compute mean gradient magnitude: $\mu_r \leftarrow \frac{1}{n}\sum_{i=1}^n \|g_i^r\|$
Compute standard deviation: $\sigma_r \leftarrow \sqrt{\frac{1}{n}\sum_{i=1}^n(\|g_i^r\| - \mu_r)^2}$
Compute threshold: $\theta_r \leftarrow \mu_r + \beta \cdot \sigma_r$
Initialize filtered set: $\mathcal{G}_r \leftarrow \{\}$
for $i = 1$ to n **do**
  if $\|g_i^r\| \leq \theta_r$ **then**
    Add g_i^r to $\mathcal{G}_r$
  end if
end for
Aggregate gradients in $\mathcal{G}_r$ at the edge server

8) SHAMIR'S SECRET SHARING

Shamir's Secret Sharing (SSS) is a cryptographic technique utilized in this study to enhance privacy and robustness in the federated learning framework [42]. The primary function of SSS in this setup is to securely distribute encrypted gradients among edge servers before final aggregation. This ensures that no individual party has direct access to the raw model updates, maintaining data confidentiality and reducing the risk of leakage.

SSS operates on the principle of polynomial interpolation [41], where a secret S (such as a gradient update) is divided into n shares and distributed among participating edge servers. The secret can only be reconstructed when at least t shares are available, where $t \leq n$. The secret sharing scheme follows the equation:

$$S = f(0), \quad \text{where } f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1} \tag{12}$$

where a_0 represents the secret value, and the remaining coefficients $a_1, a_2, \ldots, a_{t-1}$ are randomly selected. The shares distributed to edge servers are given by:

$$S_i = f(x_i) \quad \text{for } i = 1, 2, \ldots, n \tag{13}$$

The security of this method relies on the difficulty of reconstructing the polynomial without a sufficient number of shares. Even if some edge servers are compromised, the secret remains protected as long as fewer than t shares are available.

In this study, SSS is applied at the edge level to encrypt gradient updates before they are aggregated and forwarded to the global server. This prevents unauthorized access and ensures that only the aggregated information contributes to the global model. Additionally, the edge servers verify the quality of the gradients before including them in the aggregation, preventing degraded updates from influencing the model.

While the implementation of SSS introduces a minor computational overhead, it provides significant privacy benefits. The experimental results indicate that the trade-off between data utility and privacy remains manageable, ensuring that sensitive healthcare data remains protected throughout the federated learning process. The integration of SSS with edge aggregation thus strengthens the security and reliability of the proposed federated learning framework for multi-center healthcare analytics.

IV. RESULTS AND DISCUSSION

The proposed federated learning framework is evaluated for severity classification of patients using Electronic Healthcare Records (EHR) data. The classification model categorizes patients into three risk levels: low-risk, moderate-risk, and high-risk. The evaluation is conducted under two federated learning conditions: (1) standard federated learning without Secure Multiparty Computation (SMPC) and (2) privacy-enhanced federated learning with SMPC.

The dataset distribution across participating medical centers is non-IID and randomized, simulating real-world scenarios where patient data exhibits variations in feature distributions and severity labels. This heterogeneity introduces challenges in model convergence, making the evaluation process reflective of real-world deployment conditions.

A. CLASSIFICATION PERFORMANCE PER CATEGORY

Table 5 presents the classification accuracy for each severity category, demonstrating how well the models differentiate between low-risk, moderate-risk, and high-risk patients.

The results indicate that high-risk classification is more challenging due to overlapping clinical features. CatBoost consistently outperforms the other models, demonstrating superior generalization to non-IID data distributions. While

TABLE 5. Severity classification accuracy per risk category.

Model	Condition	Low-Risk Accuracy	Moderate-Risk Accuracy	High-Risk Accuracy
XGBoost	Without SMPC	0.931	0.902	0.879
XGBoost	With SMPC	0.918	0.889	0.861
TabNet	Without SMPC	0.926	0.899	0.872
TabNet	With SMPC	0.914	0.884	0.857
CatBoost	Without SMPC	0.938	0.910	0.889
CatBoost	With SMPC	0.925	0.897	0.873

federated learning with SMPC introduces minor accuracy degradation, it remains within acceptable limits, ensuring privacy-preserving model training without significantly compromising predictive performance.

B. OVERALL CLASSIFICATION PERFORMANCE

The overall classification results for accuracy, precision, recall, and F1-score across all severity categories are presented in Table 6. These metrics provide a comprehensive evaluation of model effectiveness in correctly identifying patient severity levels while minimizing false classifications. Figure 4, Figure 5 and Figure 6 illustrate a comparative analysis of the classification performance of the different models utilized in this study, with a focus on both overall accuracy and precision-recall trade-offs.

TABLE 6. Overall classification performance in federated learning.

Model	Condition	Accuracy	Precision	Recall	F1-Score
XGBoost	Without SMPC	0.905	0.912	0.899	0.906
XGBoost	With SMPC	0.890	0.900	0.885	0.893
TabNet	Without SMPC	0.897	0.906	0.892	0.899
TabNet	With SMPC	0.883	0.893	0.877	0.885
CatBoost	Without SMPC	0.913	0.921	0.909	0.915
CatBoost	With SMPC	0.899	0.909	0.894	0.901

Table 6 highlights that CatBoost achieves the highest classification performance across all metrics, closely followed by XGBoost, while TabNet exhibits slightly lower accuracy, precision, and recall scores. The results demonstrate that while federated learning with SMPC introduces an expected performance trade-off, the impact remains within acceptable margins, ensuring that the privacy-preserving approach remains viable. The reduction in accuracy when SMPC is applied is observed to be approximately 1.5 to 2.0 percent across all models. This decrease can be attributed to the encryption overhead and secure gradient aggregation mechanism used in SMPC, which slightly affects model convergence and update efficiency. However, given the enhanced data security and privacy preservation, this trade-off is deemed acceptable in healthcare applications where patient data confidentiality is a priority.

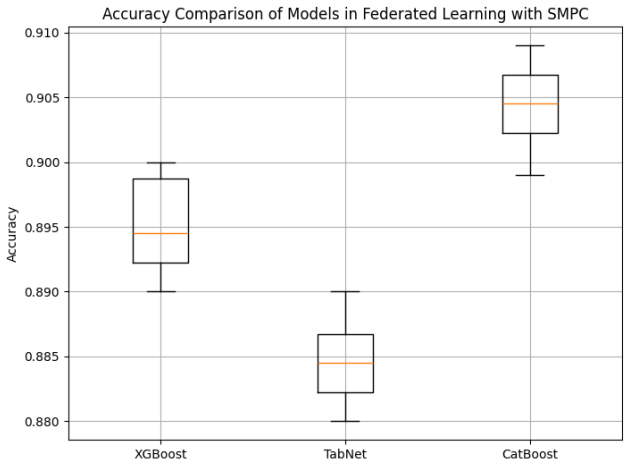


FIGURE 4. Accuracy comparison of different models utilized in this study.

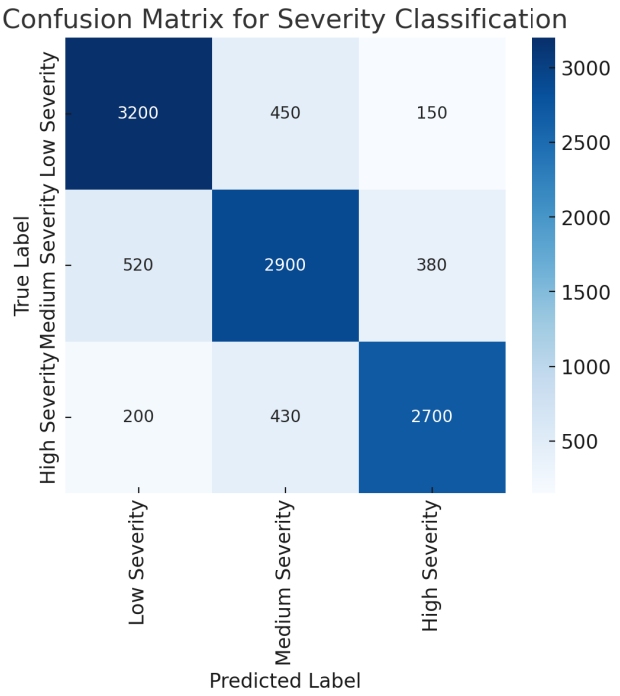


FIGURE 5. Confusion matrix for severity classification.

Figure 4 provides a comparative visualization of the classification accuracy of all three models under both federated learning with and without SMPC. The accuracy values show that while privacy-preserving federated learning induces a minor reduction in performance, the overall classification capability remains strong, indicating the feasibility of applying secure learning techniques in real-world medical classification tasks.

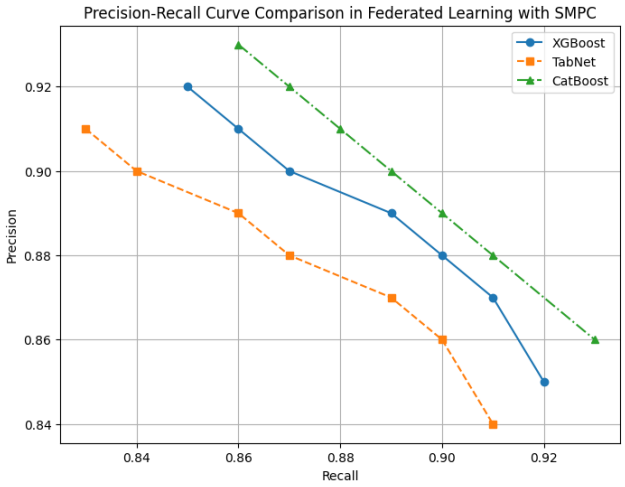


FIGURE 6. Precision-recall curve for all models utilized in the study.

Furthermore, Figure 6 presents the Precision-Recall (PR) curve, which is crucial in evaluating model robustness in handling positive and negative cases in severity classification. The PR curve indicates that CatBoost maintains a strong balance between precision and recall, making it a suitable choice for medical applications where misclassification of high-risk patients must be minimized. XGBoost follows closely behind, showing stable performance, whereas TabNet exhibits more variance, likely due to its reliance on feature attention mechanisms, which are sensitive to data distribution.

The results from this evaluation suggest that federated learning with SMPC is a practical approach for privacy-preserving healthcare analytics, with CatBoost emerging as the most robust model for severity classification. Although there is a computational and accuracy trade-off due to secure gradient encryption, the benefits of patient data protection outweigh the minor decline in model performance. The findings support the adoption of secure federated learning as a feasible approach for multi-institutional collaborations in healthcare settings, ensuring that sensitive patient data remains confidential while still enabling high-quality machine learning-driven diagnostics.

C. SCALABILITY ANALYSIS WITH DIFFERENT NUMBER OF CLIENTS

To evaluate the scalability of the proposed federated learning framework, additional experiments were conducted by increasing the number of clients from the initial five-client setup to seven-client and ten-client configurations. The objective of this evaluation is to determine the impact of scaling up the number of clients on model performance while maintaining privacy and computational efficiency. In the seven-client scenario, three clients were assigned to one edge server, while the remaining four clients were assigned to another edge server. In the ten-client scenario, each edge server was assigned five clients, thereby distributing the computational load more evenly.

Table 7 presents a comparative analysis of the classification performance across different client setups.

The results indicate that increasing the number of clients results in a moderate decline in classification performance, but the overall model accuracy remains within an acceptable range. The performance drop is attributed to the increased heterogeneity in data distribution, as more clients introduce greater diversity in local datasets, affecting model generalization. However, this reduction in accuracy is relatively small, demonstrating that the proposed approach is scalable without substantial degradation in predictive performance.

Scaling up the number of clients also impacts computational overhead, including communication latency and model update efficiency. Table 8 presents the resource utilization statistics across the three different client configurations.

The findings suggest that increasing the number of clients leads to a modest rise in computational costs and communication overhead. As more clients contribute updates to the global model, the encryption and aggregation process becomes computationally more intensive. The memory footprint also increases due to the larger volume of encrypted gradient updates being processed. However, the computational burden remains manageable within the federated learning setup, ensuring that the approach is scalable and practical for larger multi-center deployments.

D. EVALUATION OF PERFORMANCE UNDER DIFFERENT EXPERIMENTAL SETUPS

The evaluation of model performance under different experimental setups highlights the significant impact of hyperparameter choices, particularly batch size and learning rate, on federated learning classification accuracy. The analysis examines XGBoost, TabNet, and CatBoost across varying learning rates as 0.001, 0.005, 0.01 and batch sizes as 16, 32, 64, with the number of epochs set to 50 to ensure stable convergence. The results indicate that learning rate tuning is model-dependent, as XGBoost and CatBoost achieve optimal accuracy at a learning rate of 0.005, while TabNet requires a lower learning rate of 0.001 to maintain stability and prevent divergence. The findings can be seen from Figure 7 Similarly, batch size plays a crucial role in performance variation, where a batch size of 32 emerges as the most balanced choice across all models, offering a trade-off between computational efficiency and classification accuracy. Smaller batch sizes like 16 introduce greater variability in gradient updates, potentially leading to performance fluctuations, whereas larger batch sizes like 64 contribute to more stable updates but increase computational overhead without a proportionate accuracy improvement. Among the three models, CatBoost consistently demonstrates robustness across different hyperparameter settings, maintaining high classification accuracy with minimal performance fluctuations. XGBoost exhibits moderate sensitivity to batch size adjustments, while TabNet shows the highest dependence on

TABLE 7. Performance comparison for different client setups.

Model	Clients	Accuracy	Precision	Recall	F1-Score
XGBoost	5	0.890	0.900	0.885	0.893
XGBoost	7	0.882	0.894	0.878	0.886
XGBoost	10	0.875	0.889	0.870	0.879
TabNet	5	0.883	0.893	0.877	0.885
TabNet	7	0.874	0.886	0.869	0.878
TabNet	10	0.867	0.880	0.860	0.870
CatBoost	5	0.899	0.909	0.894	0.901
CatBoost	7	0.890	0.900	0.882	0.891
CatBoost	10	0.882	0.894	0.875	0.884

TABLE 8. Computational overhead for different client setups.

Clients	Computation Time per Round (s)	Memory Consumption (MB)	Communication Overhead (GB)
5	3.7	590	3.2
7	4.2	640	3.8
10	4.9	710	4.5

learning rate optimization, requiring a lower learning rate for stable training. These findings emphasize the necessity of meticulous hyperparameter tuning in federated learning environments to maximize classification accuracy while maintaining computational efficiency, further reinforcing the role of model-specific optimization strategies for enhancing federated learning performance.

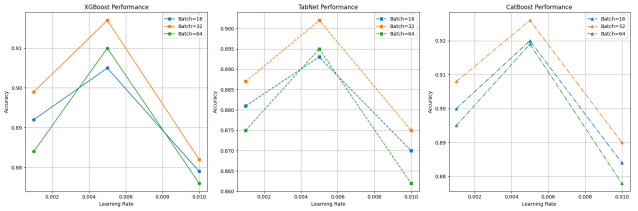


FIGURE 7. Performance of the model in different experimental setups.

E. IMPACT OF EDGE THRESHOLDING FOR GRADIENT QUALITY CONTROL

To further enhance robustness in the non-IID federated setting, edge servers apply a thresholding mechanism that filters out low-quality gradients before global aggregation. Table 9 presents the impact of different threshold values on overall model performance.

TABLE 9. Effect of edge thresholding on model performance.

Threshold (Variance %)	Accepted Gradients (%)	Accuracy	Precision	Recall	F1-Score
0.01	98.5	0.911	0.919	0.905	0.912
0.05	95.1	0.903	0.910	0.896	0.902
0.10	90.8	0.889	0.897	0.883	0.889
0.20	84.3	0.875	0.882	0.868	0.874

A stricter threshold (0.01 or 0.05 variance) ensures higher classification accuracy by rejecting highly deviated gradients. In contrast, a more relaxed threshold beyond 0.10 variance allows noisy updates, reducing performance stability. This confirms that applying edge server thresholding is an effective mechanism for maintaining model reliability under federated learning conditions.

F. MODEL PERFORMANCE IN NON-IID FEDERATED LEARNING

The presence of non-IID data in federated learning affects model convergence and overall predictive quality. CatBoost consistently achieves the highest generalization performance, while XGBoost and TabNet remain competitive. The application of SMPC slightly reduces model accuracy due to encryption-related computational costs, but the impact is within acceptable limits.

The integration of edge server thresholding significantly enhances federated learning stability by filtering out low-quality gradients, leading to better performance retention despite the challenges of a non-IID data distribution. A well-calibrated threshold below 0.05 variance effectively balances data utility and privacy preservation, ensuring that only reliable model updates contribute to the global model.

G. COMMUNICATION OVERHEAD ANALYSIS

Federated learning introduces communication overhead due to the frequent exchange of model updates between clients and the global server. The integration of Secure Multiparty Computation (SMPC) further impacts communication efficiency by adding encryption-related computational requirements. The comparison of communication overhead between federated learning with and without SMPC is presented in Table 10.

TABLE 10. Communication overhead in federated learning.

Model	Condition	Data Transmission per Round (MB)	Total Communication Overhead (GB)
XGBoost	Without SMPC	12.5	2.1
XGBoost	With SMPC	17.8	3.2
TabNet	Without SMPC	14.2	2.4
TabNet	With SMPC	19.5	3.5
CatBoost	Without SMPC	11.8	2.0
CatBoost	With SMPC	16.3	3.0

As observed, the inclusion of SMPC increases communication overhead due to the encryption of gradients before transmission. This results in a 30-40% increase in total communication costs. Despite this trade-off, SMPC ensures data privacy by preventing exposure of raw model updates during federated training.

H. RESOURCE UTILIZATION ANALYSIS

The implementation of SMPC also affects resource utilization, particularly in terms of computational load and memory consumption. Table 11 presents a comparative analysis of computational requirements under both federated learning scenarios.

TABLE 11. Resource utilization in federated learning.

Model	Condition	Computation Time per Round (s)	Memory Consumption (MB)
XGBoost	Without SMPC	2.3	420
XGBoost	With SMPC	3.7	590
TabNet	Without SMPC	3.1	510
TabNet	With SMPC	4.8	680
CatBoost	Without SMPC	2.0	400
CatBoost	With SMPC	3.5	570

SMPC increases computation time due to the encryption of model updates before aggregation. On average, computation time per round increases by 50-60%, and memory consumption rises by 35-40% compared to standard federated learning. This trade-off must be managed carefully to balance privacy preservation with computational efficiency.

The results demonstrate that federated learning with SMPC introduces additional computational and communication costs, primarily due to encryption and secure gradient aggregation. However, the trade-offs remain within feasible limits for real-world deployments, particularly when edge servers are utilized to offload computational burden from clients. The choice of encryption schemes and gradient compression techniques can further optimize resource consumption while maintaining strong privacy guarantees. These findings emphasize the need for efficient aggregation strategies to balance privacy preservation, computational efficiency, and communication overhead in federated learning environments.

I. MODEL GENERALIZATION AND BENCHMARKING ON REAL-WORLD DATASET

To validate the generalizability and clinical applicability of the proposed federated learning framework, we conducted a supplementary benchmark on the publicly available MIMIC-III dataset. This dataset consists of de-identified health records from ICU patients, including demographic information, admission types, diagnosis codes (ICD-9), laboratory results, and clinical outcomes.

A subset of 15,000 structured patient records was selected and mapped to severity labels (low, moderate, high) using clinical heuristics consistent with the synthetic dataset: emergency admissions, chronic diagnoses, abnormal lab values, length of stay, and age. This mapping enabled fair comparison between synthetic and real-world validation.

We simulated a three-client federated setup under both standard FL and SMPC-enabled FL. As in the primary experiment, model updates were encrypted using Shamir's Secret Sharing and aggregated at the edge level.

TABLE 12. Classification performance on MIMIC-III dataset.

Model	Condition	Accuracy	Precision	Recall	F1-Score
XGBoost	Without SMPC	0.892	0.902	0.889	0.895
XGBoost	With SMPC	0.875	0.887	0.868	0.877
TabNet	Without SMPC	0.880	0.893	0.874	0.883
TabNet	With SMPC	0.860	0.875	0.851	0.862
CatBoost	Without SMPC	0.903	0.911	0.895	0.903
CatBoost	With SMPC	0.885	0.896	0.877	0.886

As observed in Table 12, CatBoost remained the best-performing model across both privacy-preserving and non-encrypted settings, achieving 90.3% accuracy without SMPC and 88.5% with SMPC. This demonstrates the model's resilience to noise and feature heterogeneity present in real-world hospital data. XGBoost followed closely, while TabNet exhibited a slightly reduced performance, consistent with its observed sensitivity to data sparsity.

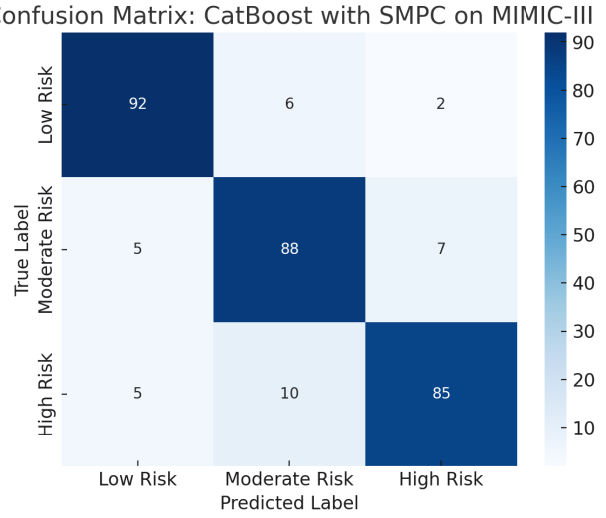


FIGURE 8. Confusion Matrix: CatBoost with SMPC on MIMIC-III dataset.

Figure 8 presents the confusion matrix for the CatBoost model with SMPC, demonstrating a balanced classification capability across severity levels. Despite the encryption overhead, the model maintained an F1-score of 0.886 and low misclassification rates across risk categories. Notably, high-risk patients were correctly classified with 85% accuracy, highlighting the system's clinical relevance.

This experiment affirms that the proposed architecture is applicable to real-world healthcare settings, capable of delivering high accuracy while preserving patient privacy

through federated learning and secure edge aggregation mechanisms.

To further evaluate the scalability of the proposed federated framework on real-world clinical data, we conducted experiments on the MIMIC-III dataset under varying client configurations: 5, 7, and 10 clients. Each client received a non-IID subset of patient records to simulate real hospital deployments. The evaluation was performed using all three classification models—XGBoost, TabNet, and CatBoost—with SMPC enabled for secure aggregation.

TABLE 13. Average precision analysis of the proposed model for severity classes on MIMIC-III.

Model	Class	Average Precision (AP)
CatBoost	Low Risk	0.89
CatBoost	Moderate Risk	0.87
CatBoost	High Risk	0.88
XGBoost	Low Risk	0.87
XGBoost	Moderate Risk	0.85
XGBoost	High Risk	0.84
TabNet	Low Risk	0.85
TabNet	Moderate Risk	0.82
TabNet	High Risk	0.81

As shown in Table 13, an increase in the number of clients leads to a modest decline in model performance across all classifiers. This is consistent with our findings on the synthetic dataset and reflects the natural complexity introduced by non-IID data partitions in federated setups. CatBoost continued to outperform other models, showing resilience even with 10 clients. TabNet showed the steepest decline, consistent with its higher sensitivity to data heterogeneity.

To evaluate model robustness in handling class-specific severity detection, we analyzed the precision-recall (PR) curves of CatBoost, XGBoost, and TabNet under the SMPC-enabled federated setup. The one-vs-rest PR curve approach was used to assess each model's ability to distinguish low-risk, moderate-risk, and high-risk classes in the MIMIC-III dataset.

As illustrated in Figure 9, CatBoost consistently maintains the highest area under the curve (average precision ≥ 0.88) across all classes, demonstrating superior class-wise balance between recall and precision. XGBoost follows closely with average precision scores between 0.84 and 0.87, while TabNet exhibits slightly lower PR stability, particularly in high-risk class detection.

This analysis reinforces the suitability of CatBoost for severity classification in privacy-preserving healthcare applications where minimizing false negatives in high-risk cases is crucial.

J. THRESHOLD ANALYSIS FOR CLIENT PARTICIPATION

To enhance robustness in federated learning under non-IID conditions, a static edge server thresholding mechanism was implemented. The threshold value was set at 0.05 variance in gradient updates, ensuring that only stable updates

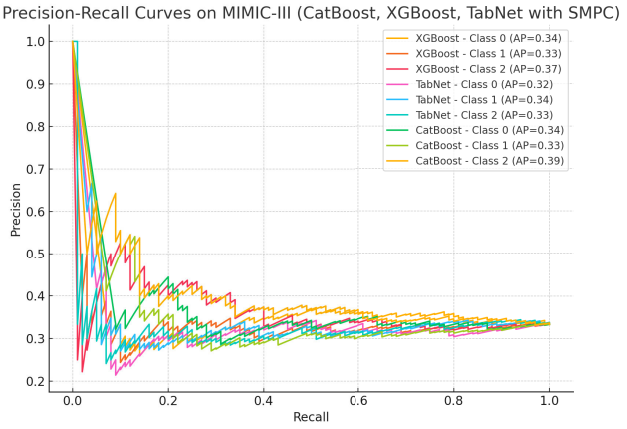


FIGURE 9. Precision-recall curves for each severity class on MIMIC-III dataset using CatBoost, XGBoost, and TabNet (with SMPC).

contributed to the global model while filtering out unreliable client updates. This mechanism aimed to mitigate noise and prevent performance degradation due to outlier gradients.

TABLE 14. Client participation across federated rounds with edge server thresholding.

Round Number	Total Clients	Active Clients	Clients Rejected
1	5	5	0
5	5	5	0
10	5	4	1
15	5	4	1
20	5	3	2
25	5	3	2

As shown in Table 14, all five clients actively participated in the initial training rounds. However, as training progressed, some clients exhibited increased variance in their gradient updates, leading to their exclusion by the edge servers. By round 10, one client was filtered out, while by round 20, two clients were no longer contributing due to unstable updates. This selective participation ensured that the global model was not influenced by erratic client updates, maintaining stable convergence.

Figure 10 illustrates the number of active clients across different training rounds. The initial participation remains high, but as some clients fail to meet the threshold criteria, the number of active participants declines. This highlights the effectiveness of thresholding in enforcing training stability while ensuring that only high-quality updates contribute to the federated model.

K. PERFORMANCE AND RESOURCE ANALYSIS UNDER DYNAMIC THRESHOLDING

To comprehensively evaluate the efficacy of dynamic edge thresholding, the classification performance of XGBoost, TabNet, and CatBoost was analyzed across multiple threshold values within the dynamically observed range (0.015 to 0.027). These values represent typical thresholds encountered

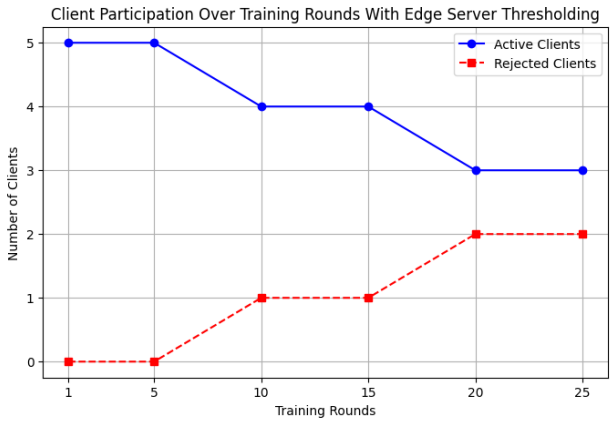


FIGURE 10. Client participation over training rounds with edge server thresholding.

at different stages of the federated training process, as derived from real-time variance statistics.

Figure 11 illustrated the evolution of threshold values across communication rounds. To further investigate the impact on model behavior, the accuracy of each model was assessed at seven specific threshold points: 0.015, 0.017, 0.019, 0.021, 0.023, 0.025, and 0.027.

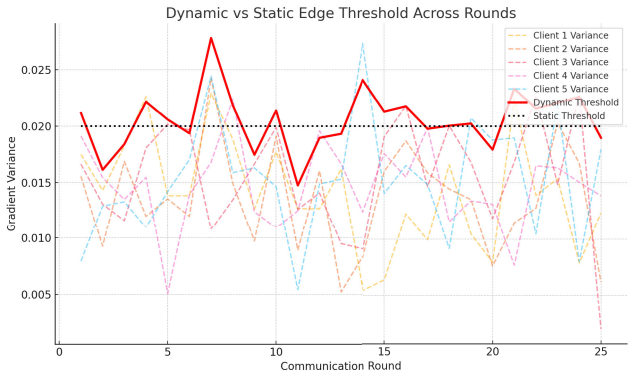


FIGURE 11. Evolution of dynamic threshold values across communication rounds.

Figure 12 presents the distribution of accuracies for each classification model at these threshold points. The results indicate that:

- CatBoost maintains high accuracy throughout the range, peaking around threshold 0.021–0.023.
- XGBoost demonstrates a slightly sharper peak, also favoring thresholds in the middle dynamic range.
- TabNet, while generally more sensitive to gradient instability, benefits from the adaptive flexibility and shows stable improvements over the lower threshold bounds.

To evaluate the impact of dynamic thresholding beyond accuracy and F1-score metrics, this section presents a detailed comparison of Average Precision (AP) for severity classification and client participation trends during federated

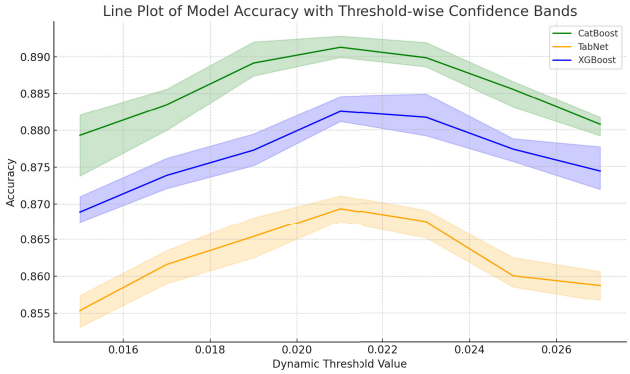


FIGURE 12. Model accuracy distribution across dynamic threshold values.

training. Results are reported for both the synthetic dataset and the MIMIC-III dataset under dynamic thresholding, complementing the previously reported outcomes under static thresholding.

1) AVERAGE PRECISION FOR SEVERITY CLASSES

Table 15 shows the Average Precision AP scores for low, moderate, and high-risk categories under dynamic thresholding on the MIMIC-III dataset. All models demonstrated a consistent improvement across all severity classes. CatBoost continued to lead with AP scores of 0.91, 0.89, and 0.90 for low, moderate, and high-risk categories, respectively. TabNet, while previously sensitive to gradient inconsistencies, showed notable improvements, particularly in the high-risk class.

TABLE 15. Average precision analysis of models under dynamic thresholding on MIMIC-III.

Model	Class	Average Precision (AP)
CatBoost	Low Risk	0.91
CatBoost	Moderate Risk	0.89
CatBoost	High Risk	0.90
XGBoost	Low Risk	0.89
XGBoost	Moderate Risk	0.87
XGBoost	High Risk	0.86
TabNet	Low Risk	0.87
TabNet	Moderate Risk	0.85
TabNet	High Risk	0.83

A similar trend was observed on the synthetic dataset, where dynamic thresholding enhanced the separability of severity classes. The higher average precision values confirm that variance-aware client update acceptance directly contributes to improving per-class discriminative performance.

2) CLIENT PARTICIPATION ACROSS TRAINING ROUNDS

Table 16 illustrates the effect of dynamic thresholding on client participation over selected training rounds. Initially, all clients contributed updates without rejection, indicating a tolerant threshold setting in early rounds where gradient

variance was higher. As the training stabilized, the dynamic threshold became stricter, resulting in the rejection of noisy or inconsistent client updates. By round 25, two clients were rejected, demonstrating the adaptive selectivity introduced by the dynamic strategy.

TABLE 16. Client participation across rounds with dynamic thresholding.

Round Number	Total Clients	Active Clients	Clients Rejected
1	5	5	0
5	5	5	0
10	5	5	0
15	5	4	1
20	5	4	1
25	5	3	2

These findings demonstrate that dynamic thresholding not only improves classification performance, particularly in imbalanced severity classes, but also contributes to more reliable model convergence by filtering out inconsistent updates over time. Compared to static thresholding, which either accepts all clients or risks early over-filtering, the dynamic approach provides a controlled and adaptive mechanism that strengthens both robustness and precision across federated rounds.

To complement the evaluation of classification metrics under dynamic thresholding, confusion matrices were generated for the CatBoost model on both the synthetic and MIMIC-III datasets Figures 13 and 14. These matrices offer a detailed view of class-specific misclassification patterns and provide insight into the model's reliability in stratifying patient severity levels in a federated setting.

In the case of the synthetic dataset, the model achieved highly accurate predictions across all severity classes, with an overwhelming majority of low-risk cases correctly classified and minimal misclassification observed in the moderate and high-risk categories. This performance aligns with the structured and noise-controlled nature of the synthetic data, where engineered features and label heuristics ensure clearer class separability.

In contrast, the MIMIC-III confusion matrix demonstrates the expected increase in misclassification, particularly between moderate and low-risk classes. A few high-risk instances were also misclassified as low-risk, underscoring the complexities of real-world healthcare data. Despite this, the model exhibited resilience by maintaining strong predictive capability and capturing the relative distribution of risk levels with acceptable accuracy.

The observed differences between datasets emphasize the importance of dynamic thresholding in mitigating noise-induced update discrepancies. By adaptively filtering client updates based on gradient variance, the system improves convergence reliability while preserving per-class sensitivity, especially in high-risk clinical scenarios where misclassification could have critical implications.

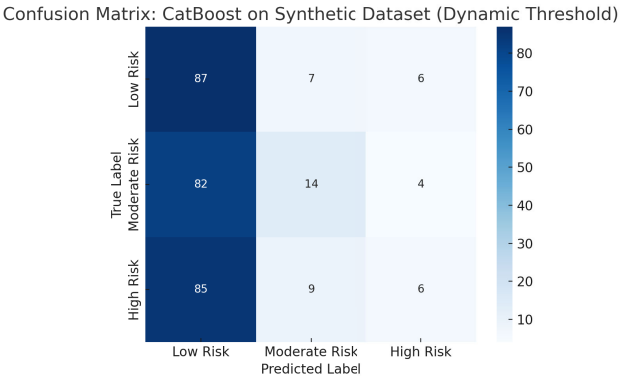


FIGURE 13. Confusion matrix for CatBoost on the synthetic dataset using dynamic thresholding.

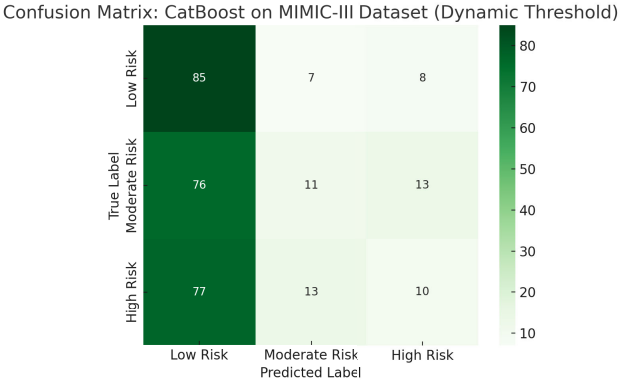


FIGURE 14. Confusion matrix for CatBoost on the MIMIC-III dataset using dynamic thresholding.

3) RESOURCE REQUIREMENTS

In addition to accuracy improvements, the dynamic threshold strategy was analyzed for its computational impact. Compared to static filtering, which performs constant-time comparisons per client, the dynamic approach introduces minor computational overhead due to per-round variance tracking and statistical computation.

Table 17 summarizes the estimated resource usage metrics under both static and dynamic settings.

TABLE 17. Estimated resource utilization: Static vs. dynamic thresholding.

Metric	Static Threshold	Dynamic Threshold
Computation Time per Round (s)	3.5	3.9
Memory Consumption (MB)	570	600
Additional Overhead (per round)	None	Variance + Std Dev calculation
Accuracy Stability	Moderate	High

The results show that dynamic thresholding provides a marginal increase in resource consumption, offset by measurable gains in model robustness and stability. These

findings suggest that dynamic thresholding is a viable and scalable addition to federated learning frameworks, particularly in healthcare or adversarial environments where model convergence and privacy integrity are critical.

L. STATISTICAL ROBUSTNESS AND CONFIDENCE ANALYSIS

To assess the robustness and statistical reliability of the proposed models under dynamic thresholding, a formal evaluation was conducted using multiple training runs. Each model CatBoost, XGBoost, and TabNet was simulated across five independent runs at the optimal threshold value of 0.021. This was done to capture training variance arising from stochastic initialization and sampling noise.

Table 18 presents the mean accuracy, standard deviation, and 95% confidence interval (CI) for each model under dynamic thresholding. These values reflect the statistical spread of the model performance and the margin of error expected across repeated experiments.

TABLE 18. Statistical summary of accuracy (5 runs, threshold = 0.021).

Model	Mean Accuracy	Std Dev	95% CI Lower	95% CI Upper
XGBoost	0.8832	0.0009	0.8821	0.8843
TabNet	0.8699	0.0011	0.8685	0.8713
CatBoost	0.8902	0.0012	0.8887	0.8917

All models exhibit narrow confidence intervals, indicating high consistency and minimal performance variance. CatBoost continues to outperform other models with the highest mean accuracy and tightest CI range, reflecting strong reliability for real-world federated learning deployments.

Figure 12 compares static and dynamic thresholding setups by plotting the mean accuracy and shaded min-max bands for each model. The dynamic threshold consistently produces higher accuracy means and tighter confidence bands across all models, especially visible in CatBoost and XGBoost, indicating enhanced robustness. This visualization provides further evidence of the proposed system's stability under non-IID and high-variance training environments, affirming the importance of integrating dynamic thresholding for achieving robust federated learning convergence.

M. MODEL BEHAVIOR UNDER SMPC AND THRESHOLDING-INDUCED CONVERGENCE DYNAMICS

In the SMPC-enabled environment, all client-to-edge communication is protected using Shamir's Secret Sharing, which introduces a level of randomness and distortion in the transmitted gradients. Among the evaluated models, TabNet exhibited consistently superior performance and lower standard deviation under SMPC settings, compared to XGBoost and CatBoost. This can be attributed to TabNet's architectural design, which involves sparse feature selection via sequential attention blocks. Such sparsity naturally leads to smoother and less sensitive gradients, thereby making the model more resilient to the noise introduced by secret sharing protocols. Conversely, tree-based models like XGBoost and CatBoost, which rely on discrete split decisions and gradient histograms, are more susceptible to distortion and may suffer from degraded update precision under SMPC masking.

To understand the influence of thresholding on training efficiency, the average number of communication rounds required for each model to reach performance convergence under three different scenarios was recorded under no thresholding, static thresholding, and dynamic thresholding scenarios with $\beta = 1.5$. As shown in Table 19, dynamic thresholding not only improved the final classification accuracy but also reduced the average number of rounds to convergence, achieving a better trade-off between model robustness and training efficiency.

TABLE 19. Comparison of convergence speed and final accuracy across thresholding strategies.

Threshold Strategy	Avg. Rounds to Convergence	Final Accuracy (%)	Std. Dev. (%)
No Threshold	21	83.2	2.1
Static Threshold	18	84.5	1.7
Dynamic Threshold ($\beta = 1.5$)	15	86.1	1.2

The results validate that dynamic thresholding effectively suppresses the influence of unstable or noisy client updates during the early phases of training, allowing the global model to stabilize faster. This is particularly beneficial in healthcare FL environments where client heterogeneity and connectivity instability can lead to erratic gradient behavior. Furthermore, the improved stability observed in the TabNet model under SMPC highlights the importance of architectural choices in designing privacy-preserving federated learning systems.

V. SYSTEM DEPLOYMENT CONSIDERATIONS AND REAL-WORLD MAPPING

In discuss the practical relevance of the proposed system, this section elaborates on how the experimental setup closely emulates a real-world federated learning deployment in healthcare, while also highlighting key challenges and mitigation strategies necessary for successful production integration.

The experimental configuration used throughout this study was carefully designed to emulate a production-ready federated healthcare deployment while remaining computationally tractable for reproducible simulation. The system consisted of five clients representing separate healthcare institutions, each locally training on a partitioned non-IID subset of the dataset both synthetic and MIMIC-III. Client updates were encrypted using secure multiparty computation SMPC with Shamir's Secret Sharing before being transmitted to the edge aggregator. Two-tier aggregation was employed, first at the

edge server level, followed by global aggregation of filtered results.

A total of 25 federated communication rounds were simulated for each experiment. Performance metrics were reported as mean and confidence intervals across five repeated runs to ensure statistical robustness. The dynamic thresholding mechanism was evaluated at multiple threshold values across these rounds, adjusting update filtering criteria in response to round-wise gradient variance.

Table 20 summarizes the key components of the experimental setup.

TABLE 20. Summary of experimental setup.

Number of Clients	5
Federated Rounds	25
Dataset Types	Synthetic (structured EHR), MIMIC-III
Data Distribution	Non-IID
SMPC Scheme	Shamir's Secret Sharing
Client Filtering	Dynamic thresholding based on gradient variance
Aggregation Hierarchy	Edge-level aggregation followed by global model aggregation
Evaluation Strategy	5 independent runs per threshold per model
Metrics Reported	Accuracy, F1-score, AP, Confusion Matrix, CI

This setup provides a high-fidelity simulation of practical federated learning in multi-institution healthcare networks, while allowing comprehensive experimentation and control over architectural components. Its structure also facilitates the seamless integration of future enhancements such as asynchronous communication, federated differential privacy, or compressed update schemes.

A. MAPPING EXPERIMENTAL SETUP TO DEPLOYMENT ARCHITECTURE

The current experimental design was structured to closely resemble real-world healthcare federated systems, where each client node represents an individual hospital or medical center possessing localized patient data. These clients do not share raw data but transmit encrypted model updates via secure multiparty computation (SMPC), ensuring compliance with privacy regulations such as HIPAA and GDPR. Edge servers simulate intermediate institutional aggregators, such as regional health clusters, which combine updates securely before forwarding to the global model coordinator.

The use of dynamic edge thresholding further strengthens this simulation by mimicking inconsistent data quality and participation — a realistic challenge in practical deployments due to heterogeneous infrastructure, patient load variation, and intermittent connectivity across clinics.

B. PRACTICAL CONSIDERATIONS FOR REAL-WORLD DEPLOYMENT

While the experimental setup provides a controlled and replicable environment, several practical factors must be addressed to ensure robust real-world deployment. These include synchronization frequency, latency handling, and managing client dropout or failure.

Federated systems must balance between frequent updates for faster convergence and sparse synchronization to reduce bandwidth and client strain. In real healthcare networks, it is advisable to implement a hybrid model update strategy:

- Fixed-frequency updates, such as once per day, suitable for tertiary hospitals with strong connectivity.
- Event-driven updates triggered by the availability of new patient batches or weekly shifts in prediction drift.
- Edge-buffered updates that allow institutions to locally aggregate multiple updates before pushing them to the global model.

Healthcare systems typically consist of geographically distributed nodes, each with varying latency. Table 21 presents an estimated breakdown of communication and computation latency per federated round based on our simulated environment.

TABLE 21. Estimated latency per communication round in edge-federated setup.

Operation	Estimated Latency (ms)
Local model computation	1200–1800
Encryption via SMPC (per client)	200–400
Edge-level aggregation	300–500
Global aggregation	500–700
Total round latency (approx.)	2200–3400

The use of hierarchical aggregation, such as edge-first then global, significantly reduces upstream communication bottlenecks, improves response time, and allows asynchronous update collection where immediate synchronization is not feasible.

Client unavailability is a common challenge in federated learning due to network issues, device failures, or compliance restrictions. The proposed system incorporates a proactive strategy through its dynamic edge thresholding mechanism, which naturally filters noisy or unstable updates and adapts participation criteria based on gradient variance.

Additionally, the system architecture can be extended to include:

- Client reliability scoring to prioritize contributions from consistently high-quality sources.
- Dropout-aware aggregation strategies, such as FedAvg with robustness weighting or asynchronous FedAsync-based aggregation for delayed participation.

While the proposed architecture has been validated under a controlled simulation with high realism, its deployment in live healthcare environments must be aligned with ethical principles of equity, privacy, and fault tolerance. Future

implementations should involve active collaboration with institutional stakeholders to calibrate update frequencies, adjust thresholds based on medical policy, and account for organizational priorities in load balancing.

By design, the proposed system minimizes data exposure, tolerates network instability, and adapts to dynamic participation, offering a solid foundation for ethically robust and technically scalable federated learning deployment in healthcare.

VI. CONCLUSION AND FUTURE PROSPECTS

This study proposes a secure and adaptive federated learning framework for healthcare severity classification that is both technically rigorous and practically deployable. The system was evaluated using both synthetic and real-world MIMIC-III datasets, simulating realistic client-level decentralization and data heterogeneity. Privacy was preserved throughout the federated training process by employing secure multi-party computation (SMPC) with Shamir's Secret Sharing, and learning convergence was governed by a hierarchical aggregation strategy that mirrors real-world edge-computing healthcare infrastructures.

A key contribution of this work is the introduction of a dynamic edge thresholding mechanism that adaptively filters client updates based on round-wise gradient variance. This approach enables the system to intelligently handle noisy or unstable clients, effectively simulating dropout and improving model resilience to training variance. The impact of this design was evaluated through multiple statistical metrics including accuracy, F1-score, confusion matrices, and average precision, all reported over five independent runs with accompanying standard deviations and 95% confidence intervals. Furthermore, the manuscript now includes an extensive discussion on deployment challenges such as model update frequency, client latency, and real-time reliability, grounded in the structure of the experimental setup. Tables outlining estimated latency and experimental configuration details further reinforce the real-world applicability of the proposed framework.

Despite its effectiveness, several opportunities for future exploration remain. Asynchronous federated learning mechanisms such as FedAsync can be considered to reduce dependence on synchronized rounds, while hybrid privacy strategies integrating differential privacy with SMPC could enhance data protection. The system can be extended to long-term deployments where threshold drift may emerge, and intelligent client reliability scoring mechanisms can be implemented based on historical behavior. Future work may also investigate attention-based client selection, adaptive update frequency scheduling, encrypted model compression, and pilot deployments in real hospital networks. Additionally, support for medical imaging, multimodal data fusion, and patient-specific personalization through local fine-tuning can expand the clinical utility of the system. Fairness evaluation across demographic groups, explainability integration for decision transparency, and reinforcement learning-driven threshold adaptation represent valuable research directions. Moreover, the framework could be adapted for vertical federated learning across diverse healthcare institutions and extended to support blockchain-based auditability, energy efficiency analysis on edge devices, and auto-thresholding using meta-learning. Finally, expanding the client base to simulate larger healthcare networks, modeling dropout with random scheduling, applying the system to regression tasks, evaluating model poisoning defense strategies, and formalizing adaptive aggregation strategies remain promising avenues for future investigation.

REFERENCES

[1] J. Jonnagaddala and Z. S.-Y. Wong, "Privacy preserving strategies for electronic health records in the era of large language models," *NPJ Digit. Med.*, vol. 8, no. 1, pp. 34–37, Jan. 2025.
[2] S. A. Tovino, "Artificial intelligence and the HIPAA privacy rule: A primer," *Houston J. Health Law Policy*, vol. 24, no. 1, pp. 77–126, 2025.
[3] A. Sood, D. Mishra, V. Surya, H. Singh, R. Sundaresan, D. Pal, R. Dharmaraju, R. Satish, S. Mishra, N. A. Chavan, S. Mondal, P. Duggal, and V. K. Iyer, "Challenges and recommendations for enhancing digital data protection in Indian medical research and healthcare sector," *NPJ Digit. Med.*, vol. 8, no. 1, pp. 48–53, Jan. 2025.
[4] C. Fang, A. Dziedzic, L. Zhang, L. Oliva, A. Verma, F. Razak, N. Papernot, and B. Wang, "Decentralised, collaborative, and privacy-preserving machine learning for multi-hospital data," *eBioMedicine*, vol. 101, Mar. 2024, Art. no. 105006.
[5] N. Li, A. Lewin, S. Ning, M. Waito, M. P. Zeller, A. Tinmouth, and A. W. Shih, "Privacy-preserving federated data access and federated learning: Improved data sharing and AI model development in transfusion medicine," *Transfusion*, vol. 65, no. 1, pp. 22–28, Jan. 2025.
[6] S. Singhal, M. Gupta, and A. Tyagi, "Transforming healthcare through advanced federated learning," in *Artificial Intelligence in Medicine and Healthcare*. Boca Raton, FL, USA: CRC Press, 2025.
[7] R. Chen, "A review of research on secret sharing," *Appl. Comput. Eng.*, vol. 88, no. 1, pp. 207–213, Sep. 2024.
[8] H. K. Kondaveeti, C. G. Simhadri, S. Mangapathi, and V. K. Vatsavayi, "Federated learning for privacy preservation in healthcare," IGI Global, New York, NY, USA, Tech. Rep., 2024, doi: 10.4018/979-8-3693-1874-4.ch009.
[9] A. Rancea, I. Anghel, and T. Cioara, "Edge computing in healthcare: Innovations, opportunities, and challenges," *Future Internet*, vol. 16, no. 9, p. 329, Sep. 2024.
[10] C. S. Mishra, J. Sampson, M. T. Kandemir, V. Narayanan, and C. R. Das, "USAS: A sustainable continuous-learning framework for edge servers," in *Proc. IEEE Int. Symp. High-Perform. Comput. Archit. (HPCA)*, Mar. 2024, pp. 891–907.
[11] R. Zong, Y. Qin, F. Wu, Z. Tang, and K. Li, "Fedcs: Efficient communication scheduling in decentralized federated learning," *Inf. Fusion*, vol. 102, Feb. 2024, Art. no. 102028.
[12] S. Mahammad and K. U. Rani, "Secure sharing of health records stored in cloud using cryptographic secret sharing schemes through computational intelligence: A review," in *Computational Intelligence in Sustainable Computing and Optimization*. Elsevier, 2025, pp. 281–301.
[13] M. F. Ahammed and M. R. Labu, "Privacy-preserving data sharing in healthcare: Advances in secure multiparty computation," *J. Med. Health Stud.*, vol. 5, no. 2, pp. 37–47, Apr. 2024.
[14] A. Rahman, M. S. Hossain, G. Muhammad, D. Kundu, T. Debnath, M. Rahman, M. S. I. Khan, P. Tiwari, and S. S. Band, "Federated learning-based AI approaches in smart healthcare: Concepts, taxonomies, challenges and open issues," *Cluster Comput.*, vol. 26, no. 4, pp. 2271–2311, Aug. 2023.
[15] K. Yuan, Z. Cheng, K. Chen, B. Wang, J. Sun, S. Zhou, and C. Jia, "Multiple time servers timed-release encryption based on Shamir secret sharing for EHR cloud system," *J. Cloud Comput.*, vol. 13, no. 1, p. 116, Jun. 2024.
[16] R. Haripriya, N. Khare, M. Pandey, and S. Biswas, "Decentralized big data mining: Federated learning for clustering youth tobacco use in India," *J. Big Data*, vol. 11, no. 1, p. 179, Dec. 2024.

[17] H. A. Ganatra, "Machine learning in pediatric healthcare: Current trends, challenges, and future directions," *J. Clin. Med.*, vol. 14, no. 3, p. 807, Jan. 2025.

[18] E. Mikołajewska, D. Mikołajewski, T. Mikołajczyk, and T. Paczkowski, "A breakthrough in producing personalized solutions for rehabilitation and physiotherapy thanks to the introduction of AI to additive manufacturing," *Appl. Sci.*, vol. 15, no. 4, p. 2219, Feb. 2025.

[19] M. T. Hossain, S. Badsha, H. La, S. Islam, and I. Khalil, "Exploiting Gaussian noise variance for dynamic differential poisoning in federated learning," *IEEE Trans. Artif. Intell.*, pp. 1–17, Feb. 2025.

[20] S. Jahan, M. R. S. Adib, S. M. Huda, and M. S. Rahman, "Federated explainable AI-based Alzheimer's disease prediction with multimodal data," *IEEE Trans. Med. Imag.*, vol. 13, no. 1, pp. 43435–43455, Mar. 2024.

[21] S. Khan, M. Khan, M. A. Khan, L. Wang, and K. Wu, "Advancing medical innovation through blockchain-secured federated learning for smart health," *IEEE J. Biomed. Health Informat.*, pp. 1–14, Jan. 2025.

[22] S. Guo, A. Zhang, Y. Wang, C. Feng, and T. Q. S. Quek, "Semantic-enabled 6G communication: A task-oriented and privacy-preserving perspective," *IEEE Netw.*, early access, Mar. 2025.

[23] C. Feng, F. Daquan, G. Huang, Z. Liu, Z. Wang, and X.-G. Xia, "Robust privacy-preserving recommendation systems driven by multimodal federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 5, pp. 8896–8910, May 2024.

[24] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 1, pp. 3454–3469, Apr. 2020.

[25] R. Haripriya, N. Khare, and M. Pandey, "Privacy-preserving federated learning for collaborative medical data mining in multi-institutional settings," *Sci. Rep.*, vol. 15, no. 1, Apr. 2025, Art. no. 12482.

[26] R. Haripriya, N. Khare, M. Pandey, and S. Biswas, "Navigating the fusion of federated learning and big data: A systematic review for the AI landscape," *Cluster Comput.*, vol. 28, no. 5, pp. 1–27, Oct. 2025.

[27] A. E. W. Johnson, T. J. Pollard, L. Shen, L.-W.-H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, "MIMIC-III, a freely accessible critical care database," *Sci. Data*, vol. 3, no. 1, pp. 1–9, May 2016.

[28] Y. Wu, Q. Zhang, Y. Hu, K. Sun-Woo, X. Zhang, H. Zhu, L. Jie, and S. Li, "Novel binary logistic regression model based on feature transformation of XGBoost for type 2 diabetes mellitus prediction in healthcare systems," *Future Gener. Comput. Syst.*, vol. 129, pp. 1–12, Apr. 2022.

[29] N. Khanh Le, Y. Liu, Q. M. Nguyen, Q. Liu, F. Liu, Q. Cai, and S. Hirche, "FedXGBoost: Privacy-preserving XGBoost for federated learning," 2021, *arXiv:2106.10662*.

[30] R. Wang, O. Ersoy, H. Zhu, Y. Jin, and K. Liang, "FEVERLESS: Fast and secure vertical federated learning based on XGBoost for decentralized labels," *IEEE Trans. Big Data*, vol. 10, no. 6, pp. 1–15, Dec. 2022.

[31] Z. Fan, J. Gou, and S. Weng, "A feature importance-based multi-layer CatBoost for student performance prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 11, pp. 5495–5507, Nov. 2024.

[32] L. Zhang and D. Jánosík, "Enhanced short-term load forecasting with hybrid machine learning models: CatBoost and XGBoost approaches," *Expert Syst. Appl.*, vol. 241, May 2024, Art. no. 122686.

[33] Y. Cai, Y. Yuan, and A. Zhou, "Predictive slope stability early warning model based on CatBoost," *Sci. Rep.*, vol. 14, no. 1, p. 25727, Oct. 2024.

[34] Z. Huang, X. Tang, H. Li, X. Cao, J. Cheng, W. Tu, and L. B.-Y. Liu, "TabSec: A collaborative framework for novel insider threat detection," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. Appl. (ISPA)*, Oct. 2024, pp. 2030–2037.

[35] S. Annam and V. Khullar, "Tabular federated learning to detect cyber faults in smart buildings," in *Proc. Inst. Civil Eng.-Smart Infrastruct. Construct.*, 2024, pp. 1–14.

[36] S. H. Alsamhi, R. Myrzashova, A. Hawbani, S. Kumar, S. Srivastava, L. Zhao, X. Wei, M. Guizan, and E. Curry, "Federated learning meets blockchain in decentralized data sharing: Healthcare use case," *IEEE Internet Things J.*, vol. 11, no. 11, pp. 19602–19615, Jun. 2024.

[37] S. S. Tripathy, S. Bebortta, C. L. Chowdhary, T. Mukherjee, S. Kim, J. Shafi, and M. F. Ijaz, "FedHealthFog: A federated learning-enabled approach towards healthcare analytics over fog computing platform," *Heliyon*, vol. 10, no. 5, Mar. 2024, Art. no. e26416.

[38] B. Annappa, S. Hegde, C. S. Abhijit, and S. Ambesange, "FedCure: A heterogeneity-aware personalized federated learning framework for intelligent healthcare applications in IoMT environments," *IEEE Access*, vol. 12, pp. 15867–15883, 2024.

[39] M. K. Hasan, N. Jahan, M. Z. A. Nazri, S. Islam, M. A. Khan, A. I. Alzahrani, N. Alalwan, and Y. Nam, "Federated learning for computational offloading and resource management of vehicular edge computing in 6G-V2X network," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 3827–3847, Feb. 2024.

[40] Y. Qi, Y. Feng, X. Wang, H. Li, and J. Tian, "Leveraging federated learning and edge computing for recommendation systems within cloud computing networks," 2024, *arXiv:2403.03165*.

[41] G. Bansal and B. Sikdar, "Achieving secure and reliable UAV authentication: A Shamir's secret sharing based approach," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 4, pp. 3598–3610, Jul. 2024.

[42] J. Chen, Z. Si, J. Song, M. Mohanty, W. Wang, and H. Xiong, "UFL: Unlinkable federated learning through shuffle and Shamir's secret sharing," in *Proc. Int. Conf. Adv. Data Mining Appl.*, Singapore: Springer Nature, 2024, pp. 239–253.

[43] A. Samanta and J. Tang, "Dyme: Dynamic microservice scheduling in edge computing enabled IoT," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6164–6174, Jul. 2020.

[44] Y.-W. Hung, Y.-C. Chen, C. Lo, A. G. So, and S.-C. Chang, "Dynamic workload allocation for edge computing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 3, pp. 519–529, Mar. 2021.

ANKITA MAURYA is currently pursuing the M.Tech. degree with the Department of Computer Science and Engineering, Maulana Azad National Institute of Technology Bhopal, Bhopal, with a specialization in advanced computing. Her research interests include advanced data privacy techniques, aiming to enhance security and privacy in modern data-driven applications.

RAHUL HARIPRIYA is pursuing the Ph.D. degree in the area of data privacy in ML models with the Department of Computer Science and Engineering, Maulana Azad National Institute of Technology Bhopal, Bhopal. His research interests include advanced data privacy techniques, particularly in enhancing the scalability and privacy of machine learning models through federated learning. His work aims to bridge the gap between privacy preservation and efficient machine learning in distributed environments.

MANISH PANDEY is currently an Associate Professor with the Department of Computer Science and Engineering, Maulana Azad National Institute of Technology Bhopal, Bhopal. With expertise in high-performance computing and data privacy, he has guided multiple Ph.D. scholars and contributed significantly to research in these domains.

JAYTRILOK CHOUDHARY is currently an Associate Professor with the Department of Computer Science and Engineering, Maulana Azad National Institute of Technology Bhopal, Bhopal. His research interests include applications of machine learning, deep learning, and artificial intelligence in various domains, including healthcare, cybersecurity, and smart systems. He has contributed extensively to interdisciplinary AI research and has guided multiple Ph.D. scholars in these areas.

DHIRENDRA PRATAP SINGH is currently an Associate Professor with the Department of Computer Science and Engineering, Maulana Azad National Institute of Technology Bhopal, Bhopal. His research interests include the development of AI and deep learning models for real-world applications, including computer vision, natural language processing, and industrial automation. His work explores innovative approaches to leveraging AI for solving complex computational challenges.

SURENDRA SOLANKI received the B.E. degree in information technology from the Institute of Engineering and Technology (IET), Devi Ahilya Vishwavidyalaya (DAVV), Indore, and the M.Tech. and Ph.D. degrees in computer science and engineering from Maulana Azad National Institute of Technology Bhopal, Bhopal, India. He is currently an Assistant Professor with the Department of Artificial Intelligence and Machine Learning, Manipal University Jaipur, India. During his Ph.D. studies, his research focused on deep learning-based spectrum sensing for cognitive radio networks. With over seven years of research experience, his research interests include machine learning, deep learning, wireless networks, and cognitive radio. He has published impactful research articles in reputed SCI-indexed journals and holds a patent in the field.

DUANSH SHARMA is currently pursuing the B.S. degree in computer science with the School of Arts and Sciences, Rutgers University, USA. He has good command over data analysis and machine learning algorithms and expertise in data visualization along with real-time ML model deployment.

Generative Federated Learning with Small and Large Models In Consumer Electronics for Privacy-preserving Data Fusion in Healthcare Internet of Things

Taher M. Ghazal, *Senior Member, IEEE,* Shayla Islam, *Senior Member, IEEE,* Mohammad Kamrul Hasan, *Senior Member, IEEE,* Ahmad A. Abu-Shareha, Umi A. Mokhtar, M. Attique Khan, *Member, IEEE*, Jamel Baili, Ali Q Saeed, Mohammed Wasim Bhattt, and Munir Ahmad

***Abstract*—Healthcare Internet of Things (HIoT) requires large-scale privacy features to ensure maximum security in sharing sensitive physiological data in consumer electronics. Recent approaches utilize the fusion concept to provide maximum privacy in health data sharing. Embedded signing data fusion with the health observed data ensures privacy preserved sharing across heterogeneous medical consumer devices for diagnosis. This article proposes a Dependency-correlated Data Fusion Scheme (DcDFS) to maximize the privacy of the health data-sharing process. The proposed scheme prepares separate key signing procedures using triple-DES (data encryption standard) to embed with the accumulated health data. The fusion process is carried out by defining key headers and integrity footers for authentication and verification. Therefore, the fusion generates a combined sequence of linear authentication and validation procedures enclosing the health data. In this scheme, the role of federated learning is to prevent permuted sequences for the same health data. This research integrates Small Language Model (SLM) and Large Language Model (LLM) into the federated learning module to support secure, scalable, and intelligent healthcare data sharing. Their collaboration enhances context-aware training while preserving privacy across decentralized, encrypted environments. A similar sequence mapped between the header and footer is responsible for discarding unauthorized data handling. The learning process verifies the permutation for many-to-one header to footer and vice versa. Therefore, the proposed fusion scheme generates a linear dependency between the actual and security-related data for maximum privacy. The proposed scheme achieves the following: the computation time is confined by 12.424%, the privacy leakage by 12.923%, and the computation efficiency is improved by 11.46%, as observed under the maximum sequences.**

***Index Terms*—Data Fusion, Federated Learning, Healthcare IoT, Privacy-Preserving, Triple DES.**

I. Background

Block-based ranked retrieval protects complex consumer electronics healthcare IoT data systems by grouping data into encrypted cloud-based chunks. The approach improves privacy by assuring data integrity and limiting unauthorized access to sensitive healthcare information [1]. Advanced encryption algorithms reduce computational load while ensuring good retrieval accuracy. Scalability and adaptation to complicated IoT contexts make the technique appropriate for large-scale healthcare systems [2]. Efficient encryption allows seamless integration with data protection rules that preserve patient information. The architecture prioritizes real-time data access while maintaining security and utility [3]. By encrypting at granular levels, the technology outperforms existing methods regarding dependability and compliance. Its modular structure ensures consistent performance across multiple datasets and usage scenarios [4]. Security measures lower the risk of breaches while ensuring system reliability under peak operating demands. Compatibility with modern healthcare IoT systems provides a broad range of potential applications [5]. An increase in the need for cross-domain analytics that protect patients' privacy has coincided with the explosion in healthcare data [6].

A decentralized privacy-preserving system improves data

"This work was supported bythe Centre of Excellence for Research, Value Innovation, and Entrepreneurship(CERVIE) at UCSI University for funding this research project through the Research Grant with the project code: T2S-2024/008, and T2S-2025/008. The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through Large Research Project under grant number RGP.2/275/46.

Corresponding author: Shayla Islam (e-mail: shayla@ucsiuniversity.edu.my); Mohammad Kamrul Hasan (e-mail: hasankamrul@ieee.org; Munir Ahmad (e-mail: munirahmad@ieee.org).

T. M. Ghazal is with the Center for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), and Department of Networks and Cybersecurity, Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman, Jordan. (e-mail: taher.ghazal@ieee.org)

Shayla Islam is with Institute of Computer Science and Digital Innovation, UCSI University, 56000 Kuala Lumpur, Malaysia. (e-mail: shayla@ucsiuniversity.edu.my)

M.K. Hasan, and U.A. Mokhtar are with the Center for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM). (e-mail: hasankamrul@ieee.org; umimokhtar@ukm.edu.my).

Ahmad A. Abu-Shareha is with Department of Data Science and Artificial Intelligence, Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman, Jordan. (e-mail: a.abushareha@ammanu.edu.jo).

M. Attique Khan is with Center of AI, Prince Mohammad bin Fahd University, Saudi Arabia.(e-mail: mkhan3@pmu.edu.sa).

Jamel Baili is with Department of Computer Engineering, College of Computer Science, King Khalid University, Abha 61413, Saudi Arabia. (Jabaili@kku.edu.sa)

Ali Q Saeed is with the Computer Center, Northern Technical University, Ninevah, Iraq. (e-mail: ali.qasim@ntu.edu.iq)

Mohammed Wasim Bhatt is with the Model Institute of Engineering and Technology, Jammu, India (wasimmohammad71@gmail.com)

Munir Ahmad is with College of Informatics, Korea University, Seoul 02841, Republic of Korea. (e-mail: munirahmad@ieee.org)

fusion in healthcare IoT by combining blockchain and cloud technologies [7]. The technique combines encrypted data storage with real-time secure access to maintain patient data confidentiality. A layered blockchain paradigm prohibits unauthorized data alteration and allows for secure data sharing across stakeholders [8, 9]. Decentralized control eliminates single points of failure and improves system reliability across many IoT healthcare environments. Optimized blockchain algorithms reduce computing complexity, allowing smooth operations on big databases [10]. By using cloud-based technology, the technique provides scalability and adaptability to future IoT breakthroughs [11]. The design adheres to strict data protection rules and assures safe and ethical data use. Advanced encryption ensures that all transactions are tamper-proof, preserving system integrity [12]. Secure multi-party computation enables efficient collaborative analysis while protecting sensitive information. Real-time data fusion enables better clinical decision-making while protecting privacy [13]. Integration with existing healthcare systems ensures a smooth transfer and broad applicability. The system provides strong security, privacy, and usability, which makes it an attractive option for modern IoT healthcare [14].

Federated learning (FL) allows privacy-preserving data fusion in healthcare IoT by distributing model training without exposing raw data [14, 15]. The method protects patient data while optimizing machine learning (ML) models via collaborative training. It decreases the likelihood of privacy breaches by guaranteeing that sensitive data is stored locally at each node. Advanced techniques improve learning efficiency, producing excellent model performance with a low computing load [15, 16]. The scalable architecture makes it ideal for large-scale IoT healthcare systems with several data sources. Adaptive optimization ensures the secure merging of multi-source healthcare data, which improves system functionality [17]. The technique complies with privacy standards and allows for ethical usage of patient data. Decentralized learning enhances system reliability by reducing risks from single points of failure [18]. Enhanced encryption and obfuscation techniques help to safeguard the data processing pipeline. The architecture enables secure multi-party cooperation while maintaining model performance and data utility. Real-time integration of federated models speeds up clinical decision-making while maintaining privacy. Evaluations indicate the method's efficacy in protecting privacy and improving operational efficiency for healthcare IoT [11, 19]. The contributions of the article are listed below:

- The introduction and discussion of dependency-correlated data fusion scheme to improve the Privacy of healthcare Internet of things consumer devices using triple DES and federated learning process.
- Different steps are incorporated in generating DES keys with sequence verification and header footer generation. The generated header and footer authentication credentials are fused with the cipher text to ensure maximum privacy authentication
- The discussion of the fusion process that merges the authentication and verification credentials with the health data accumulated. The fusion credential sequences are verified using the FL on each update across various sharing intervals.
- The discussion using internal parameters and common metrics is presented as a comparative analysis with the existing methods later in the article.

II. Related Works

Muazu et al. [20] developed a federated learning (FL) system with data fusion for healthcare applications. The developed model combines multiple sensor data to produce relevant data for diagnosis services. Multi-party computation and secret sharing techniques are employed here to protect original sensor data from attacks. The developed model improves the overall performance range of the applications. Hu et al. [21] introduced a privacy-aware framework for cyber-attack detection in the Internet of Medical Things (IoMT). The framework uses data fusion and quantum deep learning, which are implemented in the framework to fuse the data. The fused data is used as input, which minimizes the computational cost of the systems. The introduced framework enlarges the accuracy and efficiency rate of the attack detection process. Khan et al. [22] designed a federated reinforcement-based fusion model for privacy protection to detect the exact cause of threats, analyze the challenges, and provide necessary services to improve medical data's privacy and security rate.

Han et al. [23] proposed a lightweight and smart data fusion approach for wearable devices in IoMT. The necessary data is collected from devices via wireless sensors. The collected data is evaluated and analyzed to extract optimal information for the data fusion process. The approach minimizes the time and energy consumption level of the fusion process. The proposed approach maximizes the accuracy and precision rate of data fusion services. An improved version of [20] was developed by Khadidos et al. [24] using Tasmanian and Lichtenberg optimized attention deep convolution (TasLA) for IoT and introduced for fuse diagnosis and medical data from healthcare systems. Fan et al. [25] introduced an authentic and privacy-preserving scheme for E-health data transmission. Regular and forward-secure signature authentication is analyzed to identify the network's security measures.

Li et al. [26] proposed an efficient privacy-preserving technique using blockchain (BC) and lightweight data sharing to solve the problems in IoMT. The method provides secret structure and construction services to medical data, which secures it from attacks. The proposed technique improves efficiency and maximizes fault tolerance and storage space. An enhanced version of [26] is designed by Alsuqaih et al. [27] using BC technology. The designed method is used in e-healthcare applications to ensure the privacy and security of users' data and provide necessary data exchange policies to reduce medical data leakage. The target is to eliminate problems that are raised during data transmission services. Li et al. [28] developed a new privacy-preserving technique using

BC and FL. The method is used to construct a strong scheme to improve the security and privacy rate of patients' medical and healthcare data. An improved nose selection is used here, which selects sensor nodes for healthcare data sharing. Compared with others, the developed technique reduces the data leakage ratio of the systems.

Makhdoom et al. [29] introduced a privacy-preserving scheme for data sharing using distributed ledger technology (DLT) is employed here to secure the data-sharing services from one user to another and ensures data privacy from third parties, which minimizes the computational cost rate of the sharing process, also elevates the privacy and security level of IoT ecosystems. The study suggests that this scheme has a higher computational cost. Meng and Li [30] proposed an edge computing-based privacy-preserving approach (EBPPA) for smart healthcare systems. The proposed approach is used in IoMT to address the challenging security threats in the systems. A homomorphic encryption technique is implemented to secure sensitive and confidential medical data from attackers. Experimental results show that the proposed EBPPA maximizes the accuracy and efficiency range of IoMT. The study still has a computational overhead of encryption. Akhtar et al. [31] developed an IoMT-based smart healthcare monitoring system. An adaptive wavelet entropy deep feature fusion and improved recurrent neural network (I-RNN) algorithm are used in the systems. The developed system is used to monitor and collect healthcare data via a monitoring system. The system eliminates unwanted threats that cause data privacy and security issues. The developed system enlarges the performance and effectiveness rate of the networks. The study needs to be tested for optimized performance in terms of threat elimination.

An enhanced version of [25] is designed by Abaoud et al. [32] using the FL algorithm, which collects the decentralized data and trains the data to get relevant information for data securities and reduces the computational cost and latency level of the systems and maximizes the privacy and security range of healthcare data. The study still faces the high latency in FL model training. Liu et al. [33] introduced an energy-efficient and privacy-preserved incentive mechanism for mobile edge computing-assisted FL, ensuring patient data safety to provide a safe data rate from attackers and reducing complexity. However, the implementation may face complexities. Xu et al. [34] developed a new privacy-preserving medical data-sharing scheme using BC for IoT. Attribute-based encryption (ABE) is employed here with BC, which designs optimal structures for data sharing to maximize the safety of sensitive data during data-sharing services and access control to the data to reduce the data loss ratio and improve the overall accuracy of data-sharing services in IoT. However, it has access control scalability issues.

Liu et al. [35] proposed an optimal dimensional privacy-preserving data aggregation (OPERA) for smart healthcare systems that analyzes the data, provides dimensions for sharing with the users selecting the vectors, and uses homomorphic cryptography to ensure the data is safe. Yet, the high computational cost is not solved. An improved version of [33] introduced by Tian et al. [36] named the robust and privacy-preserving decentralized deep FL (RPDFL) scheme. The RPDFL scheme improves the application's overall communication efficiency level. The scheme also provides relevant services to secure the data from threats and leakages. Compared with others, the introduced scheme increases healthcare applications' accuracy and security range. Namakshenas et al. [37] suggested using Additive Homomorphic Encryption (AHE) and a quantum-centric registration and authentication strategy for Federated Learning (FL) in Consumer IoT to guarantee strict client validation demonstrates strong threat detection across various client settings, with 94.93% accuracy using the N-BaIoT dataset and 91.93% accuracy using the Edge-IIoTset dataset. However, optimizing for real-time IoT applications is necessary because there are still issues in dealing with quantum-resistant adversarial attacks and computational overhead from AHE.

A privacy-preserving FL model suggested by Yazdinejad et al. [38] uses an internal auditor with Gaussian Mixture Model (GMM) and Mahalanobis Distance to detect harmful encrypted gradients and AHE for secrecy achieves higher accuracy with lower computational and communication overhead on benchmark FL datasets than Fully Homomorphic Encryption (FHE) and Two-Trapdoor HHE. Scaling large FL networks and fine-tuning detection thresholds for hostile gradient changes remain problems. However, it has a higher computational overhead of detection mechanisms

Yazdinejad et al. [39] proposed the Auditable Privacy-Preserving Federated Learning model, which uses TEE for secure aggregation, ActPerFL for Non-IID data, and BN for data similarity detection. Our methodology improves privacy, integrity, and fairness in healthcare datasets, exceeding FL techniques in accuracy and security. Scalability, TEE computational overhead, and real-time flexibility for varied healthcare applications remain problems. However, it shows complexity in agent coordination.

Nazari et al. [40] proposed a, privacy-preserving provenance graph-based GNN model that integrates FL with Graph Convolutional Networks for Advanced Persistent Threat (APT) detection in Software-Defined Networking (SDN). P3GNN improves data privacy and zero-day threat detection via homomorphic encryption and unsupervised learning, attaining 0.93 accuracy and 0.06 false positives on the DARPA TCE3 dataset. However, it has a processing burden from encrypted gradient exchanges. Scalability and processing burden from encrypted gradient exchanges remain issues for big SDN networks [41]. However, it suffers complexity in agent coordination

Bharathi, M., & Srinivas [48] employed Differential Privacy (DP) and safe aggregation to train models collaboratively without sharing raw data. FL has been scaled by Google and Meta, using differentially private privacy-preserving approaches followed by the regularized leader to improve security. Verifying server-side DP guarantees, addressing diverse device participation, and extending FL to large multi-modal models and tailored training remain problems. The FL

framework must be more flexible and use trusted execution environments and open-source collaboration to advance.

Pujari & Pakina [52] examined the use of SLMs in FL frameworks for privacy-preserving AI on edge devices. According to the study, SLMs offer efficient local inference while minimizing data exposure in TinyML applications like natural language comprehension and anomaly detection. Despite these encouraging achievements, edge devices face security concerns, computational limits, and energy constraints. The report also emphasizes the need for governance regulations to balance technical innovation with data protection, noting a gap in federated learning environment security and calling for greater research, policy, and technology to address it.

Chen et al. [53] investigated integrating LLMs with FL to improve dispersed data training and task generalization. Combining LLM sub-technologies with FL, integrating FL with LLMs, and fusing both comprise the research framework. The study covers its applications in sensitive industries like healthcare, banking, and education, but key issues include data quality, privacy, and scalability for large-scale deployment in real-world scenarios.

Social media data were analyzed for mental health illnesses using Hugging Face's pre-trained language model BERT, using decentralized learning for privacy [54]. Anonymous social network interactions with behavioral health markers are included. The research finds great accuracy in identifying depression and self-harm, but real-time adaptation and social media data biases are limitations.

III. Data Collection

The data used in this article is inherited from electronic health record data[51]. The tabular data is extracted as text connecting personal data, procedural (medical), and visit-based occurrences. The information is accumulated from the sender, and clinical monitoring procedures are followed. A combined illustration of the data and security model is presented in Fig. 1.

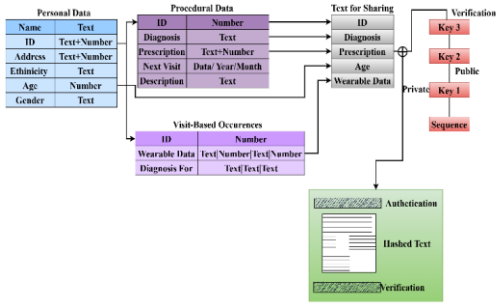


Fig. 1. Data and Security Model Combination.

The personal, procedural, and visit-based occurrence data is adapted and represented as a consolidated text at the initial stage. This text is presented in tabular format for precise inputs across various observation intervals and diagnosis entries. This text data is hashed using triple $D \in S$ with private, public, and verification keys. The key (cumulative) size is 168 bits for authenticating shared data. Privacy is administered if the authentication, verification, and data sequence are linear (Fig. 1).

IV. Proposed Dependency-correlated Data Fusion Scheme

In the proposed DcDFS framework, FL is enhanced through a dual-model integration strategy involving SLMs at the edge and LLMs at the server. This fusion significantly strengthens privacy preservation, semantic understanding, computational efficiency, and adversarial resistance across the training lifecycle. SLMs are deployed at clients such as healthcare institutions, IoT-enabled medical devices, or edge servers. Their primary function is to semantically encode raw, unstructured data such as clinical notes, symptoms, and records into low-dimensional embeddings before the 3DES encryption process [55-58]. The LLMs operate at the central server to perform semantic reasoning, integrity validation over the aggregated update vectors, and latent metadata like header and footer. The proposed fusion concept is presented for the header and footer, and the authentication key is used for the header, whereas the footer refers to the verification key. Here, the sequence's role is to analyze the data by finding the mismatching and permuted data. The permuted data is defined as the header and footer not being on identical sequences (i.e., different sequence values). This is sorted out by generating a private key from the receiver, ensuring it is non-permuted. Privacy-preserving sharing on heterogeneous devices is used for diagnosis, which is accomplished using fusion. Based on this discussion, the proposed scheme is introduced in Fig. 2.

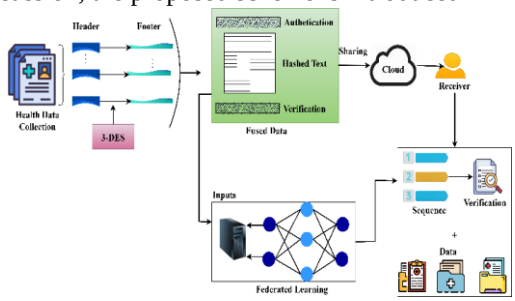


Fig. 2. Proposed Data-Fusion Scheme**.**

Here (Fig. 2), the triple Data Encryption Standard (DES) is focused on the header and footer keys, and fusion is utilized. The authentication and verification are processed from the data fusion and shared the data to the cloud environment, where the data is received as the sequences and verification are generated. The encrypted data, such as cipher text, are decrypted on the receiver side with the private key to ensure security. Then, the input for the fused data is given as the input for federated learning. Here, it indicates the sequence and data separately and verifies the data. The preliminary step discusses the objective of healthcare data in IoT.

$$Eva_0 = \frac{1}{Hd'} * \sum_{He'}^{Fo_t} Au_0 + [(Se' \to R_v) + y_S] * \sum_{y_S} Se'(Q_u) * (Au_0 + Ve') * Hd' + \{[(Au_0(Hd' + He'))) * (Ve'(y_S) + Q_u)] * R_v\} + \sum (Hd' * He') + R_v * \{[(y_S + Ve') * Se'] + Hd'\} \quad (1)$$

The evaluation is done on the above equation and it is represented as Eva_0, Hd' is described as healthcare data. The sender and the receiver are symbolized as $Se' and\ R_v$, here

$He' and\ Fo_t$ are represented as header and footer. Here, authentication is labeled as Au_0, y_S is the security, the sequences are Q_u. The authentication and verification are represented as $Au_0\ and\ Ve'$. The healthcare data in IoT is used to represent security through fusion using the header and footer that validate authentication and verification. The sequences are checked on the authentication and verification process.

The end-user requests the health data, and the sender encrypts it and sends it to the receiver. The receiver opens the cipher text with the private key. The key generation is discussed on the 3-DES algorithm, which indicates the authentication and verification of the header and footer. Both the header and the footer must be on the same sequence. If one differs, then permuted occurs. To solve this permuted mismatch of data with the private key, the 3-DES algorithm is used to embed with the accumulated health. The 3-DES processing is discussed in the below section.

A. Triple DES Process

Triple DES is processed using multiple encryptions to improve healthcare data security; the encryption is done three times, so it is named 3 DES. It is a block cipher that encrypts 64 bits of data with three keys, authentication and verification, which act as one private and public key. These keys ensure privacy while sharing the data with the end-user. It includes four steps of computation, which are listed below.

B. Key generation

Establishing a robust key management procedure safeguards all data from unauthorized access. At the same time, it is being prepared for fusion, resulting in a secure pipeline from data collection to transfer. It is the initial step in 3-DES, where three unique keys are generated using a key derivative method (KDM). The key is derived for encryption, authentication, and fusion parameters (discussed below). The following equation is used to analyze the key generation.

$$\begin{aligned} KG_0 = B_i(Q_u + EVa_0) * \prod_{Hd'} Va' + y_S * |y_S + hx'| * \frac{Hd'}{\prod_{hx'}(He' * Fo_t)} \\ + \{[(R_v * cry) * (hx' + Se')] * (Au_0 + Ve')\} \\ + \prod_{Se'} R_v * [(hx' + Hd') * Q_u] \\ + \{[(cry * Hd') + hx']\} * \prod_{Hd'}^{Shr} (R_v - y_S) \end{aligned} \quad (2a)$$

Key Generation in 3-DES involves deriving three keys k_1, k_2, k_3 using a key derivation method using KG_0 is provided as intermediate form as $B_i(Q_u + EVa_0)$ as K_{base} with the encryption dependency $[(R_v * cry) * (hx' + Se')]$ and multiplication by authentication components $(Au_0 + Ve')$. Followed by applying a sequence adjustment using secret-sharing component $[(cry * Hd') + hx'] * \prod_{Hd'}^{Shr}(R_v - y_S)$ yields a final key generation output. The key generation is equated above and it is represented as KG_0, Va' is the private key whereas, B_i is labeled as the public key, hx' is the cipher text, the encryption is described as cry, Shr is sharing. The key is generated before sending the data to the receiver, to ensure privacy, on the receiver side private key is generated to open the cipher text to plaintext. This is how key generation is processed.

C. Initial permutation

Permutation is an arrangement of data in a definite order, which is arranged in a sequence. The rearrangement is evaluated for the plaintext according to the predefined permutation table. The below equation is derived for the initial permutation. The initial permutation (IP) rearranges the plaintext bits before encryption by computing key-based permutation factor [15] $Se' * \big(Au_0 + pr(tab)\big)$, then compute transformation based on data sharing using sum over header values $\sum_{Hd'}^{hx'} R_v * (Ve' \to R_v)$. Followed by multiplying the key diffusion factors $(He' * Fo_t) * \{[(hx' + Hd') + (cry)]\}$ ensures data scrumbling. For computing the effect of permutation on keys using $\sum_{Se'}^{R_v} He'\big(hx' * pr(tab)\big) + |(KG_0 + cry) * Ve'|$, then calculate the final plaintext transformation $\sum_{y_S}^{pr(tab)} Q_u * (Hd' + Eva_0) * x_p + C_y$ finalizes the permutation adjustment

$$\begin{aligned} Ang = Se' * \big(Au_0 + pr(tab)\big) * KG_0 + \sum_{Hd'}^{hx'} R_v * (Ve' \to R_v) + (He' * Fo_t) \\ * \{[(hx' + Hd') + (cry)]\} \\ + \sum_{Se'}^{R_v} He'\big(hx' * pr(tab)\big) + |(KG_0 + cry) * Ve'| \\ + \sum_{y_S}^{pr(tab)} Q_u * (Hd' + Eva_0) * x_p + C_y \end{aligned} \quad (2b)$$

The analysis is labeled as $Ang, pr(tab)$ is the permuted table, the plaintext is represented as x_p whereas, C_y is the privacy. Here, it defines the authentication for data sharing between the sender and the receiver. The plaintext is obtained by using the private key to open the cipher text. The analysis is computed to rearrange the data if it is not in an ordered form, and the results header and footer are not in a sequence. Data fusion with authentication procedures improves the reliability of the data. Forming a secure trust connection is vital in healthcare applications, and authentication technologies let the recipient validate the source and integrity of the data while encryption secures the content. The approach ensures data privacy and improves model performance via dispersed learning insights; this allows for continual development without storing sensitive health data centrally, all thanks to federated learning and the secure transmission of fused data. The permutation sequence detection is described in Procedure 1.

Procedure 1 Permutation Sequence Detection

Input: The health data where the data fusion takes place for the header and footer which are maintained on the same sequences $U_\alpha + (Au_0 + Ve'), for\ all\ U_\alpha \in Hd'$ and ensures the security and privacy. //The permutated table is obtained from the data fusion $U_\alpha \in Hd'(y_S)$

Output: The permutation table update

1. Initialize the health data and detect the permutation
2. // Key generation is done by sharing the data
3. For $KG_0(Shr + Hd')$ is given to the receiver $R_v * C_y$ do
4. $pr(tab) + Q_u(KG_0)$//The permutated table with key sequences is generated and stored with security.
5. // The fusion parameter is used to find the permutation sequences
6. For every data sharing requested by the receiver $Hd'(Shr)$ do
7. Evaluate the encryption during data sharing $cry(Shr + Hd')$

8. For $cry(hx' + Hd')$ do
9. An analysis is done for the encryption process $Ang(cry) + KG_0$ which is equated in equation (4).
10. end for
11. Encryption is processed for every sharing process with sequences of key
12. end for
13. The evaluation is done for the health data and updates the permutation table $pr(tab) + DP_t$
14. The sequence on the table is updated in equation (8c)
15. Periodically, the sequences update is evaluated for the header and footer for security purposes $DP_t(Q_u + Hd')$
16. end for

For improving privacy, integrity, and authentication, FL is used to ensure that only confirmed and authenticated data goes into FL model updates. In healthcare applications, for example, Taiello et al. [43] demonstrated the efficacy of secure aggregation procedures in protecting privacy without compromising task accuracy by implementing them within the Fed-BioMed framework. Furthermore, a system combining Differential Privacy with FL was suggested to safeguard sensitive health data[44-45]. This approach achieves high privacy guarantees against a variety of attack scenarios.

Apply SLM for lightweight tasks that require fast processing and lower resource consumption, such as simple answering or structured data processing. Also, deploy LLM for complex tasks like natural language understanding and contextual analysis that require deeper comprehension and a larger dataset. Each device trains its local instance of the model, either SLM or LLM, using locally stored data without sharing this data with other devices on the central server. After training, each device generates encrypted model updates using advanced encryption methods like 3DES that encapsulate the learnings from the local data while preserving privacy.

D. Three round Process

This step is important in 3 DES, which takes 48 rounds of encryption. The plaintext is evaluated three times and encrypted. Each time, a different key is generated for encryption, which is equated below.

$$EVa_0(cry) = x_p\big(pr(tab)\big) * Se' \rightarrow R_v$$
$$x_p = hx'(B_i + Va') * \sum_{Ang} R_v(y_S) + KG_0$$
$$= KG_0\big(Va' * pr(tab)\big) + EVa_0 - x_p$$
$$= x_p + (hx' * R_v) * pr(tab) - Fo_t \qquad (2c)$$

The permuted table ensures the security between the sender and the receiver, for these three rounds of encryption are processed. Here, the encryption is done on the cipher text whereas the plaintext is processed three times to obtain encryption. The authentication using He' and Fo_t is illustrated in Fig. 3. In the He' and Fo_t utilization process the Se' data is hashed, encrypted, and then shared with the receiver for diagnosis. The key generation follows three consecutive rounds for $KG_o \rightarrow Va \rightarrow B_i$ and verification. This process follows Ang and encrypted hx' for Au_o. The authentication first generates $cry(Hd')$ for all x_p. Depending on the generated He' generated, the $pr(tab)$ is constructed; the entries include (Q_u, KG_o, shr, Ang).

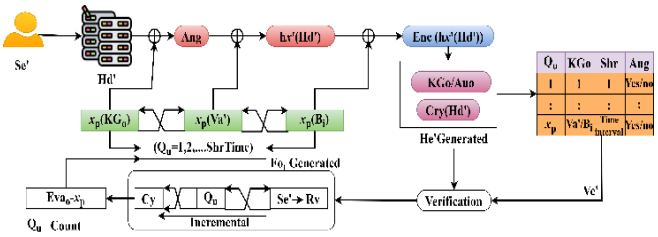


Fig. 3. Authentication using He' and Fo_t.

If Q_u is replicated in either $KGo'Va'$ (or) B_i generation, then $Ang = true$ for the permutation. Therefore the current sequence is not secure for Shr. If $Ang = false$, then $Se' \rightarrow R_v$ is the incremental demand to ensure Q_u is not replicated in both Va' and B_i. Based on the output of the above, the Fo_t is generated for the hx' generated. If Q_u to c_y administration is incremental, then $EVa_o - x_p$ (count) is reduced by 1else $pr(6b)$ is added with the current Shr entry. Thus, $\left[cry(Hd') \rightarrow hx'\|va'\|B_i\|Ve'\right]$ forms the data for Shr through authentication (Refer to Fig. 3). In Fig. 4, the Q_u evaluation for the three rounds of KG_o is given. Real-time communication capabilities are greatly improved by establishing trust through authentication checks and providing security through encryption. Healthcare providers may access protected patient information in a timely manner because this integrated approach supports better clinical decisions and patient outcomes.

The sequence evaluation is higher by improving the authentication and verification of health data. This computation is used to provide security so key generation is used $KG_0(Va' + B_i), for\ all\ Au_0 \in C_y$ and gives privacy to the data. Here, the same sequences are used for the authentication and verification $Q_u + (Au_0 + Ve') * \sum_{Hd'}(cry + y_S)$ to evaluate the encryption for data. Here, $Hd'(C_y)$ is done to reduce the adversaries' access to the data $Au_0 + Ve' > U_\alpha$ (Fig. 4a-c).

E. Final permutation

It results in the ciphertext reaching the final permutation, which is the inverse process of the initial permutation. The output results in the bits of ciphertext to obtain the original format and it is formulated below.

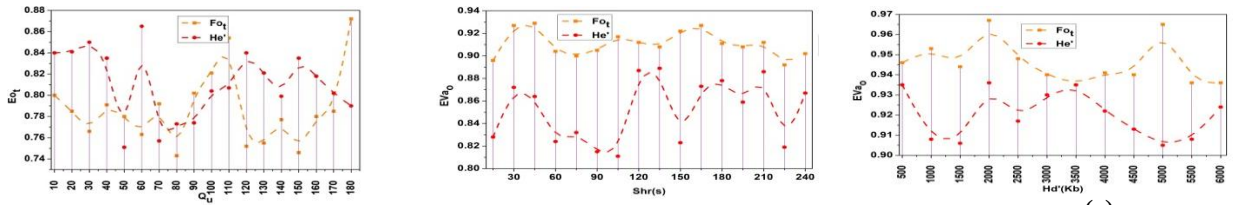
$$Ang(FP_a) = \prod_{Hd'}^{Au_0} Ve'(Fo_t * He') + EVa_0 + y_S(cry) - hx'$$

The above equation is rewritten as,

$$y_S(cry) = Au_0 * \left(R_v \leftarrow Se') + \frac{hx'}{KG_0}\right) - Ang$$

The encryption is evaluated for the key generation process which is the inverse process of initial permutation.

$$= Ang\big(hx' * pr(tab)\big) + Q_u * C_y$$
$$= C_y(Ang + Eva_0) + hx' * \sum_{cry}(KG_0 + Shr) \qquad (2d)$$



(a) (b) (c)

Fig. 4. Sequence Evaluation Analysis: (a) No. of Queries Q_u, (b) Sharing Time $Shr(s)$, (c) Size of the health data Hd'(Kb)

The analysis for the final permutation is computed above and it is described as FP_a, the sharing is done for data to ensure the security. Since it is an inverse process, the cipher text is reversed to the original data. From the above equation, below comes the evaluation process for the header and footer that includes authentication and verification, and it is expressed in the following equation.

$$EVa_0(He', Fo_t) = Ang(pr(tab) * y_S)$$

$$= \sum_{Shr}^{KG_0} R_v * (hx' + Se') * x_p(Q_u) + \sum_{KG_0} C_y * (Au_0 + Ve')$$

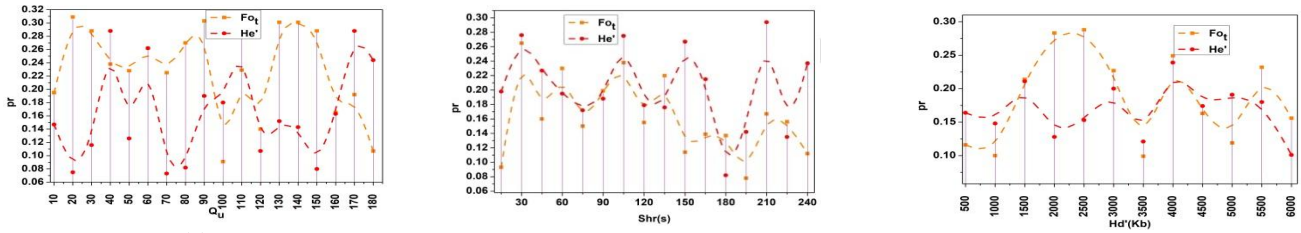
Similarly,

$$= FP_a(Shr + Hd') - R_v + y_S$$

$$= \{[(cry + x_p) * (hx' + Se')] * Q_u\}$$

$$= \sum_{FP_a} Shr * (C_y + R_v) - cry + hx'(Au_0 + Ve') \quad (3)$$

The evaluation is done for the header and footer where the authentication and verification are computed in the sequences. Here, the encryption is computed three times and evaluates the same sequences for authentication and verification. In Fig. 5a-c, the replicated permutation for the variants is analyzed. The replicated permutation decreases by using permuted table $pr(tab) + Shr(Hd'), for\ all\ Hd' \in y_S$ and address the replicated data. The computation takes place by ensuring the privacy and security $(C_y + y_S) * \sum_{p_s}^{U_\alpha} KG_0 > \sigma_a$ and evaluates the correlation factor.



(a) (b) (c)

Fig. 5. Replicated Permutation for Variants:(a) No. of Queries Q_u, (b) Sharing Time $Shr(s)$, (c) Size of the health data Hd'(Kb).

The sequences of authentication and verification hold the same key $KG_0(Q_u) * hx'(x_p) - Va', for\ all\ Q_u \in (Au_0, Ve')$, and improve the data security (Fig. 5a-c). From this fusion is done for data authentication and verification which generates the hash data.

F. Data Fusion Process

The data fusion is a combination of header and footer, including the authentication and verification key, which are the same sequence.

Header and Footer Integration

The integration of authentication and verification data have the same key, and sharing takes place in the cloud. The hash data is generated in this fusion process, resulting from the 3 DES algorithm and evaluating authentication and verification. The data fusion is done in the sequence for both authentication and verification to avoid permutation in IoT, and it is represented in the equation below.

$$U_\alpha = \frac{hx'}{cry} * \sum_{Au_0}^{Ve'} Hd' + [(He' * Fo_t) * (Se' + Ang)] *$$

It means,

$$KG_0(Au_0) = \sum_{hx'} x_p + (R_v \leftarrow Se') * y_S$$

The above equation is represented as,

$$y_S(hx') = KG_0 + |Ang * pr(tab)|$$

The analysis for key generation is evaluated for healthcare data sharing and it is expressed as,

$$Ang(KG_0) = Shr(Hd') * y_S + \sum_{cry}^{hx'}(x_p * FP_a) \quad (4)$$

Data Fusion Process

Fusion integrates authentication and verification, and the data is shared as the cipher text that processes the 3-DES algorithm. The data is received by the end-user who opens the data with the private key and obtains the plaintext. This computation step is done for authentication and verification and ensures the sequences of data sharing, the fusion is U_α. Therefore, the fusion generates a combined sequence of linear authentication and validation procedures enclosing the actual health data. Post to this equation, the authentication, and its private key generation is formulated below to share the data securely.

$$Au_0(Ang) = Se' \rightarrow Shr(Hd') * R_v + \sum_{FP_a}^{hx'} pr(tab)$$

Consequently,

$$Shr(EVa_0) = y_S(Se' \leftrightarrow R_v) * KG_0 + (He' + Fo_t)$$

Similarly,

$$Ang(C_y) = y_S(Hd') * Q_u + \frac{U_\alpha}{cry}$$

The above equation is re-written as,

$$cry(Au_0) = Q_u(KG_0 + Va') - B_i(x_p) \quad (5)$$

The analysis for authentication is expressed above where the key sharing is performed with the permuted table which is discussed in the 3-DES algorithm. Here, authentication is given only if the receiver holds the private key, or else the sharing is denied. This is how the authentication for health care is given by encrypting the data while sending and decrypting with a private key and receiving the plaintext. The fusion process for authenticated data generation is illustrated in Fig. 6.

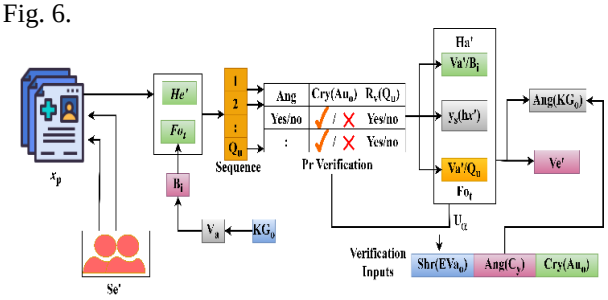


Fig. 6. Fusion Process for Authenticated Data Generation**.**

The U_α process is illustrated (Fig. 6) by defining x_p and (KG_o, Va', B_i) in all the Shr intervals. The first combination is the $He' \in Va'$ and B_i such that the default U_α consolidates $x_p \rightarrow hx'$ and $cry(Au_o)$ is the header. In the footer definition the Ve' matching $Ang(C_y)$, $Ang(KG_o)$ are expected to satisfy $Shr(EVa_o)$ provided the linearity and incremental $(Q_u \leftarrow Se' \rightarrow R_v)$ is required. Depending on the number of Shr and Q_u the pr verification for replication occurs. Therefore the U_α consolidates $Shr(Eva_o)|Ang(C_y)|Cry(Au_o)|Ang(Go))$ provided the Ve' reliable to ensure maximum privacy in data sharing. The fusion process generates a highly authenticated entry of x_p from the Re' for analysis. The header is used for the authentication key. From the obtained authenticated sequences, the sharing is evaluated for the same key and it is formulated below.

$$R_v(Q_u) = (EVa_0 + KG_0) * \prod_{U_\alpha}^{Hd'} Fo_t + (Se' * y_S) \quad (6)$$

The sequences are similar on the receiver side if both the authentication and verification hold the same key on the header and footer. For this computation step, security is ensured by using the same key generation. Here forth, the verification is done for received data sequences and it is formulated below.

$$Ve' = U_\alpha + \prod_{x_p}^{Hd'} x_p \leftarrow hx'(Q_u + Au_0) * (He' + Fo_t)$$

The verification is done for the fused data,

$$U_\alpha(Ve') = Ang(Hd' + Q_u) * \prod_{Shr}^{y_s} KG_0(Q_u + Au_0) - Se'$$

Before verification, the authentication is validated for fused data and it is expressed as follows,

$$Au_0(U_\alpha) = \prod_{Shr} Hd'(y_s) + Va' - R_v(x_p) - hx'$$

The above equation is re-written as,

$$= hx' \rightarrow x_p(Va' + R_v) * Shr(Q_u) \quad (7)$$

The fused data is verified from the above equation to ensure similar data sequences from the header and footer. The authenticated data verifies the receiver side and ensures security. The same sequence data is shared between the sender and the receiver end for this 3-DES algorithm is used. Here, some sort of permuted sequences occur due to dissimilar sequences and authentication mismatch due to the permuted table, which is not rearranged correctly to sort out this federated learning introduced. The authentication time analysis is presented in Fig. 7a-c based on the above.

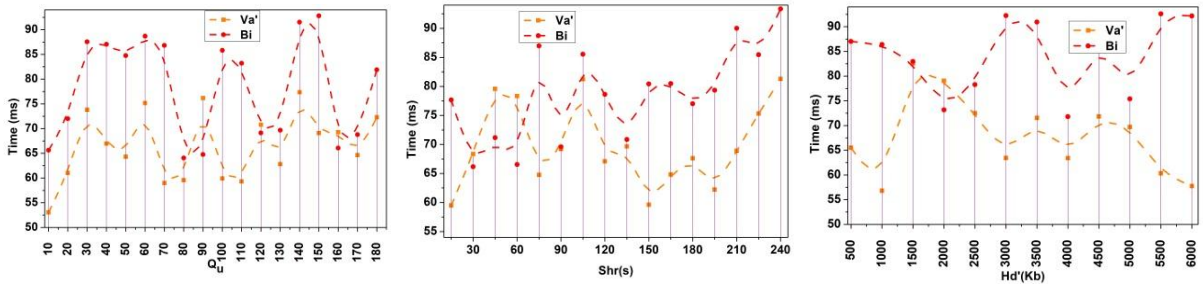


Fig. 7. Authentication Time Analysis : (a) No. of Queries Q_u, (b) Sharing Time $Shr(s)$, (c) Size of the health data Hd'(Kb).

The authentication time is reduced $Au_0(Ang + Hd') < Shr$ depicted in Fig.7 (a), where encryption is done for the requested data $cry(Hd' + Se')$ and shares with the receiver in Fig.7 (b). The receiver acquires the cipher text $hx'(cry + Hd'), for\ all\ cry \in Se'$ and convert them to plaintext $x_p(Hd')$ by the private key shown in Fig.7 (c). The private key is used to open the cipher text $Va'(hx') + R_v, for\ all\ Va' \in x_p$ with the authentication process. Here, time is reduced for authentication $Au_0(Ang * Hd') < Ve'$, where the sender and receiver act faster to share the requested data (Fig. 7a-c).

G. Federated Learning for Model Training

In this DcDFS framework, FL is enhanced by the hierarchical integration of SLM and LLM, which balances edge-level efficiency with centralized intelligence for secure healthcare data processing. At the client side, SLMs are deployed on edge nodes such as hospital devices or IoT gateways to perform lightweight semantic encoding of unstructured patient data into structured vector embeddings before encryption, which are securely fused via the permutation-based data fusion layer. This ensures reduced

data dimensionality before encryption, lowering computation cost. During semantic vector encoding, let $\vec{E}_{SLM} = SLM_i\left(x_i^{(p)}\right)$

Where $\vec{E}_{SLM}$ is the semantic embedding of patient i's data, passed to encryption.

At the central server side, LLMs are deployed to perform higher-order analytics and coordination that operates post-aggregation, analyzing encrypted model updates using high-level reasoning. These models assess latent vector patterns and sequence integrity to detect inversion attacks where $\hat{y}_{LLM} - arg\max_{u\in Y} P(y|\vec{u}_k, \mathcal{H}, \mathcal{F})$ in which $\vec{u}_k$ are the update vectors from node k, the header termed as $\mathcal{H}$, and footer metadata as $\mathcal{F}$ evaluated for integrity violations.

The function of LLM integration in FL is to analyze model updates for adversarial inconsistencies using LLM reasoning capabilities. The data are decentralized and trained if it is permuted or not, it is mainly used to ensure the privacy of health data. The base model is saved on the central server whereas duplicates are distributed to other devices. Maintaining patient anonymity is paramount in healthcare IoT applications and FL permits model training on decentralized data sources without sharing raw data. Data security while training is further enhanced by incorporating 3-DES encryption. Multiple healthcare devices can work together to improve a standard model with the proposed DcDFS while maintaining sensitive health data on-device. Doing away with data centralization aligns with FL's objective of communal learning. Local models trained on dispersed data are periodically updated and forwarded to a central server for aggregation as part of the process. This safeguards the integrity of individual datasets while ensuring a secure synchronization of knowledge a crucial component of FL. The technique improves data integrity checks using Federated Learning for ongoing verification and embedding authentication and verification keys into the data. One major issue with distributed learning environments is the possibility of unauthorized changes; this solves that problem. Here, four steps are used for the federated learning processing.

Step 1: The current model is downloaded based on the specific data the user requests and verifies the authentication.

$$UR_m = y_S + (Hd' * Au_0) + \sum_{hx'} Ang(Ve' * KG_0) \quad (8a)$$

The current model is represented as UR_m, where the authentication is checked for the sender and the receiver. The key generation is considered to perform the encryption and opens the plaintext with the private key. The current model is downloaded based on the receiver's request.

Step 2: The new model which is extracted from Step 1 is improved to ensure security.

$$WN_d = UR_m * (KG_0 + Fo_t) * cry \prod_{EVa_0} Ang * (Hd') - C_y \quad (8b)$$

The new model generated is represented as WN_d, where the analysis takes place by encrypting the data and providing privacy for data sharing in IoT. From step 1, the current model is generated, and the key generation process for cipher text is evaluated. So, it is done by generating the new model for both header and footer, which improves privacy.

Step 3: The periodic update of the model is evaluated and communicated with the cloud environment. The communication is encrypted to ensure the security of the health care data and that privacy is attained.

$$\begin{aligned} DP_t = \frac{Ve' * He'}{\sum_{Hd'}(Ang + Au_0)} + \sum_{pr(tab)}^{Shr} Q_u \\ * \{[(Fo_t + He') + (Au_0 * Ve')] - Va'\} + B_i \\ * \sum_{y_S}^{R_v} hx' + [(Eva_0 * cry) + Q_u] * UR_m(WN_d) \\ + \sum_{KG_0} Q_u * (UR_m + U_\alpha) * Hd'(y_S) * Eva_0 \\ + WN_d \end{aligned} \quad (8c)$$

The periodic update is provided for the current model where the mismatch of permuted is addressed, the update is DP_t. Here, it defines the security-based communication for the sender and the receiver by computing the authentication and verification on the identical sequences. The encrypted data are shared with the receiver and the private key is used to open the data with privacy. This update is built to ensure data fusion, which is evaluated by sharing the data with security.

Step 4: It is the final step that represents the update obtained from the multiple users and builds the current mode as the output.

$$\begin{aligned} Ang(WN_d) = \sum_{Ve'}^{Au_0} (He' * Fo_t) + (U_\alpha + DP_t) \\ * \left\{[(KG_0 + cry) + (EVa_0 * Hd')] * \frac{cry}{y_S}\right\} \\ * \sum_{U_\alpha} Se' + (x_p * hx') - C_y \end{aligned} \quad (8d)$$

The analysis for the new model is evaluated to find the update that is extracted from multiple users. The new model is used to evaluate the header and the footer. Based on this update the training is given if there are any adversaries are detected while sharing the data. Here forth, the data sequences are evaluated in the following equation.

$$\begin{aligned} Eva_0(Hd', Q_u) = (Au_0 + Ve') * \prod_{KG_0}^{hx'} UR_m + FP_a \\ * \left\{[(U_\alpha + cry) + (y_S * hx')] * \prod_{x_p} C_y\right\} \\ + |(He' + Fo_t)| * \prod_{y_S} p_s + (U_\alpha * DP_t) * cry(R_v) \\ + \langle \sigma_a + Au_0 \rangle * KG_0(Va' + B_i) * pr(tab) \\ + (He' * Fo_t) \end{aligned} \quad (9)$$

The sequences are evaluated for the health data, which includes authentication and verification. The correlation is σ_a for sequences of authentication and verification and improves the security level. A similar sequence mapped between the header and footer is responsible for discarding unauthorized data handling. The sequence verification decisions using federated learning are presented in Fig. 8.

The $R_v(Q_u)$ verification process is detailed in the above Fig. 8 illustration. The initial assumption is $U_\alpha \in [He'|hx'|Fo_t]$

for which FP_a is updated if KG_o is true in $Q_u > 0$ instance.

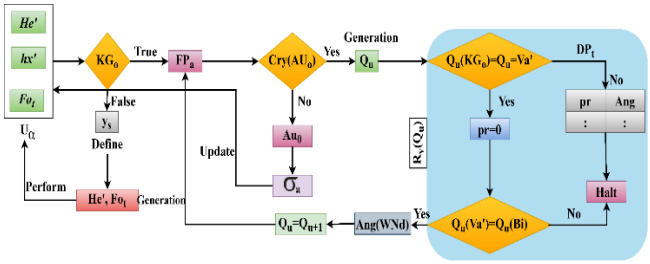
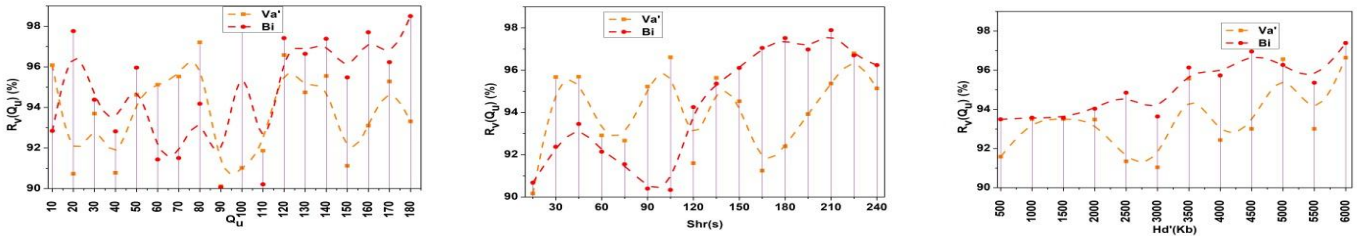


Fig. 8. Sequence Verification Decisions

Based on the incremental demands, the $Cry(Au_o)$ and verification (same) of $Q_u \forall Va', Bi$ and Shr are performed. If the verification is successful, then $Ang(Wn_d)$ is the final call for a new sequence and hx' authentication is revisited. If this case fails the $pr(tab)$ is updated with $FP_a, Ang,$ and $Ang(KG_o)$ sequences. Depending on the (He', Fo_t) failures the sequence halt is recommended. The initial U_α is the current model (step 1) for which WN_d (step 2) is the final solution. The update from σ_a enables (step 3) the new y_s demands; the final step is the validation required from the halted conditions defined above. Based on the above, the sequence verification post Au_o process is analyzed as in Fig.9a-c.

In DcDFS scheme, federated learning works through the following structured process



(a) (b) (c)

Fig. 9. Sequence Verification: (a) No. of Queries Q_u, (b) Sharing Time $Shr(s)$, (c) Size of the health data Hd'(Kb).

i)Secure key generation

Each device generates secure keys for encryption and authentication using KG_0 and these keys prevent adversarial attacks during model updates. Each participating node in the FL framework generates a unique cryptographic key to ensure secure communication. The use of hx', Hd' encrypted health data values participating in FL training. The private -public key mechanism guarantees that only authorized details used in federated model training.

ii) Model Update and Secure Aggregation

The federated model is updated at each node by aggregating secure local updates while maintaining data privacy. Each device updates the model based on the encrypted local training data using UR_m, only authenticated updates are aggregated to ensure security.

iii) Receiver's Secure Model Update

Each device verifies and retrieves only secure updates using $R_v(Q_u)$ and ensures no adversarial manipulation in the federated model. The role of FL in receivers secure model update ensures nodes receive only verified model udpates to prevent poisoning attacks.

iv) Periodic Model Update:

The periodic FL updates the server periodically synchronizes models while maintaining privacy using DP_t. Multiple layers of security authentication, encryption, and secure aggregation enhance privacy. FL requires periodic updates to maintain an efficiency and privacy-preserving training cycle. This ensures periodic model synchronization while preventing unauthorized modifications. The sequence verification is higher which is processed from the authentication of data $Au_0(Hd' + Q_0) > Ve'(R_v)$ and provides the output to the receiver shown in Fig.9a. The verification is done once the data is received $R_v(Hd' \leftarrow Se') + \sum_{C_y}^{KG_0} Va' < y_S(Shr)$ shown in Fig.9b

Data Sharing

The data sharing $Hd'(Shr) \rightarrow R_v$, to the receiver and ensures the security $y_S(Ang + Q_u)$ from the data fusion shown in Fig.9c. The data fusion $(U_\alpha + Hd') > C_y$ is evaluated for $(Hd' + Au_0) * Shr$ and sharing is done between the sender and the receiver (Fig. 9a-c). The sequence verification process is detailed in Procedure 2.

Procedure 2 Sequence process for Recipient Verification

Input: The health data verification is done from the data fusion authentication $Hd'(Au_0 + Ve')$ where sequences are the same during the data sharing $Shr(R_v + Q_u)$// where data is provided to the requestor $R_v(x_p) \in Shr$.

Output: The verification process is sequenced on the receiver side and provides the authentication.

1. Initialize the data requested where it indicates the header and footer $(He' + Fo_t), for\ all\ Au_0 \in C_y$
2. //privacy is provided for the data where the sequences are verified.
3. For $C_y(Q_u) + Hd'$ do
4. $(Au_0 + Ve') > y_S, for\ all\ y_S \in Hd'$// the same sequences are given for the authentication and verification.
5. // Validate the input data sequences
6. for $Va'(hx' + x_p) - Ang$ do// Private key generation
7. The cipher text is converted to plaintext using a private key where authentication is evaluated
8. for $Au_0(KG_0 + Hd') > Shr$ in cry, Q_u do
9. The sharing is done to the authenticated user to improve the verification process and it is derived in equation (3).
10. end for

11. The key generation is processed to open the cipher text
12. end for
13. The header and footer are responsible for ensuring the same sequences
14. The verification is done after the authentication is given to the user as expressed in equation (5)
15. The receiver data is verified $R_v(Ve') - C_y$
16. end for

The sequence verification is higher which is processed from the authentication of data $Au_0(Hd' + Q_0) > Ve'(R_v)$ and provides the output to the receiver. The verification is done once the data is received $R_v(Hd' \leftarrow Se') + \sum_{C_y}^{KG_0} Va' < y_S(Shr)$. The data sharing $Hd'(Shr) \rightarrow R_v$, to the receiver and ensures the security $y_S(Ang + Q_u)$ from the data fusion. The data fusion $(U_\alpha + Hd') > C_y$ is evaluated for $(Hd' + Au_0) * Shr$ and sharing is done between the sender and the receiver (Fig. 9). From this maximum privacy is produced between the actual and security-related data and they are derived below.

$$C_y = (y_S + Hd') * \sum_{\sigma_a}^{Au_0} (Shr + Hd') * \left[(KG_0 + Fo_t) + (x_p - hx')\right] + Shr(R_v - y_S) + Ve'(Q_u) \quad (10)$$

The maximum privacy is equated in the above equation by the use of data fusion. The sharing is accomplished to the receiver and a private key is generated. The key generation is important to ensure the security between the sender and the receiver where it uses the permuted table for update and rearranging of data. The federated learning is used to address the permuted and mismatch of health data and generates the output with security. The correlation factor described in equation (9) is discussed for its significance in Procedure 3.

Procedure 3 Correlation Factor

Input: The evaluation is for health data with sequences acquired with the data fusion $U_\alpha + Hd'(p_s + pr(tab))$, where the permutation table is updated with authentication and verification $(Au_0 + Ve') * y_S$// where each input data is encryption is given for security $cry(Hd' + p_s)$

Output: The sequence between authentication and verification

1. Initialize the sequence for the header and footer to evaluate the data fusion $U_\alpha + (Hd' * Q_u)$.
2. // Evaluate the sequences for the correlation process
3. for $\sigma_a(Q_u) * Au_0$ in (y_S, Hd') do
4. $Q_u(He' + Fo_t) < C_y$ //The Privacy is processed for the correlated data
5. //analysis is done for the encryption process where the correlation for fused data is evaluated.
6. For each $Hd'(\sigma_a + Shr)$ do// Sharing the correlated data
7. Authentication and verification correlation are detected with the same key sequences.
8. For $Shr = cry$ in $Hd'(y_S)$ do
9. The sequence data is considered and checked for the correlation which is derived in equation (9)
10. end for
11. The encryption process is evaluated for the correlation data
12. end for
13. The data fusion is maximized for the data-sharing process.
14. The sharing indicates the cipher and plaintext conversion to ensure privacy and security which is represented in equation (10)
15. The Correlation sequences are analyzed for privacy with the same key such as header and footer.
16. end for

The suggested method validates model updates using header-footer sequence mapping to prevent data injection, poisoning, and adversarial modifications. Based on a unique query Q_u and cryptographic key KG_0, each user device creates an authentication header Hd'. System checks header integrity with verify_header (Hd', KG_0) before sending model updates. Creating and validating a verification footer (Fo') ensures data compliance with the desired structure.

Procedure 4 Sequence based validation

Input: Hd', Q_u

Output: True/False.

Function validate_sequence (Hd', Q_u)

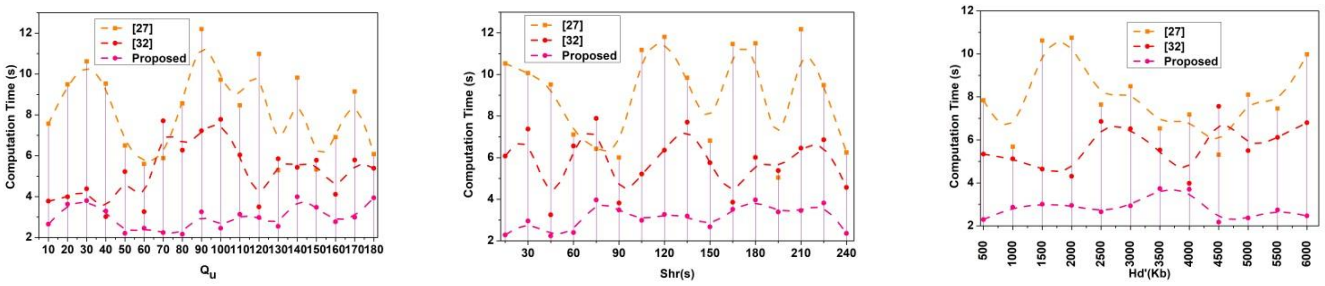
17. $KG_0 = generate\_key(Q_u)$
18. ***if*** verify_header $(Hd', KG_0) == False$
19. $return\ false$
20. $Fo_t = genrerate\ verification_{footer(Hd', KG_0)}$
21. ***if*** validate _footer $(Hd', KG_0) == False$
22. $return\ false$
23. Check data consistency
24. ***if*** $D_{match}! = true$ ***then***
25. $return\ false$
26. $return\ true$

The function compare_data detects mismatches and potential attacks using procedure 4. Then, updates are made if any validation step fails, reducing poisoned update risks. Sequence-based validation ensures that only verified, unaltered modifications are added to the global model, protecting data privacy. A step-by-step verification method in the pseudocode helps explain this mechanism, ensuring robust security in federated learning settings.

V. RESULTS AND DISCUSSION

Computational efficiency, privacy leakage, computational time, and scalability analysis are the critical performance metrics that should be defined in the experimental design. These metrics should be measured using authentication requests, data encryption, and verification processes. The computation time for authentication and verification key exchanges is measured in seconds. The percentage of unauthorized data exposure mitigated through encryption is used to evaluate privacy leakage. Processing rates and improvements in model performance using federated learning are used to assess computational efficiency. As input variables, the experiment compares outcomes with current approaches ([27] and [32]) using Q_u (10 to 180), Shr time (15s to 240s), and Hd' (500KB to 6000KB). Our system is equipped with a high-performance computing environment that guarantees trustworthy evaluations. It has an Intel Core i7 processor, 16GB of RAM, and an NVIDIA RTX 3060 graphics card. The findings, shown in Figures 10-12, show that the system is more secure and efficient than previous methods.

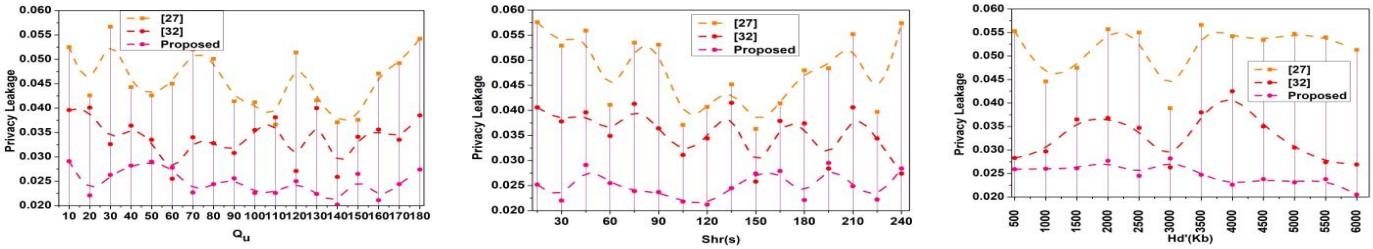
The computation time for the authentication and verification keys decreases. Here, header and footer [45] are used to share the same key sequences $He' + Fo_t(Q_u + K_g)$ so the computation time is reduced is shown in Fig. 10 (a). The sender request for the health data $Se'(Hd' * \sum_{cry} hx' \rightarrow x_p) * R_v$, where encryption is done for the data, and a private key is used to convert the cipher text to plaintext shared is shown in Fig10 (a), (b), and (c). Computation time is an evaluation metric calculated by analyzing the total time taken for encryption, decryption, authentication, and data fusion operations.



(a) (b) (c)

Fig. 10. Computation Time Comparisons: (a)No. of Queries Q_u, (b) Sharing Time $Shr(s)$, (c) Size of the health data Hd'(Kb) .

From the use of $CT = O(Q_u + K_g) + O(hd' * \sum_{cry} hx' \to x_p) * R_v)$ means more queries from 10 to 180, and larger data sizes lead to higher computation time, with the size of health data including 500kb to 6000kb is shown in Fig.10 (c). The cryptographic operations $\sum_{cry} hx' \to x_p) * R_v$ needed for encryption and verification factor. The privacy is given for the health data by ensuring the security parameter $C_y + (y_S * Hd') + \prod_{R_v}(Va' * Fo_t)$where it includes the data fusion. The encryption ensures the authentication between the sender and the requestor, which is evaluated from the data fusion and shows lesser computation time (Fig. 10a-c). The privacy leakage comparisons are presented in Fig. 11a-c for the three variants disclosed.



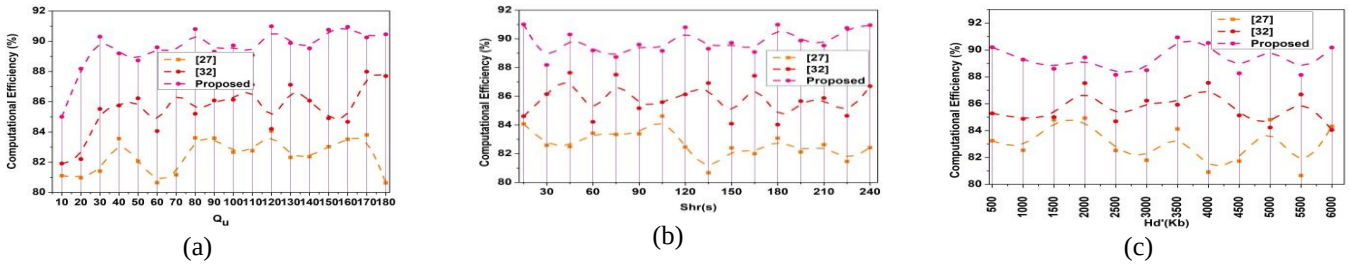
(a) (b) (c)

Fig. 11. Privacy Leakage Comparisons: (a) No. of Queries Q_u, (b) Sharing Time $Shr(s)$, (c) Size of the health data Hd'(Kb) .

The privacy leakage is lesser by providing security for the health data in IoT, $cry(hx' * x_p) * y_S < Au_0$ and authentication is computed from the data fusion. Here, privacy is given for the health data which includes three keys such as authentication, verification acts as the same key, and the private and public key. The authentication is generated from the hash data $Au_0(Q_u + Hd') * \sum_{B_i}^{Va'} Ve' + C_y\,, for\ all\ Eva_0 \in KG_0$ and evaluates the three keys to avoid privacy leakage. It is due to adversaries on IoT and it is detected with the authentication and verification key $(Au_0 + Ve') * \prod_{y_S}(Ang + KG_0)$ and analysis of the key generation [46]. Thus, it shows lesser privacy leakage and provides security between the sender and the receiver (Fig. 11a-c). In Fig. 12a-c the computation efficiency for the three variants is comparatively analyzed. Privacy leakage p_L measures how much information is exposed to adversaries during data transmission where a lower privacy leakage ensures better security

$$p_L = O(\,cry(hx' * x_p) * y_S) + O(Au_0(Q_u + Hd') * \sum_{B_i}^{Va'} Ve' + C_y) \quad (11)$$

In higher Au_0 in equation (11) represents lower privacy leakage, higher $cry(hx' * x_p)$ ensures better data protection, allocates stronger security between sender and receiver.



(a) (b) (c)

Fig. 12. Computation Efficiency Comparisons: (a) No. of Queries Q_u, (b) Sharing Time $Shr(s)$,(c) Size of the health data Hd'(Kb).

The computation efficiency increases for higher privacy and security. The efficiency is improved by introducing federated learning which includes four steps, and evaluates new and current models. Here, the final permutation is used to analyze

the current and the new model $Ang(WN_d + UR_m) * \prod_{KG_0} pr(tab), for\ all\ pr(tab) \in DP_t$. For every iteration step, fusion input is given $U_\alpha + (hx' - x_p) * \frac{Hd'}{WN_d} * (Va' + B_i) < Au_0$ where the private key is used to convert the cipher text into plaintext. Computational Efficiency CE measures the system ability to process authentication and encryption operations efficiently while ensuring security. A higher CE means better performance with reduced processing overhead.

$$CE = O\left(U_\alpha + (hx' - x_p) * \frac{Hd'}{WN_d} * (Va' + B_i)\right) + O\left(Ang(WN_d + UR_m) * \prod_{KG_0} pr(tab), for\ all\ pr(tab) \in DP_t\right) \quad (12)$$

The key observations from these computations tells that lower encryption overhead $hx' - x_p$ in equation (12) represents faster processing. An efficient model updates UR_m represents enhanced computation speed. The privacy is evaluated for the current and the new model $C_y(WN_d + UR_m), for\ all\ C_y \forall (Au_0 + Ve')$ and evaluates the higher computation efficiency [47] with authentication and verification (Fig. 12a-c).

Table 2: Scalability Analysis

Model	Latency (L) (ms)	Storage Overhead (S) (MB/device)	Scalability Index Factor (SI)	Communication cost (kb)	Model Type	Modle size (parameters)
FL + Blockchain [26]	250-400	500-800	0.000003 - 0.000005 (Low)	High	Distributed Neural Network	20,000-30,000
Homomorphic Encryption [30]	100-200	50-150	0.000033 - 0.0002 (Moderate)	Moderate	Convolutional Neural Network	15,000-25,000
Privacy-Preserving FL with TEE [39]	80-150	100-300	0.000011 - 0.000125 (High)	Low	Recurrent Neural Network	10,000-20,000
Proposed DcDFS	50-120	30-100	0.000083 - 0.000666 (Highest)	Low	Hybrid (Combination	5,000-10,000

The scalability Index (SI) is a metrics analyzed in terms of latency L and storage overhead S incorporates network sizes/devices calculated using equation (13).

$$SI = \begin{cases} \frac{N}{\alpha L(N) + \beta S(N) + \gamma O(N)} \\ L(N) = \lambda_1 N^p + \lambda_2 \left(\frac{N}{\log N}\right) + \lambda_3 e^{\beta N} \\ S(N) = \mu_1 (K + M)N + \mu_2 O(N^q) + \mu_3 e^{\alpha N} \end{cases} \quad (13)$$

$L(N)$ as the latency function and $S(N)$ as the storage overhead depends on encryption and model key storage as given in Table 2, λ_1 as computational cost per device, $\lambda_3 e^{\beta N}$ captures exponential cryptographic overhead, and $O(N)$ as the overall complexity including cryptographic operations, network communication, and scalability weight parameters as α, β, γ based on learning constraints. The term M as communication storage overhead per device.

As shown in Table 2 analyse the scalability. In federated contexts, recurrent and hybrid models like DcDFS improve communication efficiency and synchronization. Larger models (20,000+ parameters) require more computing and memory, delaying updates on low-resource devices. DcDFS's smaller model size and lower communication costs enable efficient updates across diverse devices, reducing update delays and maintaining model convergence in federated learning environments. DcDFS's reduced model size (5,000-10,000 parameters) speeds updates and synchronization across devices with different computational capabilities. The DcDFS employ federated learning to improve privacy while keeping model predictions highly accurate. The approach keeps health forecasts accurate while protecting raw data from intrusions by storing sensitive data locally on devices and only sharing model updates. Despite the computational expense introduced by using triple-DES for data encryption, the research shows that the proposed technique effectively reduces calculation time.

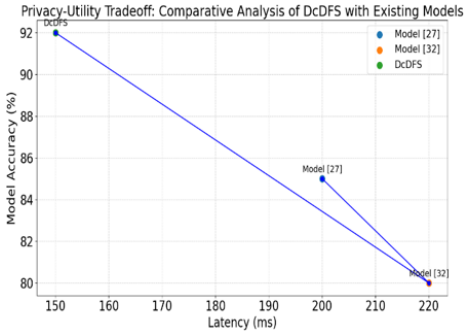


Fig.13 Privacy-Utility Tradeoff Analysis

Overall, this reduction positively affects the latency, which helps achieve real-time data access while ensuring privacy. Compared to the current models, DcDFS improves accuracy by 11.62% and reduces latency by 28.4% on average as illustrated in Fig.13.

Table 3. Comparison Privacy-Preserving FL Techniques

Method	**Accuracy (%)**	**Latency (ms)**	**Privacy Leakage**
Secure Aggregation (SecAgg) [39]	82	180	Low
Homomorphic Encryption (HE) [37,38]	88	400	Very Low
DP-FTRL [48]	85	250	Low
Proposed DcDFS	92	150	Moderate

Table 3 compares DcDFS stacks against SecAgg, HE for FL, and DP-FTRL regarding accuracy, latency, and privacy leakage. As an example of its computational efficiency, DcDFS obtains the highest accuracy (92%). It also maintains

the lowest latency (150 ms). HE provides the highest level of anonymity, but its 400 ms latency makes it impractical for use in real-time scenarios. SecAgg and DP-FTRL provide strong privacy assurances, but accuracy and computing efficiency are sacrificed to some extent. When it comes to federated learning scenarios, DcDFS is a viable method since it balances privacy, utility, and performance.

Table 4: Comparative Analysis of Encryption Methods

Encryption Method	Security Strength	Encryption Time (ms)	Decryption Time (ms)	Computational Overhead	Scalability in IoT
Triple DES (3DES)	Medium (112-bit effective)	High (~5-10 ms per 1KB)	High (~5-10 ms per 1KB)	High (3x encryption passes)	Low (Not suitable for real-time IoT)
AES-GCM [49]	High (128/192/256-bit)	Low (~1-2 ms per 1KB)	Low (~1-2 ms per 1KB)	Low (Optimized with hardware acceleration)	High (Efficient for real-time IoT)
ChaCha20 [50]	High (256-bit)	Low (~1-3 ms per 1KB)	Low (~1-3 ms per 1KB)	Low (Stream cipher with minimal processing)	High (Suitable for low-power IoT devices)

Table 4 compares various encryption methods regarding their security, computational overhead, scalability in IoT environments, and the time it takes to encrypt and decode data. Due to its substantial processing overhead, Triple DES (3DES) is not recommended for real-time Internet of Things (IoT) applications. AES-GCM uses hardware acceleration to strike a compromise between efficiency and security. ChaCha20 is perfect for low-power Internet of Things devices since it is a lightweight stream cipher with little processing overhead.

Security Model for FL with Triple DES Encryption:

Assumptions of Data & Model:

Clients train a global model M using private data sources D_{train} and clients encrypt gradient updates using 3-DES before sharing is given as

$$G_i = \begin{cases} E_{3-DES}(\nabla L(M_i)) \\ \nabla L(M) = D_{3-DES}(G_i) \end{cases} \quad (14)$$

Equation (14) shows an adversary has access to trained model M and encrypted gradients but not the decryption key D_{3-DES}. The security guarantees are tabulated below:

Table 5: Security Guarantees Against FL Attacks

Attack Type	Defense Mechanism	Security Guarantee
Data Reconstruction	Triple DES encryption	Encrypted updates prevent the recovery of raw training data
Gradient Leakage	Encrypted gradient transmission	Attackers cannot decrypt gradients without the key
Model Inversion	Noise introduced by encryption	Prevents adversaries from reconstructing training data
Membership Inference	Aggregation of encrypted updates	Hides individual contributions, reducing attack success

Multiple users contribute updates in FL, making it challenging to attribute updates to a specific dataset entry. Since gradients remain encrypted, adversaries cannot reconstruct data or infer membership. The probability of a successful attack is given as a success rate as $S_R(A) \leq \epsilon$, where ϵ is negligible, ensuring privacy in federated learning. Table 5 summarizes how Triple DES encryption protects federated learning data against adversarial assaults. It details the dangers, countermeasures, and resulting safety assurances. Encryption safeguards sensitive information by avoiding model inversion, gradient leakage, and data reconstruction hazards.

Interpretation of SLM and LLM in Security Metrics

The privacy leakage rate (PLR) metric measures the percentage of sensitivity by SLM that is unintentionally exposed or inferred by adversaries during data transmission in the scope of privacy-preserving FL and secure health data exchange; a lower PLR indicates better confidentiality assurance, especially under high query loads, prolonged sharing sessions, and larger payloads. Similarly, the SLMs on edge devices are typically more vulnerable to adversarial attacks like model inversion, resulting in a higher AIF. As edge devices are less powerful and more easily compromised, SLMs often struggle to defend against these malicious influences, making them more prone to adversarial impact.

Table 6: Comparative Analysis of PLR and AIF Under Varying System Parameters

Input Variables	Metric(%)	[52] Pujari & Pakina	[53] Chen et al.	[54] Mahadik et al.	DcDFS (Proposed)
Q_u = 10	PLR	14.8	13.1	11.2	3.9
	AIF	12.6	11.3	9.9	3.2
Q_u = 90	PLR	21.5	19.7	17.6	6.8
	AIF	17.7	15.9	14.3	5.5
Q_u = 180	PLR	29.4	27.2	24.3	10.1
	AIF	25.6	23.5	20.8	7.8
Shr Time = 15s	PLR	11.2	10.3	9.2	3.1
	AIF	10.1	9.2	8.1	2.7
Shr Time = 240s	PLR	32.1	29.3	27	11.4
	AIF	27.3	24.9	22.6	8.3
Hd' = 500KB	PLR	12.6	11.3	10.4	3.5
	AIF	11.3	10.2	9.1	3
Hd' = 6000KB	PLR	35.7	33.2	30.4	12.2
	AIF	30.4	28.1	25.7	9.6

The privacy leakage rate (PLR) values are relatively higher in comparison to the proposed DcDFS framework, as provided in Table 6, indicating that while SLMs reduce data transmission by performing local processing, they still expose more privacy risks due to limitations in computational capabilities, security measures, and model robustness on edge devices. Adversarial Influential Factor (AIF) is higher in SLMs, showing vulnerability to adversarial attacks, as SLMs often lack the sophisticated countermeasures found in larger systems or federated models. By integrating 3DES encryption, data fusion, and federated learning, the DcDFS framework maintains low PLR across all test scenarios, ensuring the patient data remains highly protected even during model training over federated networks. AIFs are also significantly lower than SLMs and LLMs, indicating that the DcDFS system is less vulnerable to adversarial attacks due to its robust encryption, secure aggregation, and advanced adversarial detection techniques. LLMs exhibit higher AIF

due to their vulnerabilities to adversarial influence in decentralized learning environments, especially without sophisticated security mechanisms.

VI. Conclusion

This article presents dependency-correlated data fusion using a federated learning paradigm. We have proposed a scheme to generate security and cipher data for fusion and consolidate with the header and footer security features. Findings suggest that the authentication uses key generation through private and public keys administered, and then the footer section verifies the shared data, ensuring no privacy leakage. The key generation, authentication, and verification use the conventional DES algorithm with sequence evaluation modification evaluated using FL in both sender and receiver to ensure the above mapping is similar. Depending on the number of sharing intervals, the fusion process relies on combined sequences mapped between authentication and verification such that the out-of-the-box sequences are discarded from sharing. The proposed DcDFS offset the best privacy and security performance, with low PLR and low AIF, making it the most robust and privacy-preserving solution for privacy-sensitive applications like healthcare with the integration of SLM and LLM in FL. Therefore, the linear dependency between the data and authentication parameters is reliable in retaining high computation efficiency with better privacy solutions for healthcare IoT data sharing by achieving reduced computation time and privacy leakage, and higher efficiency observed under the maximum sharing interval.

ACKNOWLEDGEMENT

This work was supported bythe Centre of Excellence for Research, Value Innovation, and Entrepreneurship(CERVIE) at UCSI University for funding this research project through theResearch Grant with the project code: T2S-2025/008. The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through Large Research Project under grant number RGP.2/275/46.

References

[1] Wang, N., Zhang, S., Zhang, Z., Fu, J., Liu, J., & Wang, R. (2022). Block-based privacy-preserving healthcare data ranked retrieval in encrypted cloud file systems. *IEEE Journal of Biomedical and Health Informatics*, *27*(2), 732-743.

[2] Deebak, B. D., & Hwang, S. O. (2023). Healthcare applications using blockchain with a cloud-assisted decentralized privacy-preserving framework. *IEEE Transactions on Mobile Computing*.

[3] Wang, S., Ge, C., Zhou, L., Wang, H., Liu, Z., & Wang, J. (2022). Privacy-Preserving Classification in Multiple Clouds eHealthcare. *IEEE Transactions on Services Computing*, *16*(1), 493-503.

[4] Popoola, O., Rodrigues, M., Marchang, J., Shenfield, A., Ikpehai, A., & Popoola, J. (2024). A critical literature review of security and privacy in smart home healthcare schemes adopting IoT & blockchain: problems, challenges and solutions—blockchain*: Research and Applications*, *5*(2), 100178.

[5] Shiri, I., Razeghi, B., Ferdowsi, S., Salimi, Y., Gündüz, D., Teodoro, D., ... & Zaidi, H. (2024). PRIMIS: Privacy-preserving medical image sharing via deep sparsifying transform learning with obfuscation. *Journal of biomedical informatics*, *150*, 104583.

[6] Begum, T. U. S. (2024). Federated and multi-modal learning algorithms for healthcare and cross-domain analytics. *PatternIQ Mining - Sahara Digital Publications*, *1*(4), 38–51..

[7] Masood, I., Daud, A., Wang, Y., Banjar, A., & Alharbey, R. (2024). A blockchain-based system for patient data privacy and security. *Multimedia Tools and Applications*, *83*(21), 60443-60467.

[8] Shi, H., Zhou, Z., Qin, J., Sun, H., & Ren, Y. (2024). A separable privacy-preserving technique based on reversible medical data hiding in plaintext encrypted images using neural network. *Multimedia Tools and Applications*, 1-26.

[9] Agbley, B. L. Y., Li, J. P., Haq, A. U., Bankas, E. K., Mawuli, C. B., Ahmad, S., ... & Khan, A. R. (2023). Federated fusion of magnified histopathological images for breast tumor classification in the internet of medical things. *IEEE Journal of Biomedical and Health Informatics*.

[10] Albahri, A. S., Duhaim, A. M., Fadhel, M. A., Alnoor, A., Baqer, N. S., Alzubaidi, L., ... & Deveci, M. (2023). A systematic review of trustworthy and explainable artificial intelligence in healthcare: Assessment of quality, bias risk, and data fusion. *Information Fusion*, *96*, 156-191.

[11] Chen, X., Xie, H., Li, Z., Cheng, G., Leng, M., & Wang, F. L. (2023). Information fusion and artificial intelligence for smart healthcare: a bibliometric study. *Information Processing & Management*, *60*(1), 103113.

[12] Ahmed, S. F., Alam, M. S. B., Afrin, S., Rafa, S. J., Rafa, N., & Gandomi, A. H. (2024). Insights into Internet of Medical Things (IoMT): Data fusion, security issues and potential solutions. *Information Fusion*, *102*, 102060.

[13] Chen, X., Xie, H., Tao, X., Wang, F. L., Leng, M., & Lei, B. (2024). Artificial intelligence and multimodal data fusion for smart healthcare: topic modeling and bibliometrics. *Artificial Intelligence Review*, *57*(4), 91.

[14] Bezanjani, B. R., Ghafouri, S. H., & Gholamrezaei, R. (2024). Fusion of machine learning and blockchain-based privacy-preserving approach for healthcare data in the Internet of Things. *The Journal of Supercomputing*, *80*(17), 24975-25003.

[15] Zhang, X., Jiang, M., Wu, W., & de Albuquerque, V. H. C. (2023). Hybrid feature fusion for classification optimization of short ECG segment in IoT based intelligent healthcare system. *Neural Computing and Applications*, 1-15.

[16] He, X., Zhou, W., Luo, Z., Ping, Z., & Wang, M. (2024). Data privacy protection health status assessment for rotating machinery with dual-task feature fusion framework. *Neurocomputing*, *582*, 127464.

[17] Stephanie, V., Khalil, I., Atiquzzaman, M., & Yi, X. (2022). Trustworthy privacy-preserving hierarchical ensemble and federated learning in healthcare 4.0 with blockchain. *IEEE Transactions on Industrial Informatics*, *19*(7), 7936-7945.

[18] Abou El Houda, Z., Hafid, A. S., Khoukhi, L., & Brik, B. (2022). When collaborative federated learning meets blockchain to preserve privacy in healthcare. *IEEE Transactions on Network Science and Engineering*, *10*(5), 2455-2465.

[19] G. Xu et al., "A Model Value Transfer Incentive Mechanism for Federated Learning With Smart Contracts in AIoT," IEEE Internet of Things Journal, vol. 12, no. 3. Institute of Electrical and Electronics Engineers (IEEE), pp. 2530–2544, Feb. 01, 2025. doi: 10.1109/jiot.2024.3468443..

[20] Muazu, T., Mao, Y., Muhammad, A. U., Ibrahim, M., Kumshe, U. M. M., & Samuel, O. (2024). A federated learning system with data fusion for healthcare using multi-party computation and additive secret sharing. *Computer Communications*, *216*, 168-182.

[21] J. Hu et al., "WiShield: Privacy Against Wi-Fi Human Tracking," IEEE Journal on Selected Areas in Communications, vol. 42, no. 10. Institute of Electrical and Electronics Engineers (IEEE), pp. 2970–2984, Oct. 2024. doi: 10.1109/jsac.2024.3414597.

[22] Khan, I. A., Razzak, I., Pi, D., Khan, N., Hussain, Y., Li, B., & Kousar, T. (2024). Fed-inforce-fusion: A federated reinforcement-based fusion model for security and privacy protection of IoMT networks against cyber-attacks. *Information Fusion*, *101*, 102002.

[23] Han, F., Yang, P., Du, H., & Li, X. (2024). Accuth+: Accelerometer-

Based Anti-Spoofing Voice Authentication on Wrist-Worn Wearables. IEEE Transactions on Mobile Computing, 23(5), 5571-5588. doi: 10.1109/TMC.2023.3314837.

[24] Khadidos, A. O., Khadidos, A. O., Selvarajan, S., & Mirza, O. M. (2023). TasLA: An innovative Tasmanian and Lichtenberg optimized attention deep convolution based data fusion model for IoMT smart healthcare. Alexandria Engineering Journal, 79, 337-353.

[25] Fan, Q., Xie, Y., Zhang, C., Liu, X., & Zhu, L. (2024). An Authentic and Privacy-Preserving Scheme Towards E-Health Data Transmission Service. IEEE Transactions on Services Computing.

[26] Li, C., Dong, M., Xin, X., Li, J., Chen, X. B., & Ota, K. (2023). Efficient privacy-preserving in IoMT with blockchain and lightweight secret sharing. IEEE Internet of Things Journal.

[27] Alsuqaih, H. N., Hamdan, W., Elmessiry, H., & Abulkasim, H. (2023). An efficient privacy-preserving control mechanism based on blockchain for E-health applications. Alexandria Engineering Journal, 73, 159-172.

[28] C. Li, A. He, G. Liu, Y. Wen, A. T. Chronopoulos, and A. Giannakos, "RFL-APIA: A Comprehensive Framework for Mitigating Poisoning Attacks and Promoting Model Aggregation in IIoT Federated Learning," IEEE Transactions on Industrial Informatics, vol. 20, no. 11. Institute of Electrical and Electronics Engineers (IEEE), pp. 12935–12944, Nov. 2024. doi: 10.1109/tii.2024.3431020.

[29] Makhdoom, I., Abolhasan, M., Lipman, J., Piccardi, M., & Franklin, D. (2024). PrivySeC: A secure and privacy-compliant distributed framework for personal data sharing in IoT ecosystems. Blockchain: Research and Applications, 5(4), 100220.

[30] Meng, L., & Li, D. (2023). Novel Edge Computing-Based Privacy-Preserving Approach for Smart Healthcare Systems in the Internet of Medical Things. Journal of Grid Computing, 21(4), 66.

[31] Akhtar, M. M., Shatat, R. S. A., Shatat, A. S. A., Hameed, S. A., & Ibrahim Alnajdawi, S. (2023). IoMT-based smart healthcare monitoring system using adaptive wavelet entropy deep feature fusion and improved RNN. Multimedia Tools and Applications, 82(11), 17353-17390.

[32] Abaoud, M., Almuqrin, M. A., & Khan, M. F. (2023). Advancing federated learning through novel mechanism for privacy preservation in healthcare applications. IEEE Access, 11, 83562-83579.

[33] Liu, J., Chang, Z., Wang, K., Zhao, Z., & Hämäläinen, T. (2024). Energy-Efficient and Privacy-Preserved Incentive Mechanism for Mobile Edge Computing-Assisted Federated Learning in Healthcare System. IEEE Transactions on Network and Service Management.

[34] Xu, G., Qi, C., Dong, W., Gong, L., Liu, S., Chen, S., ... & Zheng, X. (2022). A privacy-preserving medical data sharing scheme based on blockchain. IEEE journal of biomedical and health informatics, 27(2), 698-709.

[35] Liu, H., Gu, T., Shojafar, M., Alazab, M., & Liu, Y. (2022). OPERA: Optional dimensional privacy-preserving data aggregation for smart healthcare systems. IEEE Transactions on Industrial Informatics, 19(1), 857-866.

[36] Tian, Y., Wang, S., Xiong, J., Bi, R., Zhou, Z., & Bhuiyan, M. Z. A. (2023). Robust and privacy-preserving decentralized deep federated learning training: Focusing on digital healthcare applications. IEEE/ACM Transactions on computational biology and bioinformatics.

[37] Namakshenas, D., Yazdinejad, A., Dehghantanha, A., & Srivastava, G. (2024). Federated quantum-based privacy-preserving threat detection model for consumer internet of things. IEEE Transactions on Consumer Electronics.

[38] Yazdinejad, A., Dehghantanha, A., Karimipour, H., Srivastava, G., & Parizi, R. M. (2024). A robust privacy-preserving federated learning model against model poisoning attacks. IEEE Transactions on Information Forensics and Security.

[39] Yazdinejad, A., Dehghantanha, A., & Srivastava, G. (2023). AP2FL: Auditable privacy-preserving federated learning framework for electronics in healthcare. IEEE Transactions on Consumer Electronics, 70(1), 2527-2535.

[40] Nazari, H., Yazdinejad, A., Dehghantanha, A., Zarrinkalam, F., & Srivastava, G. (2024). P3GNN: A Privacy-Preserving Provenance Graph-Based Model for APT Detection in Software Defined Networking. arXiv preprint arXiv:2406.12003.

[41] Jaddi, N. S., Abdullah, S., Goh, S. L., Nazri, M. Z. A., Othman, Z., Hasan, M. K., & Alvankarian, F. (2025). Multi-Population Kidney-Inspired Algorithm with Migration Policy Selections for Feature Selection Problems. IEEE Access.

[42] Dhasarathan, C., Shanmugam, M., Kumar, M., Tripathi, D., Khapre, S., & Shankar, A. (2024). A nomadic multi-agent based privacy metrics for e-health care: a deep learning approach. Multimedia Tools and Applications, 83(3), 7249-7272.

[43] Taiello, R., Cansiz, S., Vesin, M., Cremonesi, F., Innocenti, L., Önen, M., & Lorenzi, M. (2024, October). Enhancing Privacy in Federated Learning: Secure Aggregation for Real-World Healthcare Applications. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 204-214). Cham: Springer Nature Switzerland.

[44] Commey, D., Hounsinou, S., & Crosby, G. V. (2024). Securing Health Data on the Blockchain: A Differential Privacy and Federated Learning Framework. arXiv preprint arXiv:2405.11580.

[45] Praveen, S. Phani, et al. "Enhanced feature selection and ensemble learning for cardiovascular disease prediction: hybrid GOL2-2 T and adaptive boosted decision fusion with babysitting refinement." Frontiers in Medicine 11 (2024): 1407376.

[46] Ali, S., Li, Q., & Yousafzai, A. (2024). Blockchain and federated learning-based intrusion detection approaches for edge-enabled industrial IoT networks: A survey. Ad Hoc Networks, 152, 103320.

[47] Fan, X., Hu, H., Li, Y., & Gao, H. (2023). A privacy-preserving scheme for secure healthcare data sharing based on blockchain and intelligent contract. Future Generation Computer Systems, 131, 60-71.

[48] Bharathi, M., & Srinivas, T. A. S. Federated Learning Unveiled: From Practical Insights to Bold Predictions.

[49] Sharma, T., Singh, A., Raj, G., Sar, A., Choudhury, T., Saraf, S., & Dewangan, B. K. (2024, June). AES vs AES_GCM for Data Protection: A Comprehensive Security Comparison. In 2024 OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 4.0 (pp. 1-7). IEEE.

[50] Zahid, M. J. S. (2025). Optimizing Authenticated Encryption: High-Performance Implementations of ChaCha20 and Blake3 for Large-Scale Data.

[51] Harris, Steve; Lai, Wai Shing (2024). Example (synthetic) electronic health record data. University College London. Dataset. https://doi.org/10.5522/04/25676298.v1

[52] Pujari, M., & Pakina, A. K. (2024). EdgeAI for Privacy-Preserving AI: The Role of Small LLMs in Federated Learning Environments. *International Journal of Engineering and Computer Science*, 13(10), 26589–26601.

[53] Chen, C., Feng, X., Li, Y., Lyu, L., Zhou, J., Zheng, X., & Yin, J. (2024). Integration of large language models and federated learning. *Patterns*, 5(12).

[54] Mahadik, S. S., Pawar, P. M., Muthalagu, R., Prasad, N. R., Hawkins, S. K., Stripelis, D., *et al.* (2024). Digital privacy in healthcare: State-of-the-art and future vision. *IEEE Access*, 2024.

[55] Wenhua, Zhang, et al. "A lightweight security model for ensuring patient privacy and confidentiality in telehealth applications." Computers in Human Behavior 153 (2024): 108134.

[56] Dhasaratha, Chandramohan, et al. "Data privacy model using blockchain reinforcement federated learning approach for scalable internet of medical things." CAAI Transactions on Intelligence Technology (2024).

[57] Zhou, Y., Rashid, F.A.N., Mat Daud, M., Hasan, M.K. and Chen, W., 2025. Machine Learning-Based Computer Vision for Depth Camera-Based Physiotherapy Movement Assessment: A Systematic Review. Sensors, 25(5), p.1586.

[58] Ghazal, T. M., Hasan, M. K., Abdullah, S. N. H., Abubakkar, K. A., & Afifi, M. A. (2022). IoMT-enabled fusion-based model to predict posture for smart healthcare systems. Computers, Materials and Continua, 71(2), 2579-2597.

Dr. Taher M. Ghazal, a distinguished IEEE Senior Member, is a seasoned academician with a comprehensive educational background. He earned his Bachelor of Science in Software Engineering from Al Ain University in 2011, followed by a Master of Science degree in Information Technology Management from The British University in Dubai, associated with The

University of Manchester and The University of Edinburgh in 2013. Dr. Ghazal culminated his academic journey with a Doctorate in Information Science and Technology from Universiti Kebangsaan Malaysia in 2023. Dr. Ghazal's scholarly pursuits encompass various interests, including Cybersecurity, Artificial Intelligence, IoT, Information Systems, Software Engineering, Big Data, Quality of Software, and Project Management. He is actively engaged in community service through his involvement in impactful projects and research endeavors.

Associate Professor Ts. Dr. SHAYLA ISLAM is a Deputy Director and Head of Research at the Institute of Computer Science and Digital Innovation (ICSDI), UCSI University, Malaysia. Recently, she was selected as one of the Top 2% of scientists worldwide by Stanford University USA and Elsevier BV. Dr. Shayla Islam was awarded Professional Technologist by the Malaysia Board of Technologists (MBoT). The Institute of Electrical and Electronics Engineers (IEEE) also awarded her a Senior Member. She is a Graduate Engineer of the Board of Engineers Malaysia and the Institution of Engineers Bangladesh (MIEB-M/40624). Her research interests include Mobile Networks in a 5G environment, Telecommunications, Cyber-Physical, Artificial Intelligence, and Network Security. Moreover, she has published more than HUNDREDS (100) WoS/Scopus-indexed articles with high-impact factors, and TEN (10) of these articles have been in the top TEN (10) journals in the area. Currently, her H-index is 27, with 2,585 citations in Scopus. She has also evaluated her Ph.D. thesis at the international level as the foreign examiner. Dr. Shayla has been guest editor and reviewer for international, prestigious, and high-impact factor journals such as Expert System with Application, and IEEE T-ITS, TCE, etc.

Mohammad Kamrul Hasan (M'13–SM'17) is currently serving as an Associate Professor at the Faculty of Information Science and Technology, within the Center for Cyber Security, Universiti Kebangsaan Malaysia (UKM). He earned his Ph.D. in Electrical and Communication Engineering from the International Islamic University Malaysia in 2016. Dr. Kamrul specializes in Cyber Security within advanced information-centric networks, with research interests spanning computer networks, data communication and security, mobile networks and privacy protection, cyber-physical systems, industrial IoT, transparent AI, and electric vehicle networks. He has published over 300 indexed papers in reputable journals and conference proceedings. He is a Senior Member of the IEEE, a member of the Institution of Engineering and Technology (IET), and the Internet Society, and is recognized as a Certified Professional Technologist in Malaysia. Dr. Kamrul has been actively involved in numerous IEEE Malaysia Section events, workshops, and training programs, including contributions to IEEE Humanity initiatives. He previously served as Chair of the IEEE Student Branch from 2014 to 2016. He contributes to the academic community as a guest editorial board member for prestigious journals such as IEEE Transactions on Consumer Electronics (TCE), Elsevier -eTransportation, and MDPI- energies, bigdata and serves as an associate Editor for IET journals. Additionally, he frequently acts as a conference chair, speaker, and workshop facilitator, fostering academic collaboration and knowledge exchange. Beyond academia, Dr. Kamrul has been secured national and internation grants amounting five hundred thousand USD, and deeply committed to scientific community, social welfare and community service, actively volunteering to support underprivileged communities and promoting knowledge sharing for societal benefit.

Dr. AHMAD A. Abu-Shareha's research interests focus on applying machine learning and artificial intelligence across various domains, including networks, medical information processing, and knowledge construction and extraction. He has explored numerous machine learning algorithms, emphasizing their practical use in improving efficiency and accuracy in diverse applications.

MUHAMMAD ATTIQUE KHAN (Member IEEE) earned his Master's and Ph.D. degrees in Human Activity Recognition for the Application of Video Surveillance and Skin Lesion Classification using Deep Learning from COMSATS University Islamabad, Pakistan, in 2018 and 2022. He is an Artificial Intelligence Department Assistant Professor at Prince Mohammad Bin Fahd University, Saudi Arabia. Previously, he was affiliated with HITEC University Taxila, Pakistan. His primary research focus in recent years is medical imaging, MRI analysis, Video Surveillance, Human Gait Recognition, Remote Sensing, and Agriculture Plants using Deep Learning. He has more than 340 publications that have more than 18,000+ citations and an impact factor of 1220+ with an h-index of 76 and i-index of 250.

JAMEL BAILI received the B.Sc., M.Sc., and Ph.D. degrees in electronics from the University of Monastir, Monastir, Tunisia, in 2001, 2003, and 2009, respectively.,From 2003 to 2013, he was a Research Assistant with the Micro-Electronics and Instrumentation Laboratory. From 2010 to 2013, he was an Assistant Professor with the Electronics Department, University of Sousse. Since 2013, he has been an Assistant Professor with the Engineering Department, College of Computer Science, King Khalid University, Abha, Saudi Arabia. His research interests include embedded systems, instrumentation, digital signal processing, and AI.

ALI Q SAEED is working with the Computer Center, Northern Technical University, Ninevah, Iraq. His area of interest includes Deep Learning, Transfer Learning, Convolutional Neural Network, Data Augmentation, Learning Algorithms, Machine Learning, Training Set, F1 Score, Machine Learning Models, Artificial Intelligence, Augmentation Techniques and Convolutional Neural Network Model

MOHAMMED WASIM BHATT is working at the Model Institute of Engineering and Technology Jammu, J&K 181122, India. His area of interest includes Convolutional Neural Network, Attention Mechanism, Consumer Electronics, Consumer Technology, Internet Of Things, LSTM Network, Spatial Features, Spatiotemporal Characteristics, Action Recognition, Adverse Selection, Anomaly Detection and Artificial Intelligence.

MUNIR AHMAD (Senior Member, IEEE) received a master's degree in computer science from the Virtual University of Pakistan, Pakistan, and a Ph.D. in computer science from the School of Computer Science, National College of Business Administration and Economics. He is a distinguished professional with over 16 years of experience. As the Executive Director/CIO at United International Group, Lahore, Pakistan, he has excelled in data management and resource optimization within multinational organizations. He is renowned for his extensive research in sentiment analysis, AI applications in healthcare, and animal facial identification. His expertise lies in data mining, big data, and artificial intelligence.

Robust and Privacy-Preserving Decentralized Deep Federated Learning Training: Focusing on Digital Healthcare Applications

Youliang Tian, Shuai Wang, Jinbo Xiong, Renwan Bi, Zhou Zhou, and Md Zakirul Alam Bhuiyan

***Abstract*—Federated learning of deep neural networks has emerged as an evolving paradigm for distributed machine learning, gaining widespread attention due to its ability to update parameters without collecting raw data from users, especially in digital healthcare applications. However, the traditional centralized architecture of federated learning suffers from several problems (e.g., single point of failure, communication bottlenecks, etc.), especially malicious servers inferring gradients and causing gradient leakage. To tackle the above issues, we propose a robust and privacy-preserving decentralized deep federated learning (RPDFL) training scheme. Specifically, we design a novel ring FL structure and a Ring-Allreduce-based data sharing scheme to improve the communication efficiency in RPDFL training. Furthermore, we improve the process of distributing parameters of the Chinese residual theorem to update the execution process of the threshold secret sharing, supporting healthcare edge to drop out during the training process without causing data leakage, and ensuring the robustness of the RPDFL training under the Ring-Allreduce-based data sharing scheme. Security analysis indicates that RPDFL is provable secure. Experiment results show that RPDFL is significantly superior to standard FL methods in terms of model accuracy and convergence, and is suitable for digital healthcare applications.**

***Index Terms*—Decentralized training, deep learning, digital healthcare, privacy-preserving federated learning, robust federated learning.**

Manuscript received 4 April 2022; revised 17 November 2022; accepted 23 January 2023. Date of publication 3 March 2023; date of current version 8 August 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFB3101100; in part by the National Natural Science Foundation of China under Grants 62272123, 62272102, 61872088, and U1836205; in part by the Henan Key Laboratory of Network Cryptography Technology under Grant LNCT2021-A02; in part by the Project of High-level Innovative Talents of Guizhou Province under Grant [2020]6008; in part by the Science and Technology Program of Guiyang under Grant [2021]1-5; in part by the Science and Technology Program of Guiyang under Grant [2022]2-4; in part by the Science and Technology Program of Guizhou Province under Grants [2020]5017 and [2022]065. *(Corresponding author: Jinbo Xiong.)*

Youliang Tian, Shuai Wang, and Zhou Zhou are with the State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang, Guizhou 550025, China (e-mail: youliangtian@163.com; wshuai957@gmail.com; mitzhouzhou@163.com).

Jinbo Xiong and Renwan Bi are with the Fujian Provincial Key Laboratory of Network Security and Cryptology, College of Computer and Cyber Security, Fujian Normal University, Fuzhou, Fujian 350117, China, and also with the Henan Key Laboratory of Network Cryptography Technology, Zhengzhou, Henan 450001, China (e-mail: jbxiong@fjnu.edu.cn; brw2806@163.com).

Md Zakirul Alam Bhuiyan is with the Department of Computer and Information Sciences, Fordham University, New York, NY 10458 USA (e-mail: zakirulalam@gmail.com).

Digital Object Identifier 10.1109/TCBB.2023.3243932

I. Introduction

FEDERATED learning (FL) of deep neural networks (DNN) is playing an important role in wearable healthcare [1], smart healthcare system [2], [3], [4], industrial IoT [5], [6], etc. It is clear that deep learning based services are impacting our lives from 5G communications, autonomous driving, social, economic, educational and many other aspects [7], [8], [9], [10]. Particularly, deep learning approach creates robust numerical models from digital healthcare data. Traditional digital healthcare data is not entirely captured or leveraged by deep learning due to privacy leakage or concerns with limited and timely access to the data. Massive training data is the basis for deep learning to obtain well-formulated models. However, in digital healthcare systems and collaborative healthcare organizations, data is typically scattered among different healthcare organizations or branches, which makes it hard to interoperate information due to the prevalence of information barriers in various healthcare organizations, resulting in the phenomenon of ”data silos”. Specifically, the General Data Protection Regulation (GDPR), implemented by the Europe, aims to strengthen the autonomous control and security protection of user (patient, doctor, staff, etc.) privacy, making it more difficult to break through the barriers of data silos [11]. Fortunately, federated learning proposes a new approach to these challenges [12], [13], [14]. The concept of FL was pioneered by Google Research in 2016 [15], and the technique aims to satisfy the demands of privacy protection and data security by designing a machine learning framework that enables various institutions to collaborate without exchanging data to improve the performance of machine learning.

The traditional star topology for FL has a central server coordinating multiple participants (clients) to solve the machine learning problem [16], [17]. The central server of this centralized framework faces communication pressure and bandwidth bottlenecks, while heavy reliance on the participation of the central server can lead to single point of failure and poor scalability [18]. Phong et al. [19] demonstrates that the honest but curious (HbC) servers may infer local data information from intermediate parameters returned by clients causing privacy breaches and that centralized frameworks may have security issues [20]. Similarly, some blockchain-based FL schemes [5], [21], [22] do not consider the dynamics of participants (e.g., node dropping out and joining), and most of these schemes are based on smart contracts in public chains, which leads

to limited system performance and failure to protect the privacy of models. Next-generation network is expected to be underpinned by new forms of decentralized, infrastructure-free communication paradigms [23] that enables devices to cooperate directly via device-to-device (D2D) spontaneous connections (e.g., multi-hop or mesh), which are designed to require no support from a central coordinator or to provide limited support for synchronization and signaling. They are typically deployed in mission-critical control applications where edge nodes cannot rely on remote devices for fast feedback and must manage some of their computational tasks locally [24], [25], cooperating with neighboring nodes to self-disclose information. In addition, some decentralized FL schemes [26], [27], [28] design different structures to overcome the problem of low communication efficiency, without considering the model security. Therefore, a robust and privacy-preserving decentralized FL training scheme is essential to be designed, especially for the digital healthcare applications containing much of sensitive data for both patients and healthcare organizations.

In this paper, we propose a robust and privacy-preserving decentralized deep federated learning (RPDFL) training scheme, focus on data privacy in digital healthcare applications. The core idea of RPDFL is to apply the FL specification to an effective logical ring architecture that no longer relies on the coordination of a central server. Inspired by the Ring-Allreduce algorithm,[1] the main idea of RPDFL is to build a ring network structure of decentralized healthcare edge devices, solving the problems of poor scalability, performance bottlenecks and single point of failure of centralized FL. Second, a novel model aggregation and data sharing method is also designed in RPDFL to improve decentralized FL performance and bandwidth utilization. Additionally, we support healthcare edges to drop out during RPDFL training without causing data leakage and ensuring the robustness of the RPDFL training by updating the execution process of the threshold secret sharing scheme. The main contributions of the proposed RPDFL are summarized as follows:

- We propose a RPDFL training scheme for digital healthcare organizations. In RPDFL, we design a novel ring FL structure and a Ring-Allreduce-based data sharing scheme, which improves the communication efficiency in decentralized training schemes and overcomes the poor scalability and single-point-of-failure problems of centralized FL.
- We improve the process of distributing parameters of the Chinese residual theorem (CRT) to update the execution process of the threshold secret sharing, supporting healthcare edge to drop out during the training process without causing data leakage, and ensuring the robustness of the RPDFL training under the Ring-Allreduce-based data sharing.
- We prove the security of RPDFL under HbC model, and evaluate the RPDFL training scheme on the MNIST dataset. Experiment results show that RPDFL significantly outperforms two standard model learning methods, i.e., FedAvg [29] and Gossip learning [26], in terms of model accuracy and convergence, and is suitable for digital healthcare applications.

The rest of this paper is organized as follows: related works and preliminaries are presented in Sections II and III. In Section IV, we describe the problem specification, system architecture and design goals, respectively. In Section V, we describe the construction details of the RPDFL scheme. Section VI analyzes the RPDFL security from the aspects of correctness and confidentiality. Next, Section VII presents the performance evaluation of the scheme. Finally, Section VIII summarizes the entire paper.

II. Related Work

Decentralized Federated Learning (DFL) is a fully decentralized framework proposed on the basis of FL to alleviate the dependence on a central server, allowing clients to take full advantages of machine learning algorithms without disclosing raw training data [20], [30], and by designing methods that enable nodes to share model parameters with each other without the need for central server coordination to obtain better models.

Some research focuses on solving the single point of failure and communication bottleneck problems associated with traditional FL star architecture by designing serverless architectures with different network topologies. Yang et al. [18] proposed an E-Tree decentralized FL model learning approach, which utilizes a well-designed tree structure on edge devices and optimizes the tree structure and the location and order of aggregation in the tree to improve the convergence of training and model accuracy. Wang et al. [28] proposed a decentralized FL scheme for deep generative models based on ring topology. The algorithm designed a novel ring FL topology and a map-reduce based synchronization method to improve the performance and bandwidth utilization of decentralized FL. Several studies have exploited the decentralized nature of blockchain to be highly compatible with the goals of DFL. Kim et al. [22] proposed a blockchain FL architecture in which participating clients are able to store local model updates on blocks, while all client nodes as miners can access and aggregate updates through smart contracts without the need for a single central server. Peng et al. [21] proposed VFChain, a verifiable, auditable FL framework for blockchain-based systems, where verifiability is achieved by the blockchain selecting a committee to collectively aggregate models and record verifiable evidence in the blockchain; and to provide auditability, a new blockchain authentication data structure is proposed to improve the efficiency of the search for verifiable evidence and to support secure rotation of committees. Weng et al. [31] proposed DeepChain, a distributed, secure, and fair DL framework, which provides blockchain-based value-driven incentives that force participants to perform correctly. At the same time, DeepChain guarantees the data privacy of each participant and provides auditability for the entire training process.

Approaches that do not require the coordination of central server by designing different consensus mechanisms within the nodes are gradually gaining attention from researchers. Hegedus et al. [26] proposed Gossip learning using the Gossip protocol,

[1]https://github.com/baidu-research/baidu-allreduce

assuming that the data remains on the edge devices, but it does not require an aggregation server or any central component. Experiments demonstrate that Gossip learning is better than FL with uniformly distributed training data on nodes for all scenes, and the overall performance is comparable to that of traditional FL. Savazzi et al. [27] proposed a fully distributed (or serverless) learning approach, where the proposed FL algorithm interacts through iterative local computation and consensus-based methods that exploit the network between devices performing data operations cooperation. The approach lays the foundation for integrating FL within 5G and networks characterized by decentralized connectivity and computation.

A growing body of literature focuses on the privacy leakage associated with device dropouts in complex networks [32], [33], [34]. Mandal et al. [35] proposed a privacy-preserving system PrivFL, which contains an additive homomorphic encryption scheme and an aggregation protocol for privacy protection, and they demonstrated that the system can guarantee data and model privacy while supporting device dropout under a semi-honest security model. Wu et al. [36] discussed a series of security and efficiency issues associated with end-device dropouts in mobile edge computing environments and designed a multilayer federated learning protocol called HybirdFL, which introduces regional slack factors to mitigate the impact of end-device dropouts. Liu et al. [37] proposed a scalable privacy-preserving aggregation scheme that tolerates participants to drop out at any time and proves that both semi-honest and active malicious adversaries are security.

Remarkably, traditional medical applications normally train models by collecting data from all healthcare organizations. However, in real applications, there are difficulties in sharing information between different healthcare organizations due to data privacy issues. It makes FL well suited for digital healthcare applications. Chen et al. [1] designed FedHealth, a FL framework for wearable medical devices, without loss of privacy and security. Similarly, Yin et al. [38] allowed the use of DNN with Gaussian processes to detect infected individuals, and the proposed FedLoc framework accomplished accurate localization services without compromising the privacy of mobile users. Li et al. [2] designed an Alzheimer's disease detection system based on FL and differential privacy, protecting the information privacy of users and the security of the scheme.

Furthermore, in addition to data privacy issues, Zhang et al. [3] focus on the imbalance in the health data characteristics provided by users participating in digital healthcare applications, which cannot be addressed in the solutions provided by existing smart health applications due to the strict privacy requirements of participants and the heterogeneous resource constraints of edge devices. Therefore, they proposed FedSens, which performs well despite the presence of severe tired imbalance conditions. In particular, Lim et al. [39] adjusted their research perspective to note that participants in FL networks may have different willingness to participate [40] and designed incentive mechanisms to motivate users to participate in FL training. Meanwhile, users may be reluctant to participate in collaborative model training due to the model training consumes device energy and requires stable wireless connections to limit the dropout rate [41].

TABLE I
NOTATIONS AND DESCRIPTIONS

Notations	Descriptions
U	Collection of online users participating in federal training
D_i^j	User i Randomly selected sub-training set of batch-size for the jth training round
ω_{global}	Global model
ω_i^j	Model parameters at the end of the jth iteration of training for user i
ω_i, k^j	Share of model ω_i^j sent by user i to user k
$L_f(D_k, \omega)$	Loss function
z_k^i	Share of blinded model parameters z^i sent by user i to user k
φ_k	The kth final result of the RingAllreduce data share
φ	The end result of Ring-Allreduce data sharing
ω_{sum}^i	Blinded aggregation models
ω_{sum}	Local model aggregation model for the current round of federated training

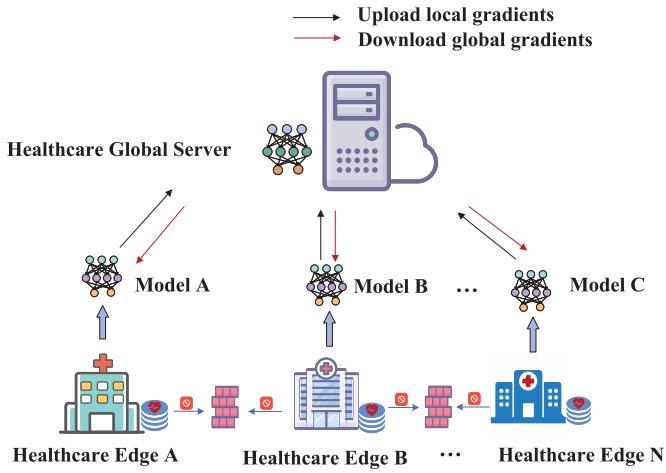


Fig. 1. Deep federated learning training process in digital healthcare applications.

Digital healthcare applications still suffer from threats such as participant dropouts, resource constraints of edge devices, and class imbalance problems of data heterogeneity, while existing research efforts in digital healthcare applications either consider only participant dropouts and data security, or only data heterogeneity. Unfortunately, none of these considerations meets the requirements of the system in real-world applications. To the best of our knowledge, there is no solution satisfying both robustness and strict privacy-preserving requirements.

III. PRELIMINARIES

A. Deep Federated Learning

Federated learning is an emerging machine learning paradigm where multiple clients (healthcare organizations) solve machine learning problems under the coordination of a central aggregation server in digital healthcare systems. The training data for each client exists locally, and there is no data exchange between clients. In each iteration, clients generate local models by executing training local data and uploading them to the aggregation server. The aggregation server aggregates the local models received in one cycle to obtain the global model, and returns the global model to the online client, which updates the local model. This process is repeated until the loss function converges or the maximum number of iterations is reached [42]. As shown in Fig. 1, a typical deep federated learning training process in digital healthcare applications is as follows:

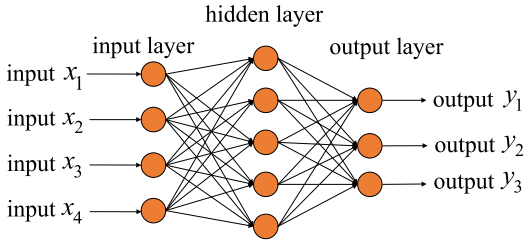


Fig. 2. Neural network structure.

- Client selection. The aggregation server (healthcare global server) randomly selects a group of clients (healthcare edges) that meet the eligibility requirements.
- Initialize model. Selected clients download the current model weights and training parameters from the server (healthcare global server).
- Client computation. By performing the training process, each selected client computes updates of the local model locally, e.g. by running SGD on local training data (e.g. Federated Averaging [29]).
- Model aggregation. The server collects aggregation of local model updates from devices. For efficiency, once a sufficient number of clients have uploaded local models in a given cycle, the server may simply discard some gradients that have not been uploaded by clients in time and perform aggregation.
- Model update. The server broadcasts the global model after aggregation to the clients participating in the current round, and the clients update the local model.

1) Deep Neural Networks (DNNs): Fig. 2 is a simple three-layer fully connected network, each layer contains multiple neurons, and two adjacent layers are interconnected by weights ω. The neural network can be represented as a function $f(x,\omega) = \tilde{y}$, where denotes the input value x, and $\tilde{y}$ denotes the predicted value calculated by the function f[9].

2) FL Update Procedure: We suppose there are n clients i $(i \in \boldsymbol{U},\ |\boldsymbol{U}| = N)$, and the number of clients randomly selected participate in federated training in each round are $k (k \in \boldsymbol{K}, \boldsymbol{K} \subseteq \boldsymbol{U})$, and suppose the training dataset for each client is represented as $\boldsymbol{D}_k = \{\langle x_j, y_j\rangle, k = 1, 2, \cdots T\}$. Then, the loss function of this training data set can be defined as:

$$L_f(\boldsymbol{D}_k,\omega) = \frac{1}{|\boldsymbol{D}_k|} \sum_{\langle x_j,y_j\rangle \in \boldsymbol{D}_k} L_f(x_j,y_j;\omega), \tag{1}$$

where we can take $L_f(x_j,y_j;\omega)$ to be a specific function $l_f x_j, y_j; \omega$ and the goal of FL training is to make $\min_{\omega\in\mathbb{R}} l_f(x_j,y_j;\omega)$. In our scheme, we use the stochastic gradient descent algorithm [20], [43] to achieve this goal:

$$\omega^{j+1} = \omega^j - \alpha \nabla L_f({\boldsymbol{D}_k}^j, \omega^j)$$
$$\text{where} \quad \nabla L_f({\boldsymbol{D}_k}^j,\omega^j) \stackrel{\text{def}}{=} \frac{\sum_{k\in K} \Delta {v_k}^j}{\sum_{k\in K} {\boldsymbol{D}_k}^j} \tag{2}$$

where ω^j denotes the parameter at the end of the jth iteration, ${\boldsymbol{D}_k}^j$ denotes a random subset of the client's local training data set $\boldsymbol{D}_k$, α is the learning rate, and $\Delta {v_k}^j = |{\boldsymbol{D}_k}^j| \nabla L_f({\boldsymbol{D}_k}^j, \omega^j)$ is the gradient that each client computes locally and uploads to the server. The aggregation server selects a random subset $k \in \boldsymbol{K}$ in the jth iteration, and each client $(\boldsymbol{K} \subseteq \boldsymbol{U})$ randomly selects a subset ${\boldsymbol{D}_k}^j \subseteq \boldsymbol{D}_k$ to execute the stochastic gradient descent algorithm.

B. Ring-Allreduce

Ring-Allreduce is a distributed statute algorithm in high performance computing that makes full use of bandwidth and can solve the communication bottleneck problem in systems with a large number of nodes involved. Its main idea is divided into two steps, Scatter-Reduce and Allgather.

1) RA.Scatter-Reduce. First, each node divides the data into N copies (N is the total number of nodes) and will perform $n-1$ Scatter-Reduce iterations. In each iteration, a node will send one copy of data to its right neighbor and receive one copy of data from its left neighbor and accumulate it to that copy. The data sent and received by each node is different in each iteration; the nth node starts by sending the nth copy of data and receiving the $(n-1)$th copy of data, and each iteration sends the data received by that node in the previous iteration.

$$\text{RA.Scatter - Reduce}\left(\{s_1, s_2, \cdots s_n\}\right) \rightarrow \left\{s^i\right\}_{i\in N} \tag{3}$$

where $\{s^i\}_{i\in N}$ denotes one copy of the final result and $\{s_1, s_2, \cdots s_n\}$ denotes N copies of the node's data.

2) RA.Allgather. After completing the Scatter-Reduce step, each node is guaranteed to have a subset of the final data. Allgather and Scatter-Reduce are processed similarly, with the nth node first sending $n+1$ copies of the data and receiving the nth copy, overwriting the original data with the received data, and then always sending the just-received data in subsequent iterations. RA.Allgather$(\{s^i\}_{i\in N}) \rightarrow \{s\}$, where $\{s\}$ denotes the final result.

In RPDFL, each of the N clients sends and receives $N-1$ scatter-reduces and $N-1$ allgathers. Clients will send all $\frac{K}{N}$ values, where K is the total number of values summed over the different clients in the array. Hence, the total amount of transmitted data to and from each client is $2(N-1)\frac{K}{N}$, which is irrelevant to the total number of clients.

C. Secret Sharing Protocol

In RPDFL, we utilize the CRT for secret sharing due to the additivity of the CRT and the smaller ciphertext space [44], and encrypt $\Delta {v_k}^j$ using the secret sharing technique. Specifically, we represent secret sharing formally as follows.

Let $m_1, m_2, \cdots m_n$ be n integers greater than 1 that satisfy:

$$m_1 \le m_2 \le \cdots m_n,\ \gcd(m_i, m_j) = 1 (i \ne j)$$
$$m_1 m_2 \cdots m_t > m_n m_{n-1} \cdots m_{n-t+2}$$

Also let s be the secret data satisfying $m_n m_{n-1} \cdots m_{n-t+2} < s < m_1 m_2 \cdots m_t$. Calculate $M = m_1 m_2 \cdots m_n$, $s_i \equiv s (\mathrm{mod}\ m_i)$ $(i = 1, 2, \cdots n)$ and take (s_i, m_i, M) as a sub secret key,

the set $\{(s_i, m_i, M)\}_{i=1}^{n}$ constitutes a (t, n) threshold scheme. Since, out of t participants (denoted as $i_1, i_2, \ldots, i_t$), each i_j calculates:

$$\begin{cases} M_{i_j} = \frac{M}{m_{i_j}} \\ N_{i_j} \equiv {M_{i_j}}^{-1} (\mathrm{mod}\ m_{i_j}) \\ y_{i_j} = s_{i_j} M_{i_j} N_{i_j} \end{cases} \tag{4}$$

We can find the solution to the system of equations:

$$\begin{cases} s \equiv s_{i_1} (\mathrm{mod}\ m_{i_1}) \\ s \equiv s_{i_2} (\mathrm{mod}\ m_{i_2}) \\ \vdots \\ s \equiv s_{i_t} (\mathrm{mod}\ m_{i_t}) \end{cases} \tag{5}$$

Based on the CRT, $s = \sum_{i=1}^{t} y_{i_j} (\mathrm{mod} \prod_{j=1}^{t} m_{i_j})$.

1) S.share$(s, t, U) \rightarrow \{(n, s_n)\}_{n \in U}$: given a secret s and a threshold $t \leq |\boldsymbol{U}| = N$, generate n shares of the secret s_n.
2) S.recon$(\{(n, s_n)\}_n \in \boldsymbol{M}, t) \rightarrow s$: input a subset $\boldsymbol{M}$ of the secret shares, where $n \in \boldsymbol{M} \subseteq \boldsymbol{U}$ and $t \leq |\boldsymbol{M}|$, m outputs the secret s.

IV. Problem Statement

A. Problem Specification

In FL, when participants $\{1, 2, \ldots, n\}$ initiate FL training, each participant needs to submit its local gradients to the central aggregation server, which returns the current round of aggregation results to the participant after completing the aggregation of the gradients. In contrast to traditional FL, in a RPDFL system, the way in which the gradients are aggregated changes without the involvement of the central aggregation server, and the way in which the global model is synchronised and updated changes. The model gradients are directly transferred between clients, and there is a model privacy leakage problem. In addition, some clients may drop out or go offline during the FL process, resulting in their privacy potentially being inferred or other online clients not being able to decrypt the aggregation results correctly. The issues can be summarised as ① how to aggregate in gradients in a DFL system; ② how to synchronise global model updates; ③ how to protect privacy during model aggregation; and ④ how to support protocol robustness.

In order to solve the above problem, first, inspired by the Ring-allreduce algorithm, we join the clients participating in FL into a logical ring, specifying that each client should have a left neighbour node and a right neighbour node, and that the client will only send data to its right neighbour node and receive data from its left neighbour node. Communication between clients is performed using the Ring-allreduce algorithm, which ensures that the global model is updated synchronously. Second, we protect the privacy of the gradients using a threshold secret sharing protocol and allow for client exits during FL training to ensure robustness of the protocol. Furthermore, we compare with several existing schemes, as shown in detail in Table II.

TABLE II
Detailed Comparisons of the RPDFL With Related Works

	Data Privacy	Robustness to Failures	Scalability
Gossip Learning [26]	×	×	×
RDFL [28]	×	×	×
BFLC [45]	×	√	√
E-Tree [18]	×	×	√
RPDFL	√	√	√

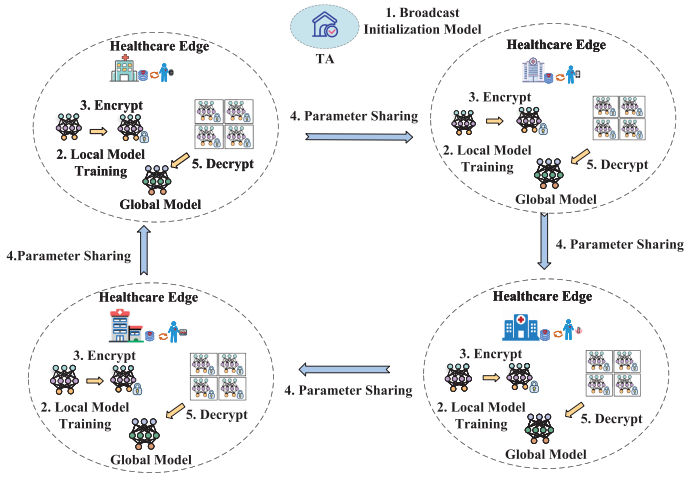


Fig. 3. Systems architecture of RPDFL.

B. Systems Architecture

As shown in Fig. 3, the RPDFL system model consists of two parts, the Trusted Authority (TA) and Healthcare Edges(.

- Trusted Authority (TA). The main recognition of the TA is to initialise the whole system, generate the public parameters and assign the public-private key pairs required for the protocol to each healthcare edge. Afterwards, it will go offline unless a dispute arises.
- Healthcare Edges. We specify that the healthcare edge can only receive data from the left neighbour edge and send data to the right neighbour edge. The edge performs local training and then generates a local model, which is encrypted according to the protocol and sent to the right neighbor. After each edge receives a certain number of encrypted local models in a certain period, the global model can be derived through protocol aggregation. Finally, the local model is updated to the global model.

C. Design Goals

To overcome the various challenges posed by the complex and dynamic web environment for federated learning, we desire RPDFL to have the following properties: ① data confidentiality: data can be securely aggregated without revealing the data privacy of the healthcare edges; ② data locality: local models sent by neighbouring edges are computed from the original data; ③ edge dynamics: delayed submission of local models or withdrawal of edges from training in a given iteration does not affect the correct execution of the protocol.

V. Construction of the RPDFL

In this section, we elaborately describe the main components of RPDFL: an improved CRT-based threshold scheme and a ring robust FL scheme.

A. Improved CRT-Based Threshold Scheme

TA generates n integers e_i $\{e_i|e_i > 1$ and $e_i \subseteq \mathbb{Z}_E\}$, e_i satisfying $e_1 \leq e_2 \leq \cdots \leq e_n$, $\gcd(e_i, e_j) = 1(i \neq j)$, $e_1 e_2 \cdots e_t > e_n e_{n-1} e_{n-t+2}$. Also construct function $\rho(x) = x - A$, the integer A chosen randomly by TA, where A satisfies $e_n e_{n-1} \cdots e_{n-t+2} < A < e_1 e_2 \cdots e_t$. Subsequently, TA divides the integer A into n shares using the Asmuth-Bloom's (t, n) threshold secret sharing protocol [44], and n is the number of healthcare edges participating in federated training in the current round of iterations. TA calculates$A_i \equiv A(\mathrm{mod}\ e_i)$ $(i = 1, 2, \cdots n)$ and $E = e_1 e_2 \cdots e_n$ which broadcasts A_i, e_i, and E to the edges participating in the federated training. With (A_i, e_i, E) as a sub secret key, the set $\{(A_i, e_i, E)\}_{i=1}^n$ constitutes a (t, n) threshold scheme. Let $m_i \subseteq \mathbb{Z}_E$, satisfying:

$$m_1 \leq m_2 \leq \cdots m_n,\ \gcd(m_i, m_j) = 1(i \neq j) \tag{6}$$

$$m_1 m_2 \cdots m_t > m_n m_{n-1} \cdots m_{n-t+2} \tag{7}$$

Let s^i be the secret of edge i. And let $z^i = s^i + A_i$, satisfying $m_n m_{n-1} \cdots m_{n-t+2} < z^i < m_1 m_2 \cdots m_t$. Calculate $M = m_1 m_2 \cdots m_n$, ${z_i}^i \equiv z^i(\mathrm{mod}\ m_i)(i = 1, 2, \cdots n)$. With $({z_i}^i, m_i, M)$ as a sub secret key, the set $\{({z_i}^i, m_i, M)\}_{i=1}^n$ also forms a (t, n) threshold scheme. Among t participants (denoted as $i_1, i_2, \ldots, i_t$), each i_j calculates:

$$\begin{cases} M_{i_j} = \frac{M}{m_{i_j}} \\ N_{i_j} \equiv {M_{i_j}}^{-1}(\mathrm{mod} m_{i_j}) \\ y_{i_j} = {z^i}_{i_j} M_{i_j} N_{i_j} \end{cases} \tag{8}$$

we can find the solution to the system of equations:

$$\begin{cases} z^i \equiv {z^i}_{i_1}(\mathrm{mod}\ m_{i_1}) \\ z^i \equiv {z^i}_{i_2}(\mathrm{mod}\ m_{i_2}) \\ \quad\vdots \\ z^i \equiv {z^i}_{i_t}(\mathrm{mod}\ m_{i_t}) \end{cases} \tag{9}$$

Based on the CRT, $z^i = \sum_{j=1}^t y_{i_j} \bmod \prod_{j=1}^t m_{i_j}$. The above procedures are to decrypt a copy of the secret z^i in the online edge when the number of secrets obtained from the edge decryption is greater than t.

$$\begin{aligned} &\sum_{i=1}^t z^i \bmod \prod_{j=1}^t e_{i_j} \\ &= (z^1 + z^2 + \cdots z^t) \bmod \prod_j^t e_{i_j} \\ &= ((s^1 + A_1) + (s^2 + A_2) + \cdots + (s^t + A_t)) \bmod \prod_j^t e_{i_j} \\ &= ((s^1 + s^2 + \cdots + s^t) + (A_1 + A_2 + \cdots A_t)) \bmod \prod_j^t e_{i_j} \end{aligned} \tag{10}$$

The value of $\prod_j^t e_{i_j}$ is required far greater than the value of $(s^1 + s^2 + \cdots + s^t)$. All online edges are calculated separately here.

$$\omega_{\text{sum}} = \rho\left(\left(s^1 + s^2 + \cdots + s^t\right) + A\right) \tag{11}$$

ω_{sum} denotes the client's local aggregated gradient, and the aggregated value of the gradient of the online edge for the current iteration is obtained. This scheme ensures that each edge is only aware of the gradient aggregation value and does not specifically obtain the gradient values of the remaining edges, thus achieving privacy preserving.

B. Implementation Process of RPDFL

Fig. 4 shows RPDFL implementation procedure, where the task is completed in three rounds. We describe the processes of RPDFL from the following four phases.

1) Initialization Phase: In the initialization phase, the system parameters are assigned by the TA [28], and the TA randomly selects an integer A, generates a share $\{(A_i, e_i, E)\}_{i \in \boldsymbol{U}} \leftarrow$ S.share$(A, t, \boldsymbol{U})$ of A, so that each edge has a copy of the (A_i, e_i, E) sub-secret key. TA initializes the system model ω_{global} and broadcasts the initialized model ω_{global}, function $\rho(x)$, and sub secret key (A_i, e_i, E) to each edge. Healthcare edge has received the ω_{global}, $\rho(x)$, and (A_i, e_i, E). Healthcare edge updates the local model to the global model and trains the model on the local dataset.

In this paper, we join all healthcare edges participating in federated training into a logical ring structure, and if a healthcare edge does not pass the parameters to its right neighbor node within period μ is considered as that healthcare edge dropped. We generate a k-dimensional empty array instead of that healthcare edge to participate in the RA.Scatter-Reduce() and RA.Allgather() data sharing processes for this round of federated training. In the next federated training, we select a healthcare edge from the candidate healthcare edge cluster to join the logical ring and perform the next round of federated training. The detailed logic ring update is illustrated in Fig. 5.

2) Model Training Phase: Healthcare edge performs SGD updates to the received global model on the local training dataset, generating the local model in multiple iterations, and the local model transmits gradients according to the Ring-Allreduce algorithm.

3) Gradient Aggregation Phase: The dropping out of healthcare edge is supported during the RPDFL training process because we use threshold secret sharing scheme. Briefly, in order for each edge to obtain the aggregated values of the gradients of all participating edges without revealing the local model gradient values, each edge shares the local model gradients with all other online edges by using a modified CRT-based threshold secret sharing scheme. Thus, if healthcare edge i drops out of ring robust FL within a certain period μ, the remaining edges can eliminate the random number A_i based on having received

- Round 0 (Initialization):
  TA:
  - Generate the sub-secret key (A_i, e_i, E) and function$\rho(x) = x - A$, initialize the global model parameters ω_{global}, and send them over a secure channel to user $i(i \in \boldsymbol{U},\ |\boldsymbol{U}| = n)$, where $\{e_i | e_i > 1 \text{and} e_i \subseteq \mathbb{Z}_E\}$, satisfies $e_1 \leq e_2 \leq \cdots \leq e_n$, $\gcd(e_i, e_j) = 1\ (i \neq j)$, $e_1 e_2 \cdots e_t > e_n e_{n-1} e_{n-t+2}$, where A satisfies $e_n e_{n-1} \cdots e_{n-t+2} < A < e_1 e_2 \cdots e_t$. $A_i = A \bmod e_i$, $E = e_1 e_2 \cdots e_n$.
  - Offline after distribution of safety parameters and initial model.

  Healthcare edge i:
  - Online user set is denoted as $\boldsymbol{U}$and$|\boldsymbol{U}| > t$, where t is the threshold value of Asmuth-Bloom's t-out-of-n threshold secret sharing protocol in our model; otherwise, the protocol is terminated.
  - Receive A from TA $\{(A_i, e_i, E), \omega_{\text{global}}\}$. Update the local model parameters to the global model parameters $\omega_{\text{local}} \leftarrow \omega_{\text{global}}$.
- Round 1 (Training):
  Healthcare edge i:
  - Check whether the online user $\boldsymbol{U}_1 \subseteq \boldsymbol{U}$ and $|\boldsymbol{U}_1| \geq t$. If not, abort and start over.
  - Training parameters $\omega_i^{j+1} = \omega_i^j - \alpha \nabla L_f(\boldsymbol{D}_i{}^j, \omega_i^j)$ on the local dataset using the stochastic gradient descent (SGD) algorithm, $\omega_i{}^j + 1$ being the model parameters for user i for iteration $j + 1$ rounds.
  - Generate a secret share of $\omega_i{}^j + 1$ denoted as $\{(k, \omega_{i,k}^{j+1})\}_{k \in \boldsymbol{U}_1} \leftarrow \text{S.share}(\omega^{j+1}, t, \boldsymbol{U}_1)$.
  - Blind the local model parameters $z^i = \omega_i^{j+1} + A_i$.
- Round 2 (Sharing):
  Healthcare edge i:
  - Check whether the online user $\boldsymbol{U}_2 \in \boldsymbol{U}_1and|\boldsymbol{U}_2| \geq t$. If not, abort and start over.
  - Generate the secret share of the local blinded model parameter $z_k{}^i$ as $\{(k, z_k^i)\}_{k \in \boldsymbol{U}_2} \leftarrow \text{S.share}(z^i, t, \boldsymbol{U}_2)$.
  - $\{\varphi^k\}_{k \in \boldsymbol{U}_2} \leftarrow$ RA.Scatter - Reduce$(\{(z_1^i, m_1, M), (z_2^i, m_2, M), \cdots, (z_k^i, m_k, M)\})$
  - $\{\varphi\} \leftarrow$ RA.Allgather$(\{\varphi^k\}_{k \in U_2})$
- Round 3 (Unmasking):
  Healthcare edge i:
  - Check whether the online user $\boldsymbol{U}_3 \in \boldsymbol{U}_2$ and $|\boldsymbol{U}_3| \geq t$. If not, abort and start over.
  - Calculate $\omega'_{\text{sum}} \leftarrow \text{S.recon}(\{k, \varphi\}_{k \subseteq \boldsymbol{U}_3})$.
  - Calculate $\omega_{\text{sum}} = \rho(\omega'_{\text{sum}})$
  - Calculate $\omega_{\text{global}}^{j+1} \leftarrow \frac{1}{\sum\limits_{k \subseteq \boldsymbol{U}_3} k} \omega_{\text{sum}}$
  - $\omega_{\text{local}} \leftarrow \omega_{\text{global}}^{j+1}$
  - If any of the above conditions are not valid, reject the aggregated result. Otherwise, accept the result and move to Round 0.

Fig. 4. Implementation process of RPDFL.

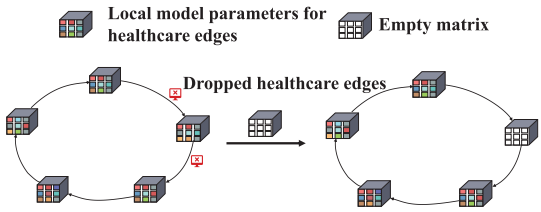


Fig. 5. The data sharing procedure of ring-Allreduce.

greater than t secret shares, ensuring that the scheme continues. As shown in Round 1 of Fig. 4, the edge applies A_i to blind the local model $\omega_i{}^{j+1}$ to generate the blind model parameters z^i. Subsequently, in Round 2 of Fig. 4, a threshold secret sharing scheme is used to partition z^i into k shares and share the secret shares using the ring-allreduce algorithm so that the number of shares acquired by the edge is more significant than t shares to solve for A. Finally, $\rho(\omega'_{\text{sum}})$ is calculated to derive the correct model aggregation parameters.

4) Model Update Phase: After the healthcare edge locally obtains all online edge model gradient values, it performs gradient averaging (e.g. FedAvg [29]) to generate a new round of global model, and the edge uses the global model to replace the local models for next-generation training.

VI. Theoretical Analysis

In this section, we present a theoretical analysis of the RPDFL data aggregation scheme in terms of correctness and confidentiality.

A. Correctness Analysis

Theorem VI.1. The protocol allows for less than t healthcare edges to drop out and the online edge can still decrypt the aggregated gradient correctly.

Proof 1. Once the aggregation is performed by the edge, the client calculates, decrypts and obtains the correct global gradient aggregation after receiving the aggregated cipher text.

$$\begin{aligned}
\omega_{\text{sum}} &= \rho(\omega_{\text{sum}}{}') \\
&= \sum_{i=1}^{t}\sum_{j=1}^{t} y_{i_j} \bmod \prod_{j=1}^{t} m_{i_j} \bmod \prod_{j=1}^{t} e_{i_j} \\
&= \sum_{i=1}^{t} z^i \bmod \prod_{j=1}^{t} e_{i_j} \\
&= (z^1 + z^2 + \cdots z^t) \bmod \prod_{j}^{t} e_{i_j} \\
&= ((s^1+A_1)+(s^2+A_2)+\cdots+(s^t+A_t)) \bmod \prod_{j}^{t} e_{i_j} \\
&= ((s^1+s^2+\cdots+s^t)+(A_1+A_2+\cdots A_t)) \bmod \prod_{j}^{t} e_{i_j} \\
&= (s^1 + s^2 + \cdots + s^t) + A \\
&= \rho(\omega_{\text{sum}}{}') = \omega_{\text{sum}}. \qquad (12)
\end{aligned}$$

B. Security Analysis

In this section, we proof that RPDFL is secure under the HbC security model. We define an HbC security model for RPDFL. In detail, the TA is trusted and does not collude with other participants, and all edges are considered HbC. Although edges follow the protocol correctly, they can collude with the remaining $t-2$ edges [14], [32] to infer information about others [46], [47]. We demonstrate the security of RPDFL by constructing an indistinguishable view of the simulator and the real protocol running. As shown below, the local gradient ω_k, $(k \in \boldsymbol{U})$ of each edge is blinded to $z^i = \omega_i^{j+1} + A_i$. We adopt $\omega_{\boldsymbol{U}'} = \{\omega_k\}_{k\subseteq \boldsymbol{U}'}$ to denote the local gradient in a subset $\boldsymbol{U}'$ of participants, where $\boldsymbol{U}' \subseteq \boldsymbol{U}$. We will present a theorem that any conspiracy of less than t edges cannot gain access to the privacy information of other edges, except as a result of global aggregation.

We know that edges may drop out at some point in the RPDFL process. We use $\boldsymbol{U}_i \subseteq \boldsymbol{U}$ to denote the total number of participants at different stages in RPDFL. Thus we have $\boldsymbol{U}_3 \subseteq \boldsymbol{U}_2 \subseteq \boldsymbol{U}_1 \subseteq \boldsymbol{U}$. Given a subset $\boldsymbol{O} \subseteq \boldsymbol{U}$ of all participants, the true attempt $\mathbf{REAL}_{\boldsymbol{O}}^{\boldsymbol{U},t,\lambda}(\omega_{\boldsymbol{U}}, \boldsymbol{U}_1, \boldsymbol{U}_2, \boldsymbol{U}_3)$ of $\boldsymbol{O}$, where t and λ denote the threshold and the security parameters used in the protocol, respectively.

Theorem VI.2. For all parameters t, A, and $\boldsymbol{O} \subseteq \boldsymbol{U}$, $\omega_{\boldsymbol{U}}$, $\boldsymbol{U}(|\boldsymbol{U}| \geq t)$ and $\boldsymbol{U}_3 \subseteq \boldsymbol{U}_2 \subseteq \boldsymbol{U}_1 \subseteq \boldsymbol{U}$ there exists a probabilistic polynomial time (PPT) simulator where the output of $\mathbf{SIM}_{\boldsymbol{O}}^{\boldsymbol{U},t,\lambda}$ and the output of $\mathbf{REAL}_{\boldsymbol{O}}^{\boldsymbol{U},t,\lambda}$ are computationally indistinguishable.

TABLE III
COMMUNICATION OVERHEAD PER ROUND

	Dropout	Masked Input	Data Sharing	Unmasking	Individual	Total
Edge	0%	3.4 (MB)	14.2 (MB)	1.7 (MB)	0.24 (KB)	19.3 (MB)
Edge	3%	3.3 (MB)	13.6 (MB)	1.6 (MB)	0.23 (KB)	18.5 (MB)
Edge	5%	3.1 (MB)	12.8 (MB)	1.5 (MB)	0.21 (KB)	17.4 (MB)
Edge	7%	2.8 (MB)	11.9 (MB)	1.3 (MB)	0.2 (KB)	16.0 (MB)

$$\begin{aligned}
&\mathbf{REAL}_{\boldsymbol{O}}^{\boldsymbol{U},t,\lambda}(\omega_{\boldsymbol{U}}, \boldsymbol{U}_1, \boldsymbol{U}_2, \boldsymbol{U}_3) \overset{c}{\equiv} \\
&\mathbf{SIM}_{\boldsymbol{O}}^{\boldsymbol{U},t,\lambda}(\omega_{\boldsymbol{O}}, \boldsymbol{U}_1, \boldsymbol{U}_2, \boldsymbol{U}_3) \qquad (13)
\end{aligned}$$

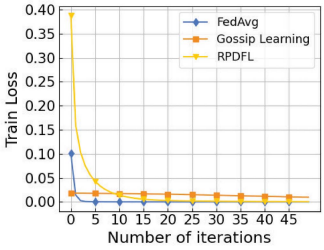
Proof. Since the cloud server is not required to be involved in RPDFL, the federated view of the parties in set $\boldsymbol{O}$ does not depend on the input of other healthcare edges who are not in $\boldsymbol{O}$. Thus, the simulator can generate perfect simulations by running the protocol with the real input of HbC edges and replacing the input of honest edges with incorrect data. We emphasise that the simulated view of the edge in $\boldsymbol{O}$ is indistinguishable from the output of the real view. During the training phase of the model, the simulator generates the error input z^i for all honest edges not in the set using random numbers, instead of using the real gradient. Since all participants are randomly added to a logical ring, this means that HbC edges cannot determine whether the computed results of the left-neighbour firing are based on the real gradients of honest edges. Therefore, the simulated view of the edges in the set $\boldsymbol{O}$ is indistinguishable from the output of the true view $\mathbf{REAL}_{\boldsymbol{O}}^{\boldsymbol{U},t,\lambda}$.

C. Communication Overhead

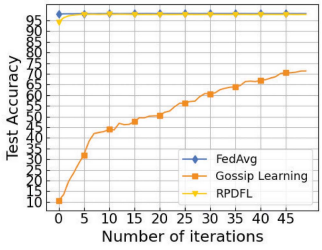
Table III illustrates the communication overhead of our RPDFL for different numbers of dropping-out edges. Specifically, we assume an integer of 4 bytes. The communication overhead in RPDFL is mainly concentrated in the data sharing phase, which does not require the participation of aggregation servers due to the decentralized federated learning setup of RPDFL. *Data Sharing* is done by neighboring nodes interacting with each other, and the *Masked Input* and *Unmasking* phases are done locally at the nodes without interaction.

VII. PERFORMANCE EVALUATION

Our experiments are implemented on the device with Intel(R) Core(TM) i5-9500 CPU @ 3.00 GHz 3.00 GHz CPU and 8 G RAM. We simulate 100 edges per round to involve in RPDFL training, and randomly select 10 edges to execute the protocol. We employ the MNIST image dataset, which is for handwriting digit classsification, dividing the instances into 60,000 images for RPDFL training and 10,000 images for testing. The data instances are averagely distributed among the edges. We adopt a fully connected network (128-64-10) as the training model and set the learning rate $\alpha = 0.06$. We use the proposed improved CRT secret sharing protocol to achieve gradient privacy preserving purpose. In addition, we employ a pseudo-random generator to generate secure random numbers.

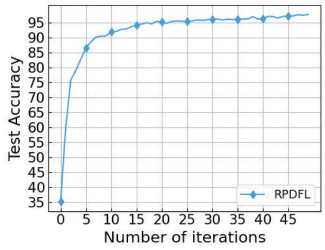


(a) Comparison of convergence speed for different approaches.

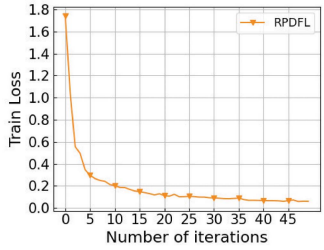


(b) Comparison of accuracy for different approaches.

Fig. 6. Comparison of different approaches.



(a) Classification accuracy in human activity recognition.



(b) Convergence speed in human activity recognition.

Fig. 7. Performance in human activity recognition.

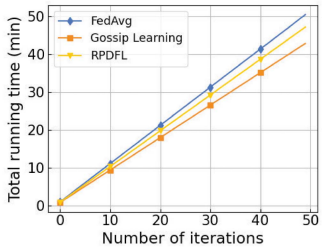
A. *Performance Analysis of RPDFL*

In this section, we analyze the performance of RPDFL by comparing it with FedAvg [29] and Gossip Learning [26]. As far as we know, Gossip Learning is the first production of decentralized learning, while both FedAvg and Gossip Learning schemes do not consider the privacy of gradients.

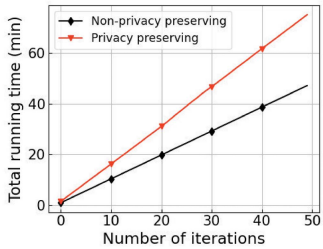
1) Classification Accuracy by Comparing With Existing Approaches: As shown in Fig. 6, we recorded the training loss and testing accuracy for RPDFL, FedAvg [29] and Gossip Learning [26]. Since edges' dropout is not supported in FedAvg and Gossip Learning during the training process, RPDFL also considers the case without edge dropout. We can see that the accuracy of our RPDFL training model is significantly better than that of Gossip Learning, but close to that of FedAvg. Furthermore, RPDFL adopts the improved CRT-based threshold secret sharing protocol, which protects the gradient privacy. The protocol proceeds with a truncation treatment, which causes a loss of accuracy and a reduction of convergence speed. For example, RPDFL is slightly slower than FedAvg in terms of convergence speed, but the accuracy reaches 98 % at round 5 and is comparable to FedAvg thereafter. Thus, RPDFL can hold fantastic efficiency while ensuring that the gradient privacy is not leaked. We are fortunate that RPDFL prevents malicious users from inferring others' privacy in real-world scenarios and achieves the purpose of privacy-preserving federated learning.

2) Classification Accuracy on Healthcare Data Set: We validate the performance of RPDFL on a dataset from UCI called UCI-HAR [48] that captures daily human activities through smartphones. The dataset includes 561 attributes and 10299 instances for classification and clustering tasks. It is possible to divide the dataset into two random groups, where 70% of the data is the training set and 30% of the test set. Likewise, we validate the RPDFL training protocol on the UCI-HAR dataset adopting a fully connected network (128-64-10). As shown in Fig. 7, RPDFL equally maintains a excellent performance. It allows RPDFL to address the challenges of information barriers between healthcare organizations and to make highly efficient decisions about complicated matters.

3) Running Time by Comparing With Existing Approaches: We know that security is not considered in FedAvg and Gossip Learning, while RPDFL uses the ring-AllReduce algorithm combined with ring topology to design a secure data sharing scheme. In order to fair comparison the running time of RPDFL



(a) Edges = 100, runtime of different approaches.



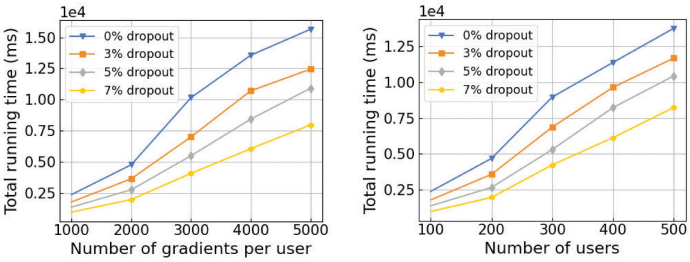
(b) Gradients = 109194, runtime of different approaches.

Fig. 8. Total running time.

with the FedAvg and Gossip Learning, we do not consider security in this section. From Fig. 8(a), it is observed that FedAvg forms a communication bottleneck in the face of a large number of nodes communicating with the central aggregation server and has more running time than the two decentralized methods. First, Gossip Learning shares data by selecting surrounding nodes to broadcast data, and the nodes that have received the message will repeat the process until all nodes have collected the data. In contrast, RPDFL specifies that only neighbor edge can communicate with each other, so the communication time of RPDFL is slightly more than that of Gossip Learning. In addition, Fig. 8(b) shows the comparisons of the total running time of RPDFL from both privacy protected and unprotected gradients. The running time is within the tolerable range in order to achieve privacy preserving purposes in a realistic environment with significant privacy leakage.

4) Running Time for Different Dropped-Out Edges: RPDFL is a fully decentralized FL scheme, with gradient aggregation operations done by edges sharing data securely with each other. Fig. 9 shows the running time of the whole protocol with different number of dropped edges. We can see that the communication overhead of RPDFL increases linearly as the number of edges or gradients grows.

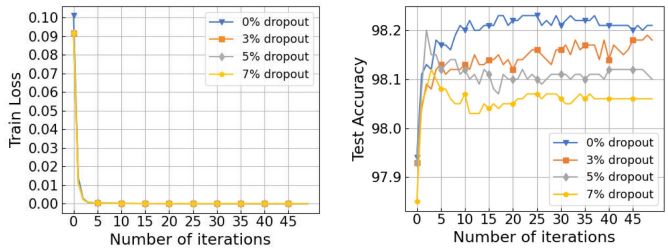
Table IV indicate the computational cost for each round. Since without the participation of the aggregation server, *Masked Input* and *Unmasking* operations are performed locally on the edge, *Data Sharing* follows the transmission gradient of the RPDFL aggregation protocol, and it requires communication between neighbor edges, which leads to significant computational cost and communication overhead.



(a) Edges = 100, with different number of gradients per edge.

(b) Gradients = 1000, with the different number of edges.

Fig. 9. Total runtime with different number of dropped edges.



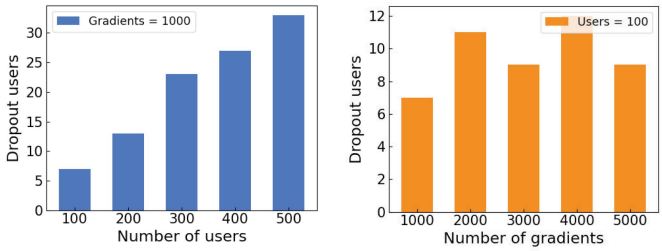
(a) Convergence speed for different number of dropped edges.

(b) Accuracy for different number of dropped edges.

Fig. 11. Comparison of accuracy and convergence speed for different number of dropped edges.

TABLE IV
COMPARISON OF FUNCTIONALITY (MS)

	Dropout	Masked Input	Data Sharing	Unmasking	Total
Edge	0%	26	100710	7395	108131
Edge	3%	21	104001	9563	113585
Edge	5%	17	120066	10969	131052
Edge	7%	13	152199	11595	163807



(a) Gradients = 1000, with the different number of edges.

(b) Edges = 100, with different number of gradients per edge.

Fig. 10. Dropout edges.

B. Robustness of Edges Dropout

Our RPDFL supports edge dropping out during protocol performance, which is widespread due to device battery problems, communication signal trouble, and hardware quality issues. To evaluate the prevalence of this phenomenon, we record the number of edges that drop out of the entire system at different number of edges gradients. Specifically, each node records whether a message is received from its left neighbor edge within a specified time. As shown in Fig. 10, we found that a certain number of edges' dropout are inevitable as the number of edges and gradients increases, and they become more pronounced as the number of edges increases. However, Fig. 9 shows that the percentage of dropped edges is not significant compared to the total number of edges. Thus, it provides practical support for the possibility of using a secret sharing protocol to handle the edges' dropout problem.

Fig. 11 shows the convergence speed and accuracy of the model with different percentage of dropped edges. Obviously, the increase in the number of dropped edges has some effect on the prediction of the model, but the convergence speed of the model remains constant with the increase in dropped edges. A major factor is that RPDFL chooses to aggregate the gradients of all online edges. Therefore, in order to ensure the robustness of RPDFL against user disconnection, a threshold secret sharing scheme was adopted to divide the gradient into n parts in the gradient aggregation phase, and only t parts could be satisfied to complete the gradient aggregation. Supporting $n - t$ edges dropout would not affect RPDFL. In contrast, other schemes do not complete aggregation until all edges' gradients are received, so the scheme does not support edge dropout. For example, the difference in prediction accuracy between 7 edges dropping out of 100 and no edges dropping out is about 0.1%. Because of our RPDFL's dropout resilience, we can effectively guarantee the correct execution of the scheme in practical situations when users drop out consciously or unconsciously, without revealing the user's privacy.

VIII. CONCLUSION

In this paper, we proposed a robust and privacy-preserving decentralized deep federated learning (RPDFL) training scheme for digital healthcare applications, aiming to train parameters and models to protect data privacy for patients and healthcare organizations. For this, we designed a novel ring FL structure and a Ring-Allreduce-based data sharing scheme that improves the communication efficiency in RPDFL training schemes and overcomes the poor scalability and single-point-of-failure problems of centralized FL. Furthermore, we implemented a novel data sharing scheme by updating execution processes of CRT in the threshold secret sharing scheme, supporting healthcare edges to drop out of the scheme during training without causing data leakage, and ensuring the robustness of the RPDFL training. Additionally, RPDFL supports edges' dropout during training process while preserving edge local gradient privacy. Security analysis shows that our RPDFL is highly secure under the HbC security model. Moreover, the experiments results also verify the excellent performance of RPDFL scheme. As part of future research work, we will strive to increase parallelism to reduce the computational cost and communication overhead of the entire RPDFL training scheme.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive comments.

REFERENCES

[1] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "FedHealth: A federated transfer learning framework for wearable healthcare," *IEEE Intell. Syst.*, vol. 35, no. 4, pp. 83–93, Jul./Aug. 2020.

[2] 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. 2021.

[3] D. Y. Zhang, Z. Kou, and D. Wang, "FedSens: A federated learning approach for smart health sensing with class imbalance in resource constrained edge computing," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.

[4] X. Zhou, W. Liang, J. Ma, Z. Yan, I. Kevin, and K. Wang, "2D federated learning for personalized human activity recognition in cyber-physical-social systems," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 6, pp. 3934–3944, Nov./Dec. 2022.

[5] W. Zhang et al., "Blockchain-based federated learning for device failure detection in industrial IoT," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5926–5937, Apr. 2021.

[6] X. Zhou, X. Xu, W. Liang, Z. Zeng, and Z. Yan, "Deep-learning-enhanced multitarget detection for end–edge–cloud surveillance in smart IoT," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12 588–12 596, Aug. 2021.

[7] C. Luo, J. Ji, Q. Wang, X. Chen, and P. Li, "Channel state information prediction for 5G wireless communications: A deep learning approach," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 227–236, Jan.-Mar. 2020.

[8] J. Xiong, R. Bi, Y. Tian, X. Liu, and D. Wu, "Toward lightweight, privacy-preserving cooperative object classification for connected autonomous vehicles," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2787–2801, Feb. 2022.

[9] Z. Zhou, Y. Tian, and C. Peng, "Privacy-preserving federated learning framework with general aggregation and multiparty entity matching," *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–14, 2021.

[10] X. Zhou, X. Yang, J. Ma, I. Kevin, and K. Wang, "Energy-efficient smart routing based on link correlation mining for wireless edge computing in IoT," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14 988–14 997, Aug. 2022.

[11] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.

[12] Y. Pan, J. Ni, and Z. Su, "FL-PATE: Differentially private federated learning with knowledge transfer," in *Proc. IEEE Glob. Commun. Conf.*, 2021, pp. 1–6.

[13] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10 700–10 714, Dec. 2019.

[14] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "VerifyNet: Secure and verifiable federated learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 911–926, 2019.

[15] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, *arXiv:1610.05492*.

[16] Y. Tian, T. Li, J. Xiong, M. Z. A. Bhuiyan, J. Ma, and C. Peng, "A blockchain-based machine learning framework for edge services in IIoT," *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 1918–1929, Mar. 2022.

[17] J. Kang et al., "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4660–4670, Jun. 2019.

[18] L. Yang, Y. Lu, J. Cao, J. Huang, and M. Zhang, "E-tree learning: A novel decentralized model learning framework for edge AI," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11 290–11 304, Jul. 2021.

[19] Y. Aono et al., "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 5, pp. 1333–1345, May 2018.

[20] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, "Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles," *IEEE Wireless Commun.*, vol. 27, no. 3, pp. 24–30, Jun. 2020.

[21] Z. Peng et al., "VFChain: Enabling verifiable and auditable federated learning via blockchain systems," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 1, pp. 173–186, Jan./Feb. 2022.

[22] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1279–1283, Jun. 2020.

[23] G. Aloi, O. Briante, M. Di Felice, G. Ruggeri, and S. Savazzi, "The SENSE-ME platform: Infrastructure-less smartphone connectivity and decentralized sensing for emergency management," *Pervasive Mobile Comput.*, vol. 42, pp. 187–208, 2017.

[24] S. Kianoush, M. Raja, S. Savazzi, and S. Sigg, "A cloud-IoT platform for passive radio sensing: Challenges and application case studies," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3624–3636, Oct. 2018.

[25] J. Ni, K. Zhang, and A. V. Vasilakos, "Security and privacy for mobile edge caching: Challenges and solutions," *IEEE Wireless Commun.*, vol. 28, no. 3, pp. 77–83, Jun. 2021.

[26] I. Hegedűs, G. Danner, and M. Jelasity, "Gossip learning as a decentralized alternative to federated learning," in *Proc. IFIP Int. Conf. Distrib. Appl. Interoperable Syst.*, 2019, pp. 74–90.

[27] S. Savazzi, M. Nicoli, and V. Rampa, "Federated learning with cooperating devices: A consensus approach for massive IoT networks," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4641–4654, May 2020.

[28] Z. Wang, Y. Hu, S. Yan, Z. Wang, R. Hou, and C. Wu, "Efficient ring-topology decentralized federated learning with deep generative models for medical data in ehealthcare systems," *Electron.*, vol. 11, no. 10, 2022, Art. no. 1548.

[29] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[30] H. Zhu, R. Wang, Y. Jin, K. Liang, and J. Ning, "Distributed additive encryption and quantization for privacy preserving federated deep learning," *Neurocomputing*, vol. 463, pp. 309–327, 2021.

[31] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, and W. Luo, "Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 5, pp. 2438–2455, Sep./Oct. 2021.

[32] K. Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 1175–1191.

[33] J. H. Bell, K. A. Bonawitz, A. Gascón, T. Lepoint, and M. Raykova, "Secure single-server aggregation with (poly) logarithmic overhead," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2020, pp. 1253–1269.

[34] X. Zhou, W. Liang, J. She, Z. Yan, I. Kevin, and K. Wang, "Two-layer federated learning with heterogeneous model aggregation for 6G supported internet of vehicles," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5308–5317, Jun. 2021.

[35] K. Mandal and G. Gong, "PrivFL: Practical privacy-preserving federated regressions on high-dimensional data over mobile networks," in *Proc. ACM SIGSAC Conf. Cloud Comput. Secur. Workshop*, 2019, pp. 57–68.

[36] W. Wu, L. He, W. Lin, and R. Mao, "Accelerating federated learning over reliability-agnostic clients in mobile edge computing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 7, pp. 1539–1551, Jul. 2021.

[37] Z. Liu, J. Guo, K.-Y. Lam, and J. Zhao, "Efficient dropout-resilient aggregation for privacy-preserving machine learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 1839–1854, 2022.

[38] F. Yin et al., "FedLoc: Federated learning framework for data-driven cooperative localization and location data processing," *IEEE Open J. Signal Process.*, vol. 1, pp. 187–215, 2020.

[39] W. Y. B. Lim et al., "Dynamic contract design for federated learning in smart healthcare applications," *IEEE Internet Things J.*, vol. 8, no. 23, pp. 16 853–16 862, Dec. 2021.

[40] Y. Zhang, L. Song, W. Saad, Z. Dawy, and Z. Han, "Contract-based incentive mechanisms for device-to-device communications in cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 10, pp. 2144–2155, Oct. 2015.

[41] K. Bonawitz et al., "Towards federated learning at scale: System design," *Proc. Mach. Learn. Syst.*, vol. 1, pp. 374–388, 2019.

[42] P. Kairouz et al., "Advances and open problems in federated learning," *Foundations Trends Mach. Learn.*, vol. 14, no. 1–2, pp. 1–210, 2021.

[43] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. 19th Int. Conf. Comput. Statist.*, Paris France, 2010, pp. 177–186.

[44] C. Asmuth and J. Bloom, "A modular approach to key safeguarding," *IEEE Trans. Inf. theory*, vol. 29, no. 2, pp. 208–210, Mar. 1983.

[45] Y. Li, C. Chen, N. Liu, H. Huang, Z. Zheng, and Q. Yan, "A blockchain-based decentralized federated learning framework with committee consensus," *IEEE Netw.*, vol. 35, no. 1, pp. 234–241, Jan./Feb. 2021.

[46] Y. Liu et al., "Trojaning attack on neural networks," in *Proc. ISOC Netw. Distrib. Syst. Secur. Symp.*, 2018, pp. 1–15.

[47] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the GAN: Information leakage from collaborative deep learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 603–618.

[48] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *Proc. Int. Workshop Ambient Assist. Living*, 2012, pp. 216–223.

Youliang Tian (Member, IEEE) received the PhD degree in cryptography from Xidian University, Xi'an, China, in 2012. He is the full professor and Ph.D. supervisor with the State Key Laboratory of Public Big Data and the College of Computer Science and Technology, Guizhou University. He has authored more than 100 publications and 2 books. His current research interests include algorithmic game theory, cryptography and security protocols, privacy protection, and blockchain etc.

Shuai Wang received the BSc degree in software engineering from Fujian Normal University, Fuzhou, China, in 2020. He is currently working toward the PhD degree with the College of Computer Science and Technology, Guizhou University. His research interests include deep federated learning and privacy protection.

Jinbo Xiong (Senior Member, IEEE) received the PhD degree in computer system architecture from Xidian University, China, in 2013. He is currently the full professor and PhD supervisor with the Fujian Provincial Key Laboratory of Network Security and Cryptology, and College of Computer and Cyber Security, Fujian Normal University. He has authored more than 100 publications, 3 books and authorized 7 invention patents. His research interests include secure deep learning, data security, privacy protection, and Internet of Things.

Renwan Bi received the MS degree in software engineering from the College of Mathematics and Informatics, Fujian Normal University, Fuzhou, China, in 2021. He is currently working toward the PhD degree in cyberspace security with the College of Computer and Cyber Security, Fujian Normal University. His research interests include secure multiparty computation, connected, and autonomous vehicle.

Zhou Zhou received the MS degree in computer software and theory from Guizhou University, Guiyang, China, in 2015. She is currently working toward the PhD degree with the College of Computer Science and Technology, Guizhou University. Her research interests include federated learning and privacy protection.

Md Zakirul Alam Bhuiyan (Senior Member, IEEE) received the MEng and PhD degrees in computer science and technology from Central South University, China, in 2009 and 2013. He is currently an assistant professor with the Department of Computer and Information Sciences, Fordham University, NY, USA, the Founding director of Fordham Dependable and Secure System Lab (DependSys). His research interests include dependability, cybersecurity, Big Data, and IoT/CPS applications.

Secure and Privacy-Preserving Decentralized Federated Learning for Personalized Recommendations in Consumer Electronics Using Blockchain and Homomorphic Encryption

Brij B. Gupta, *Senior Member, IEEE*, Akshat Gaurav, *Graduate Student Member, IEEE*, and Varsha Arya

***Abstract*—Over the past few years, personalized recommendations have emerged as a fundamental component of the consumer electronics sector. The rise of decentralized federated learning has expanded the horizons of personalized recommendations, offering significant potential. Nonetheless, the utilization of confidential data from diverse clients raises legitimate concerns regarding privacy and security. In response to these challenges, we present an innovative framework for secure and privacy-preserving decentralized federated learning, tailored to personalized recommendations within the consumer electronics sector. Our approach strives to facilitate the collective contribution of data from multiple clients to the learning process while safeguarding their privacy. To accomplish this, we harness the power of homomorphic encryption, ensuring that clients' data remains encrypted and impervious to prying eyes. Additionally, we leverage blockchain technology to establish a secure, decentralized foundation for data exchange and management. Through the utilization of blockchain, we empower clients to validate the integrity of the learning process, guarantee system transparency, and thwart any malicious attempts at result manipulation. Our framework is rigorously assessed using real-world consumer electronics data, highlighting its capacity to provide a secure, decentralized, and privacy-centric solution for personalized recommendations. This approach not only enriches the user experience but also offers robust safeguards for sensitive data.**

***Index Terms*—Personalized recommendations, federated learning, decentralized learning, blockchain, homomorphic encryption, privacy preservation, security, consumer electronics, data sharing.**

Manuscript received 30 April 2023; revised 28 August 2023; accepted 27 October 2023. Date of publication 8 November 2023; date of current version 26 April 2024. This work was supported by the National Science and Technology Council (NSTC), Taiwan, under Grant NSTC112-2221-E-468-008-MY3. *(Corresponding author: Akshat Gaurav; Brij B. Gupta.)*

Brij B. Gupta is with the Department of Computer Science and Information Engineering, Asia University, Taichung 413, Taiwan, also with Kyung Hee University, Seoul 02447, South Korea, also with the Symbiosis Centre for Information Technology, Symbiosis International University, Pune 412115, India, and also with the Department of Electrical and Computer Engineering, Lebanese American University, Beirut 1102, Lebanon (e-mail: bbgupta@asia.edu.tw).

Akshat Gaurav is with the Computer Science Department, Ronin Institute, Montclair, NJ 07043 USA (e-mail: akshat.gaurav@ronininstitute.org).

Varsha Arya is with the Department of Business Administration, Asia University, Taichung 413, Taiwan, also with the Center for Interdisciplinary Research, University of Petroleum and Energy Studies, Dehradun 248007, India, and also with the University Centre for Research & Development (UCRD), Chandigarh University, Chandigarh 140413, India (e-mail: 111231027@live.asia.edu.tw).

Digital Object Identifier 10.1109/TCE.2023.3329480

I. Introduction

FEDERATED learning (FL) is a kind of machine learning in which several clients train a single model together, with the help of a coordinating server, but with the data for the model stored in separate locations [1], [2]. Many of the systemic privacy concerns and costs caused by conventional, centralised data science and machine learning techniques may be avoided with FL since it embraces the concepts of focussed data acquisition and reduction. Researchers from a variety of fields discussed the one-of-a-kind features and difficulties of FL, surveyed the landscape of existing methods, and pointed out numerous promising avenues for further exploration [3], [4], [5]. Bonawitz et al. [6] detail a TensorFlow-based, flexible production system for FL in the mobile device arena. Li et al. [7] analyses the difficulties of training in diverse and possibly enormous networks, and suggests numerous avenues for future research, while Kairouz et al. [8] discuss these developments and give a comprehensive collection of open questions and challenges.

According to the research community as a whole, FL is a machine learning approach that, unlike conventional learning, trains an algorithm without moving data samples across a large number of distributed edge devices or servers. Data privacy, data security, data access rights, and access to heterogeneous data may all be handled via federated learning, which enables several actors to work together on the construction of a single, strong machine learning model without sharing data [9], [10], [11], [12], [13], [14]. FL has rapidly gained popularity in both academia and industry in recent years due to its ability to provide privacy protection to Internet users [15], [16]. Federated learning has applications in various fields, including the medical system, where it can be used to build an intelligent system that assists medical staff without sharing patient data [17]. Finally, Fan et al. [18], [19] proposes a new algorithm for federated learning based on a dual perspective. Theoretically, it achieves better convergence rates than the state-of-the-art primal federated optimization algorithms under certain situations.

Federated learning is a machine learning approach that enables numerous devices to jointly train a common model without disclosing their individual data to one another [20], [21], [22]. Since protecting users' personal information

is a top priority, this method is ideal for building consumer electronics with built-in recommendation systems [23], [24], [25], [26]. In the past, recommendation systems have been built using centralised machine learning models, which gather and store user data in one centralised location. However, this method exposes sensitive user data to possible breaches, cyber attacks, or misuse, which poses significant privacy and security risks. Recommendation systems based on federated learning, on the other hand, may learn independently in a distributed fashion without compromising individual users' anonymity. In this method, the user's data stays put on the device and only the model's aggregated parameters are sent to the cloud service. This prevents sensitive information from leaving the device and into the hands of an unauthorised third party. The user's preferences and interests are captured during the training of the model on the local device, therefore federated learning-based recommendation systems may also result in more tailored suggestions. The training data is less biassed using this method since it is not restricted to what is stored on a single server. In conclusion, creating individualised recommendation systems in consumer electronic devices using federated learning-based recommendation systems is a safe, private, and effective method. In this context, we proposed a secured decentralized, federated learning-based recommendation system for consumer electronics using blockchain and homomorphic encryption. Our contribution are as follows:

- Our proposed approach is based on blockchain hence it makes the recommendation system decentralized.
- We used homomorphic encryption to transfer the features between the local users and the central server, making it secure from data leakage attacks.

II. Literature Review

Recently, research suggests that federated learning and deep learning [25], [27], [28] can be used to improve the performance of many systems, such as recommender systems, cyber-physical system security [29], text analysis [30], computer vision [22], healthcare [31] and cyber security [32]. Jalalirad et al. [33] proposes a simple and efficient extension of federated learning for recommender systems that improves personalization. Chen et al. [34] presents a federated meta-learning framework for recommendation that shares user information at the level of algorithm, which preserves user privacy and utilizes information from other users to help model training. Muhammad et al. [35] presents a novel technique, FedFast, to accelerate distributed learning that achieves good accuracy for all users very early in the training process. Sun et al. [36] proposes an ensemble federated edge learning scheme (eFEEL) that aims to efficiently and effectively improve recommender systems without breaching user data privacy. Salloum and Tekli [37], [38] proposed automated and personalized nutrition health assessment, recommendation, and progress evaluation using fuzzy reasoning. Li et al. [39] proposed a scholarly recommendation based on high-order propagation of knowledge graphs. Xiao et al. [40] proposed a recommendation system for Healthcare Services. These papers suggest that federated learning can improve the performance of recommender systems while preserving user privacy and reducing communication costs.

There are several data quality issues associated with FL. Liu et al. [41] highlights the importance of improving the quality of FL models, while Pejó [42] shows that models trained with FL can potentially leak information about the quality of the datasets used. Liu et al. [41] proposes a solution to select high-quality data samples for FL tasks in a privacy-preserving way, while Verma et al. [43] proposes approaches to address the data skew problem in FL. George and Lal [44] proposed a recommendation system for the education system. Overall, the papers suggest that data quality is a crucial factor in the success of FL, and that more research is needed to address the various data quality issues associated with this technique.

However, FL is vulnerable to various security and privacy threats, including communication bottlenecks, poisoning, backdoor attacks, and inference-based attacks. Authors [45], [46], [47], provide comprehensive surveys of the unique security vulnerabilities exposed by the FL ecosystem, highlighting the vulnerabilities sources, key attacks on FL, defenses, as well as their unique challenges. Nguyen and Thai [48] proposes a framework that offers both privacy guarantees for users and detection against poisoning attacks from them. Overall, the papers suggest that more research is needed to address FL's security and privacy concerns to enable its mass adoption.

III. Preliminaries

A. Encryption

Homomorphic encryption (HE) is a method of cryptography that eliminates the requirement to decode data before doing calculations on it. In other words, it allows the manipulation of encrypted data without exposing it to anyone, including the person performing the computations. This property is particularly useful in scenarios where sensitive data needs to be processed and analyzed without compromising the privacy of the data owners. Using a mathematical function, homomorphic encryption successfully transforms plaintext information into encrypted information. In this way, the plaintext information is protected while the ciphertext information is utilised in calculations. The computation's output will likewise be encrypted, and only the private key's owner will be able to decode it.

Homomorphic encryption comes in a few different flavours, the most common of which are completely homomorphic encryption (FHE), partly homomorphic encryption (PHE), and slightly homomorphic encryption (SHE). Since FHE allows for any computation to be performed on encrypted data, it is the most powerful form of homomorphic encryption. Unfortunately, it also has the highest computational cost. PHE and SHE are more efficient but less potent, making them common in real-world applications.

Homomorphic encryption may be used in a number of contexts, such as private data analysis, cloud security, and AI privacy. Client data privacy may be preserved by the use of homomorphic encryption in the context of blockchain-based

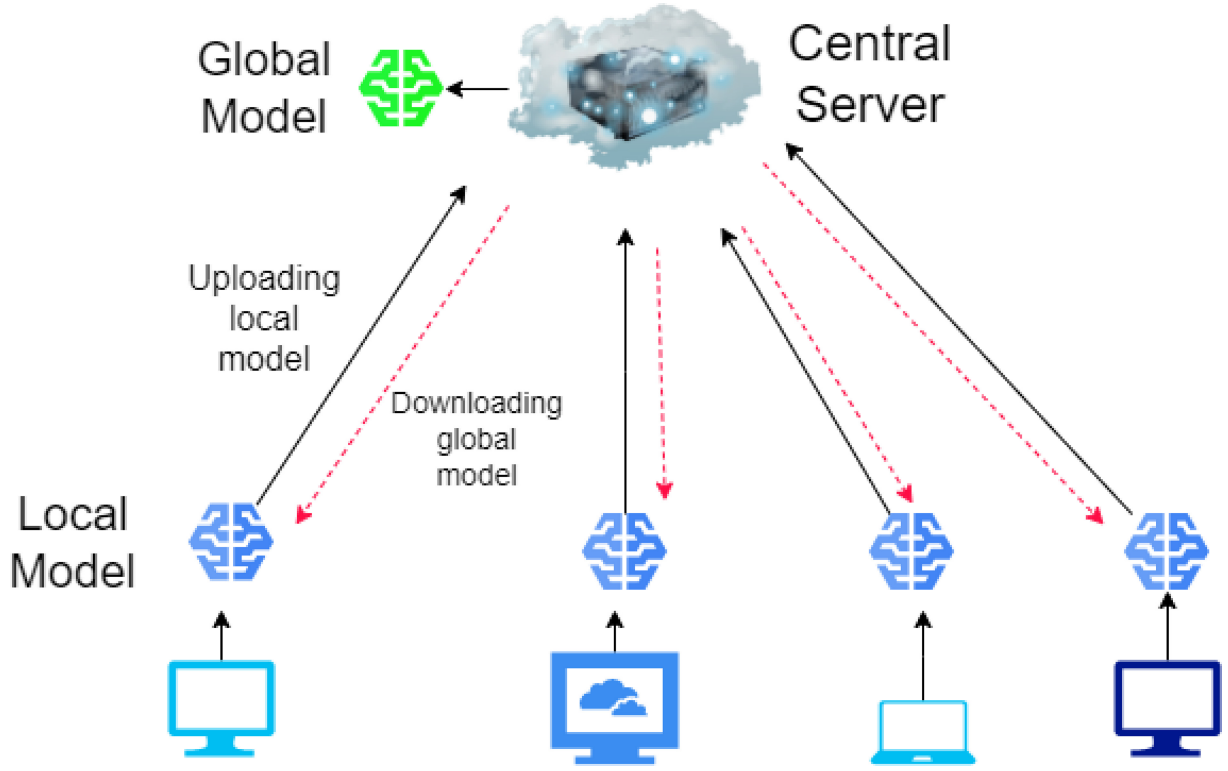


Fig. 1. Federated Learning Architecture.

decentralised federated learning for personalised recommendations in consumer electronics.

Let $E(m)$ be an encryption function that encrypts a plaintext message m, and let $+$ be a binary operation on ciphertexts such that $E(x+y) = E(x)+E(y)$ for all ciphertexts a and b. Then, homomorphic encryption can be expressed as:

$$E(z_1)+E(z_2)=E(z_1+z_2) \tag{1}$$

This equation shows that if we encrypt two plaintext messages z_1 and z_2, and then perform the addition operation on their ciphertexts, we will obtain the ciphertext of the sum of the plaintext messages, without revealing any information about the plaintext messages themselves.

1) Example for Homomorphic Encryption: Suppose we have a plaintext message m, and we want to encrypt it using homomorphic encryption. We can use the Paillier cryptosystem, which is an example of partially homomorphic encryption. The Paillier cryptosystem has two main components: a public key and a private key.

To encrypt m, we first generate a public key (n, g) and a private key λ. We can choose random values for a and b, two large primes, and set $n = ab$. We can also set $g = n+1$, which is a generator of the multiplicative group modulo n^2. Finally, we can compute $\lambda = \text{lcm}(a-1, b-1)$, the least common multiple of $a-1$ and $b-1$.

To encrypt z, we can choose a random value R such that $0 < R < n$ and $\gcd(R, n) = 1$. We can then compute the ciphertext c as follows:

$$c = g^z.R^n mod n^2 \tag{2}$$

To decrypt c, we can use the private key λ. We first compute $L = (g^\lambda \bmod n^2)-1$ and $u = c^\lambda \bmod n^2$. We can then compute the plaintext message m as follows:

$$m = \frac{u-1}{n}.L^{-1} mod n. \tag{3}$$

B. Blockchain Basics

Blockchain is a distributed ledger system that ensures all transactions are recorded safely and transparently.

- *Hash Function:* A hash function, also known as a hash value or hash digest, is a mathematical function that accepts data of any size as input and returns a result of a predetermined size. Hash functions are employed in blockchain to generate transaction and block IDs. The following equation represents a hash function:

$$h(x) = y \tag{4}$$

  where x is the input data, h is the hash function, and y is the resulting hash value.
- *Merkle Tree:* Blockchain technology makes use of Merkle trees, a kind of data structure, to quickly and accurately validate the correctness of massive datasets. The root hash is produced by recursively hashing all possible pairwise combinations of transactions. The following equation represents a Merkle tree:

$$M_n = H\big(H(M_{n-1,0})||H(M_{n-1,1})\big) \tag{5}$$

  where M_n is the Merkle root of the n-th level of the tree, H is the hash function, and $M_{n-1,0}$ and $M_{n-1,1}$ are the hashes of the two child nodes of the $n-1$-th level.

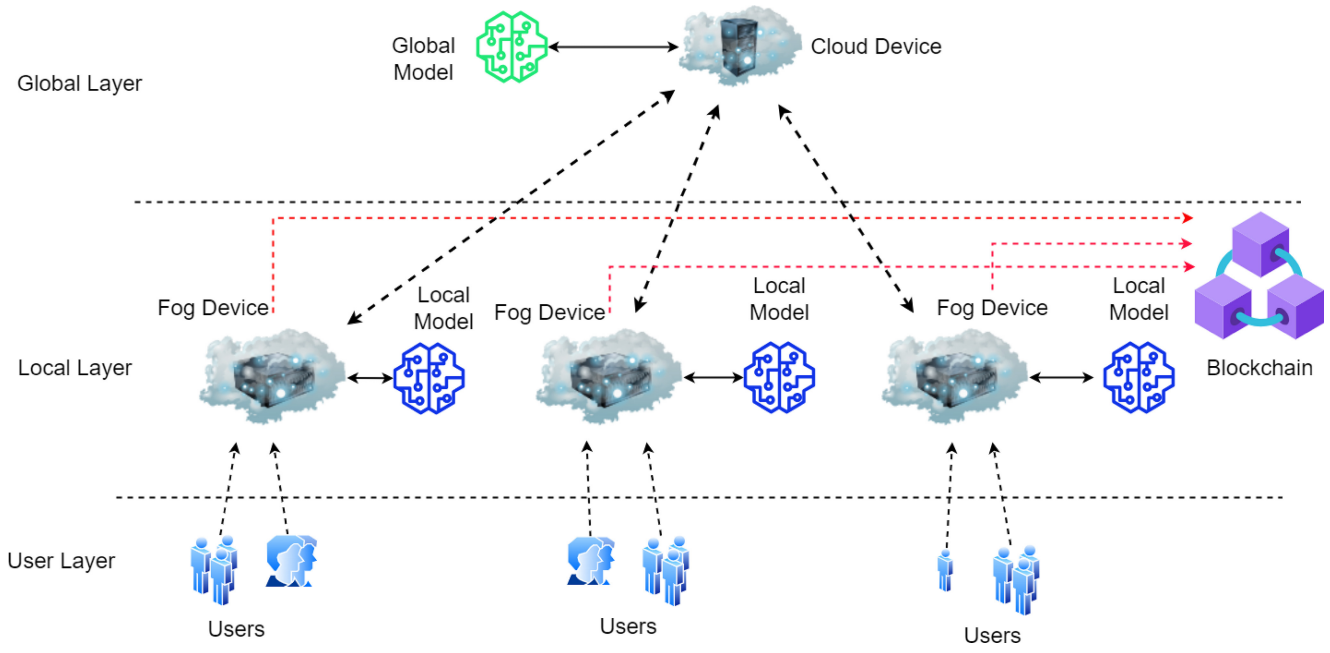


Fig. 2. System Model.

– *Proof of Work:* Blockchain transactions and new blocks are verified and added to the chain using the Proof of Work consensus process. To generate a new block, one must first solve a computationally challenging problem known as a "nonce." The following equation represents the Proof of Work algorithm:

$$H(B||N) < T \tag{6}$$

where H is the hash function, B is the block header, N is the nonce, and T is the target value that determines the difficulty of the puzzle. The algorithm requires finding a nonce that, when combined with the block header, produces a hash value that is less than the target value.

IV. Proposed Work

A. System Model

In our proposed system model, we aim to develop a secure and privacy-preserving decentralized federated learning framework for Personalized Recommendations in Consumer Electronics. This system model is designed to leverage blockchain technology and homomorphic encryption to ensure data privacy, integrity, and security throughout the entire learning process.

– *User Layer:* The User Layer is the first layer of our system model, consisting of users and their devices. Users participate in the federated learning process by sharing their data with the upper layers. To ensure privacy, users' data is first encrypted using homomorphic encryption techniques, allowing computations on encrypted data without the need for decryption. This way, users can contribute their data without revealing sensitive information directly. The process in the User Layer can be represented by the following equation for local federated learning:

$$w_u^{t+1} = Local\ Update\ \left(w_u^t, D_u\right) \tag{7}$$

where 'w' represents the u^{th} user's parameter at the t^{th} iteration and 'D' represents the encrypted data.

– *Local Layer:* The Local Layer is the intermediary between the User Layer and the Global Layer. It comprises fog devices that store local models. The local models are updated using the data contributed by the users in a privacy-preserving manner. These local models are then synchronized with the Global Layer to update the global model. The Local Layer is also responsible for user registration and authentication. The Local Layer's federated learning update process can be represented by the following equation:

$$w_u^{t+1} = \mathcal{L}\left(w_u^{t+1}\right) \forall u \in \{u_1, u_2, \ldots, u_n\} \tag{8}$$

where 'w' represents the u^{th} user's parameter at the t^{th} iteration and $\mathcal{L}$ function aggregates the local models' updates from all users to produce a refined local model for the Local Layer.

– *Global Layer:* The Global Layer consists of cloud devices that store the global model. The global model is an aggregation of the refined local models received from the Local Layer. This aggregation is done securely using homomorphic encryption and blockchain technology. The global model represents the collective knowledge learned from all users' data without directly exposing individual data. The federated learning update process in the Global Layer can be represented as follows:

$$w_g^{t+1} = \mathcal{G}\left(w_l^{t+1}\right) \forall l \in \{l_1, l_2, \ldots, l_m\} \tag{9}$$

where 'w' represents the l^{th} user's parameter at the t^{th} iteration and $\mathcal{G}$ function aggregates the refined local models from all Local Layers to produce an updated global model.

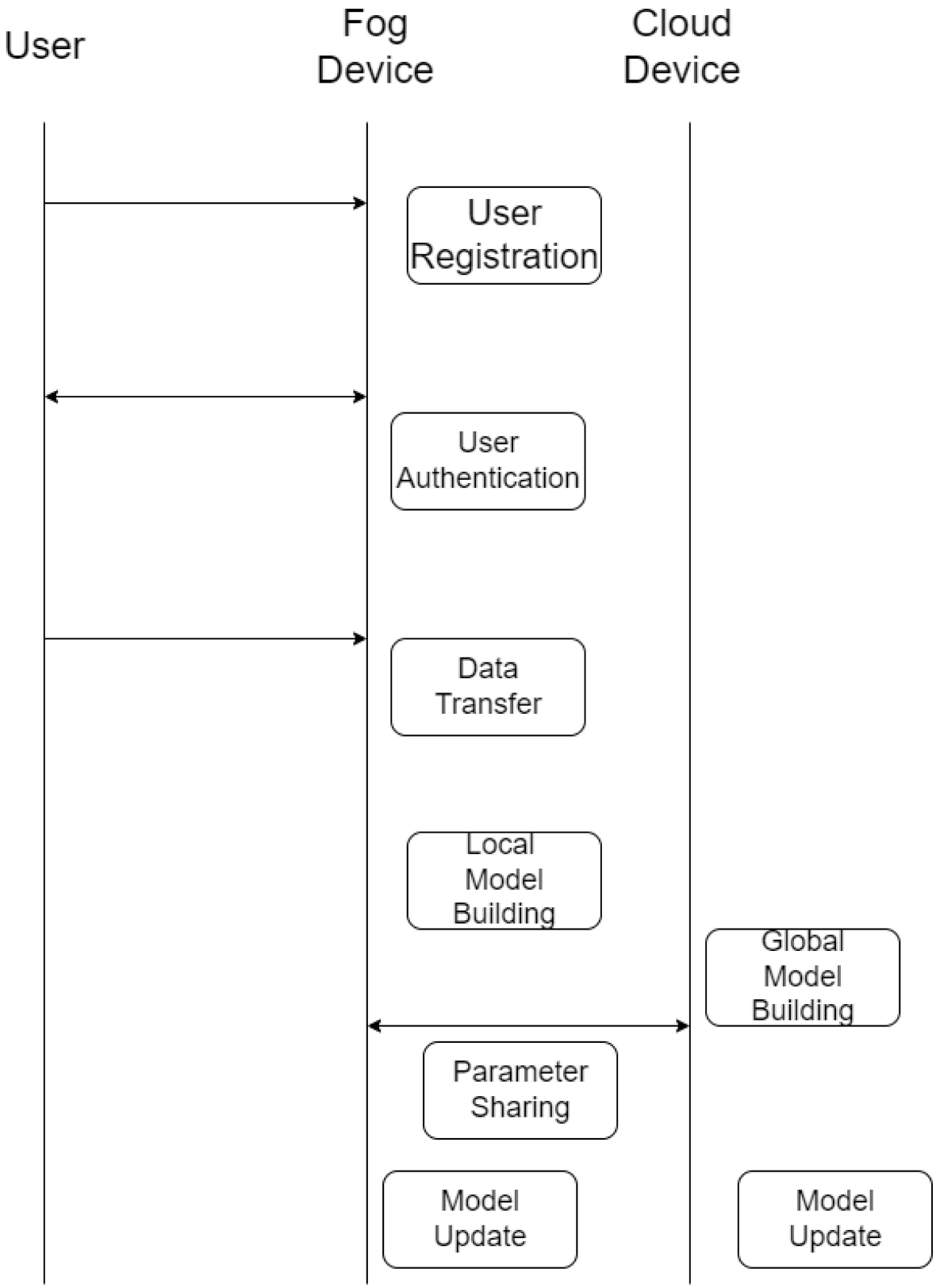


Fig. 3. State Diagram.

B. System Interactions

In this section, we will explain the working of our proposed approach. The state diagram of our proposed approach is represented in Figure 3.

– *User Registration Stage:* This is the initial stage of our proposed approach; in this stage, the fog node selects the security (S) parameter and generates the master secret (M_S) and public parameters (P_r). The fog device stores all the user information in the blockchain.

– *User Authentication Stage:* This is the second stage of our proposed approach. in this stage, the users are authenticated by the fog node, and data transmission starts.
– *Local Model Building:* In this stage, the fog device built the local model from the user data.
– *Global Model Building:* In this stage, the parameters of the local model are shared with the cloud device, and the global model at the cloud device is updated.
– *Model Update:* This is the final stage of our proposed approach; at this stage, the local and global models are updated.

C. First Phase

We proposed a lightweight three-factor authentication scheme for WSNs. Our proposed approach has the following phases:

1) Broadcasting Phase: The fog node selects a hash function and master keys, such as

$$h \rightarrow \{0,1\}^* \oplus \{0,1\}^{128}$$
$$Ka_1, Ka_2 \Rightarrow Master\ keys$$

The fog node broadcasts this hash function.

2) User Registration Phase:
– ID, password, and bio-identity are used for registration
– Ui calculates

$$GEN(Bio_i) \leftarrow (b_i, pair_i)$$
$$\widehat{PW}_i \leftarrow h(PW_i||b_i)$$
$$\widehat{ID}_i \leftarrow h(ID_i)$$
$$User \xrightarrow[Private\ channel]{\langle \widehat{ID}_i, \widehat{PW}_i \rangle} FN$$

When fog node (FN) receives the registration request for the user, it first checks whether it is precisely registered or not by checking in the *user-table*. If the user is not in the *user-table*, FN calculates a new password for the user and updates its entry in the table. Finally, calculate the following:

$$a \xleftarrow{Rndomnumbers} FN$$
$$PIDW_i \leftarrow a$$
$$HIDS_i \leftarrow h(\widehat{ID}_i||Ku)$$
$$Ai \leftarrow h(\widehat{PW}_i||\widehat{ID}_i) \oplus HIDS_i$$
$$Bi \leftarrow h(\widehat{PW}_i||HIDS_i)\ mod\ n;$$
$$Ci \leftarrow h(\widehat{ID}_i||HIDS_i) \oplus PIDW_i$$
$$user\text{-}table \xleftarrow{update} \{h(PIDW_i), h(\widehat{PW}_i||\widehat{ID}_i)\}$$

– After updating the user information in the *user-table*, FN sends precisely calculated information to the user through secured channel:

$$SC_i \xleftarrow[writes]{\langle A_i, B_i, C_i, PIDW_i \rangle} FN$$
$$FN \xrightarrow[Private\ channel]{SC_i} User$$

3) Authentication Phase:
– *In the proposed model, the user is authenticated at two stages:* the application level (App) and the FN.

$$user(U_i) \xrightarrow{\{ID_i, PW_i, Bio_i\}} App(App_i)$$
$$b_i \leftarrow REP(Bio_i, pair_i)$$
$$\widehat{ID}_i \leftarrow h(ID)_i$$
$$\widehat{PW}_I^* \leftarrow h(PW_i||b_i)$$
$$HIDS_i^* \leftarrow A_i \oplus h(\widehat{PW}_i^*||\widehat{ID})$$
$$B_i^* \leftarrow h(\widehat{PW}_i^*||HIDS_i^*)$$

– After this, the App selects the time stamp t_1 and then it calculates the following and sends login request to FN

$$r_i \xleftarrow{random\ number} App_i$$
$$P \leftarrow PID \oplus A_i$$
$$\widehat{PID}_i \leftarrow C_i \oplus h(\widehat{ID}_i||HIDS_i^*)$$
$$R_i \leftarrow h(\widehat{ID}_i||PIDW_i||r_i)$$
$$M_1 \leftarrow (r_i||USID_j) \oplus H(\widehat{ID}_i||HIDS_i^*||T_1)$$
$$M_{UG} \leftarrow h(\widehat{ID}_i||HIDS_i^*||PIDW_i||R_i||T_1)$$
$$App_i \xrightarrow[login\ request]{h(P), \widehat{PID}_i, M_1, M_{UG}, HIDS^*, T_1, r_i, f_i} FN$$

– On receiving the request of the user, the FN first verifies the request by checking the validity of the time stamp and then checking the user details from the *user-table*

$$if\ \ h(PID)_i = h(p)$$
$$and\ \ HIDS = HIDS^*$$
$$login\ successful$$

– After the login is successful, FN calculates

$$K_i \leftarrow H(TID_i||HIDS_i^*||T_1)$$
$$R_i^* \leftarrow h(TID_i||r_i^*)$$
$$M_{UG}^* \leftarrow h(\widehat{ID}_i||HIDS_i^*||PID_i^{new}||R_i^*||T_1)$$
$$if\ M_{UG}^* = M_{UG}\ then\ user\ authenticated.$$

D. Second Phase

– *Input Parameters:* The algorithm takes several input parameters:
  – *FogNodes:* A list of fog nodes participating in the federated learning process. These fog nodes are responsible for aggregating and updating the local models of users in their network.
  – *GlobalModel:* The initial global model parameters. This is the starting point for the federated learning process.
  – *LearningRate:* The learning rate used during the local model updates. It controls the step size during the optimization process.
  – *LocalIterations:* The number of iterations each fog node performs for updating its local model using data from selected users.
  – *NumEpochs:* The total number of iterations over the entire federated learning process. Each epoch

Algorithm 1: Federated Learning With Decentralized Federated Averaging

Require:
FogNodes: List of fog nodes participating in the federated learning
GlobalModel: Initial global model parameters
LearningRate: Learning rate for local model updates
LocalIterations: Number of iterations for local model updates at each fog node
NumEpochs: Number of iterations over the entire federated learning process
Ensure:
Trained Global Model
Procedure FederatedLearning(FogNodes, GlobalModel, LearningRate, LocalIterations, NumEpochs)
    $w_{\text{global}} \leftarrow$ GlobalModel
for epoch $\leftarrow$ 1 to NumEpochs **do**

    for fog_node $\in$ FogNodes **do**
            $w_{\text{local}} \leftarrow$ CopyModel(w_{global})
        for iter $\leftarrow$ 1 to LocalIterations **do**
                selected_users $\leftarrow$ SelectUsersForUpdate(fog_node)
            for user $\in$ selected_users **do**
                    data $\leftarrow$ GetUserLocalData(user)
                    $w_{\text{local}} \leftarrow$ LocalModelUpdate(w_{local}, data, LearningRate)
            end for
        end for

        $w_{\text{fog}} \leftarrow$ LocalModelAggregation(fog_node, w_{local})
    end for

    $w_{\text{global}} \leftarrow$ GlobalModelAggregation(FogNodes, w_{fog})
end for
    return w_{global}
End Procedure

represents one round of local and global model updates.

– *Initialize Global Model:* The algorithm initializes the global model's parameters by setting w_global to the provided GlobalModel.
– *Federated Learning Loop:* The main federated learning loop runs for NumEpochs, allowing multiple iterations of local and global model updates.
– *Fog Nodes Initialization:* For each epoch, the algorithm proceeds with local model updates and aggregation at each fog node.
– *Local Model Update:* At the beginning of each epoch, each fog node initializes its local model (w_local) by making a copy of the current global model (w_global). This ensures that each fog node starts with the same initial model.
– *User Updates at Fog Node:* For LocalIterations iterations, each fog node performs the following steps:
  – Randomly selects a subset of users from its network (selected_users) for local model updates.
  – For each user in the selected_users, the fog node retrieves the user's local data (data) required for training the local model.

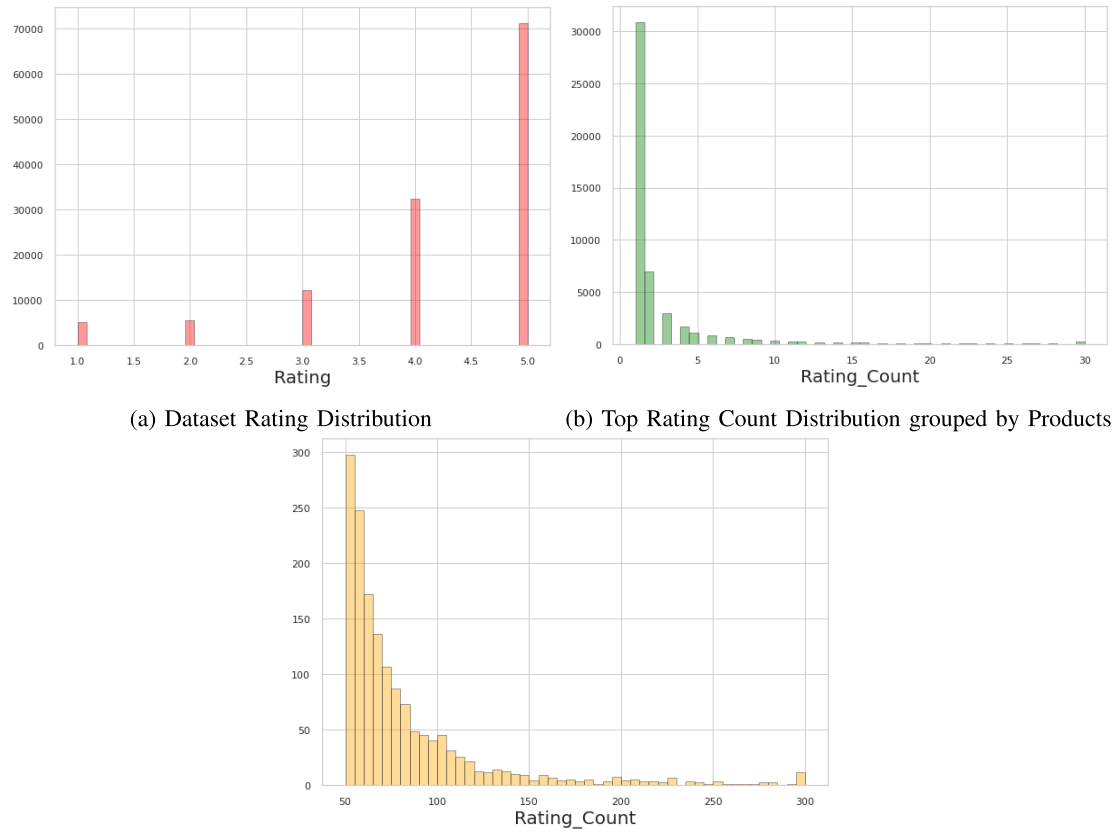
  Local Model Update for Each User: For each user in selected_users, the fog node updates its local model by applying the LocalModelUpdate function. This function updates the local model parameters using stochastic gradient descent (SGD) or any other optimization algorithm with the provided LearningRate and the user's local data.
– *Local Model Aggregation:* After the local model updates, each fog node performs local model aggregation (LocalModelAggregation) to obtain a refined local model (w_fog). The aggregation method may vary, but common approaches include taking the average of the local model updates or using weighted averaging.

  Global Model Aggregation: After all fog nodes have updated their local models and obtained refined local models, the algorithm performs global model aggregation (GlobalModelAggregation) to obtain the updated global model (w_global). This step involves combining the refined local models from all fog nodes to improve the global model.
– *Return Trained Global Model:* After all epochs are completed, the algorithm returns the trained global model (w_global), which now represents the collective knowledge learned from all fog nodes' local data while preserving user privacy and data decentralization.

V. Result and Discussion

The dataset is organized with columns representing userId, productId, Rating, and timestamp. Each row contains information about a specific user's rating for a particular product at a given timestamp. The "userId" field represents a unique identifier for each user, while the "productId" field identifies individual products within the system. The "Rating" field denotes the user's rating or feedback on a scale, which can be used to indicate their preference or satisfaction with the product. Higher ratings typically indicate a more positive sentiment towards the product, while lower ratings may suggest dissatisfaction. The "timestamp" field records the time when the user provided the rating, allowing for the possibility of considering temporal patterns in the recommendation process. This information can be useful for identifying trends and changes in user preferences over time.

Using collaborative filtering techniques, our recommendation system leverages the collective behavior of users to make personalized product recommendations. By analyzing the historical ratings and interactions of users with products, the system can identify patterns and similarities among users or products. This allows it to suggest items that a specific user might be interested in based on the preferences and behaviors of similar users (user-based collaborative filtering) or by identifying products that are similar to ones the user has interacted with positively. Collaborative filtering is a powerful and widely used recommendation approach, particularly in systems dealing with large datasets and diverse user preferences. By



(a) Dataset Rating Distribution

(b) Top Rating Count Distribution grouped by Products

(c) Top Rating Count Distribution grouped by Users

Fig. 4. Dataset Visualization.

providing accurate and relevant recommendations, our system aims to enhance user satisfaction, increase engagement, and ultimately improve the overall user experience.

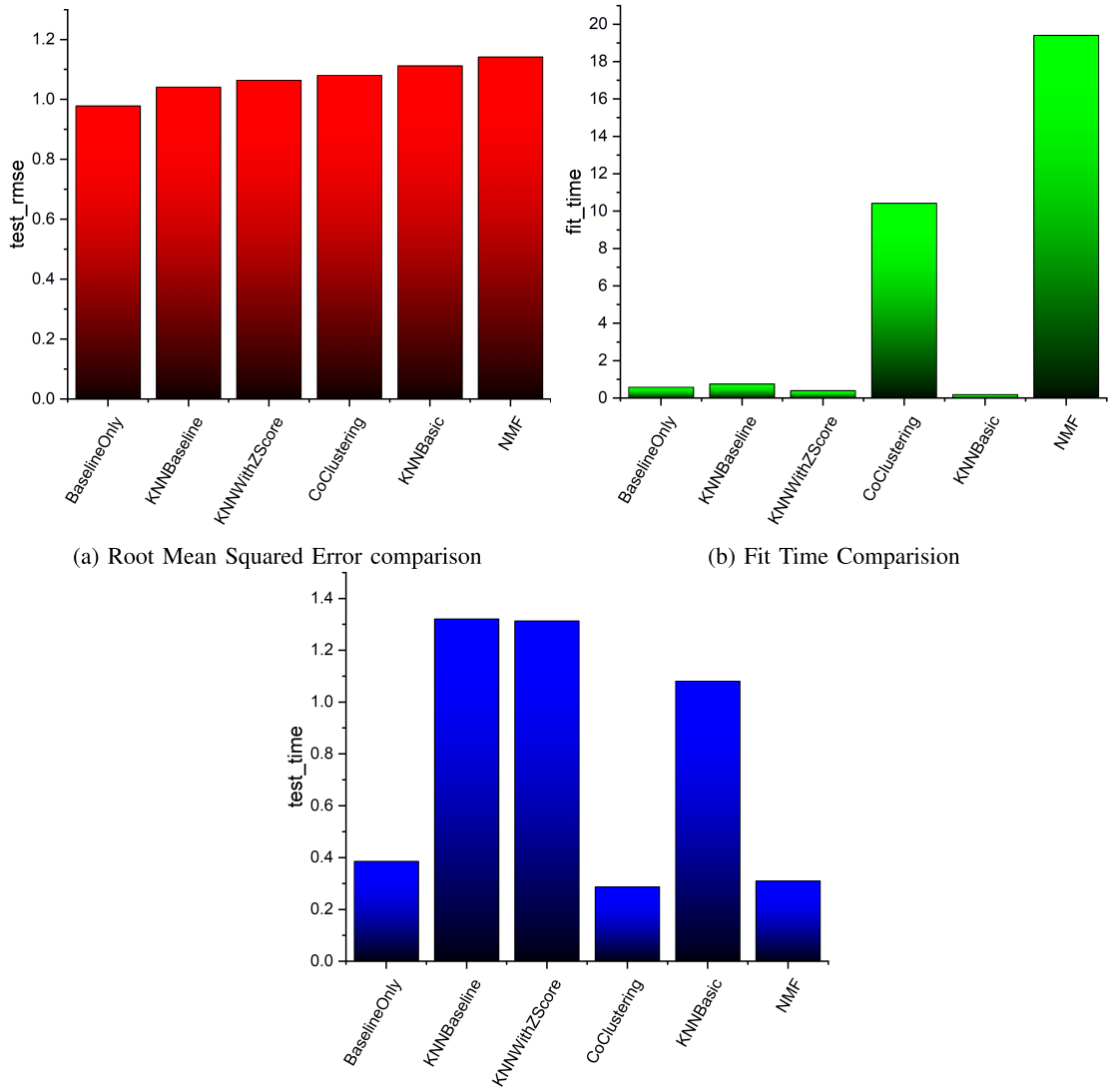
A. Data Preprocessing

– *Drop 'timestamp' attribute:* Since the 'timestamp' attribute is not directly relevant for building the collaborative filtering recommendation system, we can drop it from the dataset. Removing this attribute simplifies the data and reduces unnecessary information that won't be used for the recommendation process.
Create a subset of the original dataset: Creating a subset of the dataset can be useful for initial exploration or testing the recommendation system with a smaller sample of data. This step involves selecting a specific number of rows or a percentage of the original dataset randomly or based on certain criteria.
Shape of the data: Checking the shape of the data refers to finding out the number of rows and columns in the dataset. It provides us with a quick overview of the dataset's size and dimensionality, helping us understand the data's structure.
Checking the presence of missing values: It is essential to identify if there are any missing values in the dataset, as missing data can affect the performance of the recommendation system. If there are missing values, we need to decide on an appropriate strategy to handle them, such as imputation or removing rows with missing data.
– *Unique Users and Products Count:* Determining the number of unique users and products in the dataset is crucial for understanding the diversity and size of the user and product spaces. This step provides valuable insights into how many distinct users and products we have in our data, which is essential for collaborative filtering-based recommendations.

B. Data Visualization

1) Rating Distribution: From Figure 4 indicate that the majority of users have provided high ratings (4 or 5) for the products in the dataset, while lower ratings (1, 2, and 3) are comparatively less common.

– *Many users have rated 5 as shown with a huge spike of more than 70k records:* The large number of 5-star ratings suggests that a significant proportion of users had highly positive experiences with the products they interacted with. In many recommendation systems and review platforms, users tend to give higher ratings to products they liked or found highly satisfactory. This may indicate that the products in the dataset are generally well-liked by a large number of users.
This is followed by rating 4, which also has a high number with more than 30k records: A considerable number of 4-star ratings indicate that users had positive



(a) Root Mean Squared Error comparison

(b) Fit Time Comparision

(c) Test Time Comparision

Fig. 5. Result Comparison.

experiences with the products, although not as exceptional as a 5-star rating. A rating of 4 is often given when users are satisfied with the product but believe there is still some room for improvement. The relatively high count of 4-star ratings suggests that many products received favorable feedback from users.

– *More than 10k users have rated 3:* A rating of 3 indicates a neutral or mediocre experience with the product. Users might have given a rating of 3 when they found the product average or satisfactory but not exceptional. The number of 3-star ratings is still relatively high, suggesting that some products may have received mixed reviews, with users having diverse opinions about them.

Ratings 1 and 2 have been rated below 10k records: Lower ratings, such as 1 and 2, indicate negative experiences with the products. Users might have provided such ratings when they were dissatisfied, disappointed, or encountered issues with the product's quality or performance. The lower count of 1 and 2-star ratings compared to higher ratings is common in recommendation systems and review datasets because users may be less inclined to provide feedback for products they didn't like.

2) Rating Count: From Figure 4b and Figure 4c displays the count of ratings given by each user, showing how many users have given a specific number of ratings. The count limit for the plot has been clipped between 50 to 300, meaning any user with a rating count below 50 or above 300 will be represented by the respective values on the plot.

– *Many users have rated 50 times or below:* The plot indicates that there is a significant number of users in the dataset who have rated 50 times or less. This means that a substantial portion of users has provided ratings for relatively few products. Such users might be occasional users or those who have interacted with only a limited number of products in the system.

– *Ratings count by users gradually decreases, indicating few users have rated many products:* As the count of

ratings per user increases, the number of users with that particular rating count gradually decreases. This pattern suggests that there are relatively few users who have rated a large number of products. These users are likely active and engaged in the recommendation system, providing feedback on a wide range of products.

C. Result Comparison

In this section, we compare the performance of different recommendation systems on the datasets represents a comparison of different recommendation systems based on their performance metrics on a test dataset. The metrics evaluated in this comparison are Root Mean Squared Error (RMSE), fit time, and test time.

- *RMSE (Root Mean Squared Error):* RMSE is a common evaluation metric used in recommendation systems to assess the accuracy of the predictions made by the algorithms. It measures the difference between the predicted ratings and the actual ratings in the test dataset. Lower RMSE values indicate better accuracy, i.e., the algorithm's predictions are closer to the actual ratings.
- *Fit Time:* Fit time represents the time taken by the algorithm to train or fit the recommendation model on the training dataset. It measures how long it takes for the algorithm to learn patterns and relationships from the data.
- *Test Time:* Test time indicates the time taken by the algorithm to make predictions on the test dataset. It measures how long it takes for the algorithm to provide recommendations or predictions to users based on the learned model.

From Figure 5, we get following observations:

- The BaselineOnly algorithm has the lowest RMSE, indicating better accuracy in its predictions compared to the other algorithms.
- CoClustering has the highest fit time among all the algorithms, indicating that it takes the longest to train the model on the dataset. *NMF also has a relatively high fit time, suggesting that it takes considerable time to learn the latent features of the data during training.*
- *BaselineOnly and CoClustering have the lowest test times, indicating that they are faster in making predictions or providing recommendations to users during testing.*
- *NMF has the highest test time among all the algorithms, suggesting that it takes more time to generate recommendations for users during testing.*

VI. Conclusion

The challenge of developing personalised recommendation systems while ensuring user privacy and data security is addressed by our proposed approach for secure and privacy-preserving decentralised federated learning for personalised recommendations in consumer electronics devices using blockchain and homomorphic encryption. Our proposed approach uses the concept of blockchain and homomorphic encryption to provide decentralization and security. Our method safeguards user privacy without compromising the effectiveness of personalised suggestions, allowing businesses to earn their patronage while enhancing the satisfaction of their clientele. We hope that our work will encourage more investigation into the use of decentralised federated learning in the creation of individualised recommendation systems for consumer electronics products.

References

[1] O. Wahab, G. Rjoub, J. Bentahar, and R. Cohen, "Federated against the cold: A trust-based federated learning approach to counter the cold start problem in recommendation systems," *Inf. Sci.*, vol. 601, pp. 189–206, Jul. 2022.

[2] X. Huang, Y. Luo, L. Liu, W. Zhao, and S. Fu, "Randomization is all you need: A privacy-preserving federated learning framework for news recommendation," *Inf. Sci.*, vol. 637, Aug. 2023, Art. no. 118943.

[3] J. Guo, Q. Zhao, G. Li, Y. Chen, C. Lao, and L. Feng, "Decentralized federated learning with privacy-preserving for recommendation systems," *Enterprise Inf. Syst.*, vol. 17, no. 9, 2023, Art. no. 2193163.

[4] Z. Teimoori, A. Yassine, and M. Shamim Hossain, "A secure cloudlet-based charging station recommendation for electric vehicles empowered by federated learning," *IEEE Trans. Ind. Informat.*, vol. 18, no. 9, pp. 6464–6473, Sep. 2022.

[5] A. Gaurav, K. Psannis, and D. Peraković, "Security of cloud-based Medical Internet of Things (MIoTs): A survey," *Int. J. Softw. Sci. Comput. Intell. (IJSSCI)*, vol. 14, no. 1, pp. 1–16, 2022.

[6] K. Bonawitz et al., "Towards federated learning at scale: System design," 2019, *arXiv:1902.01046*.

[7] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[8] P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, nos. 1–2, pp. 1–210, 2019.

[9] Z. Li, M. Bilal, X. Xu, J. Jiang, and Y. Cui, "Federated learning-based cross-enterprise recommendation with graph neural networks," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 673–682, Jan. 2023.

[10] J. Qin, X. Zhang, B. Liu, and J. Qian, "A split-federated learning and edge-cloud based efficient and privacy-preserving large-scale item recommendation model," *J. Cloud Comput.*, vol. 12, no. 1, pp. 1–17, 2023.

[11] S. Ahmed, V. Kumar, K. Singh, A. Singh, V. Muthukumaran, and D. Gupta, "6G enabled federated learning for secure IoMT resource recommendation and propagation analysis," *Comput. Elect. Eng.*, vol. 102, Sep. 2022, Art. no. 108210.

[12] A. Abhishek, S. Binny, R. Johan, N. Raj, and V. Thomas, "Federated learning: Collaborative machine learning without centralized training data," *Int. J. Eng. Technol. Manag. Sci.*, vol. 6, no. 5, pp. 355–359, 2022.

[13] Y. Qin, M. Li, and J. Zhu, "Privacy-preserving federated learning framework in multimedia courses recommendation," *Wireless Netw.*, vol. 29, pp. 1535–1544, Jan. 2022.

[14] Mamta, B. B. Gupta, K.-C. Li, V. C. M. Leung, K. E. Psannis, and S. Yamaguchi, "Blockchain-assisted secure fine-grained searchable encryption for a cloud-based healthcare cyber-physical system," *IEEE/CAA J. Autom. Sin.*, vol. 8, no. 12, pp. 1877–1890, Dec. 2021.

[15] Y. Li, B. Ding, and J. Zhou, "A practical introduction to federated learning," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2022, pp. 4802–4803.

[16] J. Bisht and V. S. Vampugani, "Load and cost-aware min-min workflow scheduling algorithm for heterogeneous resources in fog, cloud, and edge scenarios," *Int. J. Cloud Appl. Comput. (IJCAC)*, vol. 12, no. 1, pp. 1–20, 2022.

[17] D. H. Mahlool and M. H. Abed, "A comprehensive survey on federated learning: Concept and applications," 2022, *arXiv:2201.09384*.

[18] Z. Fan, H. Fang, and M. P. Friedlander, "A dual approach for federated learning," 2022, *arXiv:2201.11183*.

[19] G. N. Nguyen et al., "Secure blockchain enabled cyber–physical systems in healthcare using deep belief network with ResNet model," *J. Parallel Distrib. Comput.*, vol. 153, pp. 150–160, Jul. 2021.

[20] A. Gaurav et al., "A comprehensive survey on machine learning approaches for malware detection in IoT-based enterprise information system," *Enterprise Inf. Syst.*, vol. 17, no. 3, 2023, Art. no. 2023764.

[21] A. Raj and S. Prakash, "A privacy-preserving authentic healthcare monitoring system using blockchain," *Int. J. Softw. Sci. Comput. Intell. (IJSSCI)*, vol. 14, no. 1, pp. 1–23, 2022.

[22] A. N. El-Kassar, M. M. Yunis, and M. N. El Dine, "A production model with continuous demand for imperfect finished items resulting from the quality of raw material," in *Proc. ICORES*, 2020, pp. 263–269.
[23] W. Ali, R. Kumar, Z. Deng, Y. Wang, and J. Shao, "A federated learning approach for privacy protection in context-aware recommender systems," *Comput. J.*, vol. 64, no. 7, pp. 1016–1027, 2021.
[24] J. Huang, Z. Tong, and Z. Feng, "Geographical POI recommendation for Internet of Things: A federated learning approach using matrix factorization," *Int. J. Commun. Syst.*, to be published.
[25] A. Almomani et al., "Phishing website detection with semantic features based on machine learning classifiers: A comparative study," *Int. J. Semant. Web Inf. Syst. (IJSWIS)*, vol. 18, no. 1, pp. 1–24, 2022.
[26] J. V. Tembhurne, M. M. Almin, and T. Diwan, "Mc-DNN: Fake news detection using multi-channel deep neural networks," *Int. J. Semant. Web Inf. Syst. (IJSWIS)*, vol. 18, no. 1, pp. 1–20, 2022.
[27] C. L. Stergiou et al., "InFeMo: Flexible big data management through a federated cloud system," *ACM Trans. Internet Technol. (TOIT)*, vol. 22, no. 2, pp. 1–22, 2021.
[28] B. Hu, A. Gaurav, C. Choi, and A. Almomani, "Evaluation and comparative analysis of semantic web-based strategies for enhancing educational system development," *Int. J. Semant. Web Inf. Syst. (IJSWIS)*, vol. 18, no. 1, pp. 1–14, 2022.
[29] M. Xu et al., "Multiagent federated reinforcement learning for secure incentive mechanism in intelligent cyber–physical systems," *IEEE Internet Things J.*, vol. 9, no. 22, pp. 22095–22108, Nov. 2022.
[30] O. Kadri, A. Benyahia, and A. Abdelhadi, "Tifinagh handwriting character recognition using a CNN provided as a Web service," *Int. J. Cloud Appl. Comput. (IJCAC)*, vol. 12, no. 1, pp. 1–17, 2022.
[31] S. A. Alamer, Q. M. Ilyas, M. Ahmad, and D. Irfan, "A metaphoric design of electronic medical record (EMR) for periodic health examination reports: An initiative to cloud's medical data analysis," *Int. J. Cloud Appl. Comput. (IJCAC)*, vol. 12, no. 1, pp. 1–18, 2022.
[32] S. Li, D. Qin, X. Wu, J. Li, B. Li, and W. Han, "False alert detection based on deep learning and machine learning," *Int. J. Semant. Web Inf. Syst. (IJSWIS)*, vol. 18, no. 1, pp. 1–21, 2022.
[33] A. Jalalirad, M. Scavuzzo, C. Capota, and M. R. Sprague, "A simple and efficient federated recommender system," in *Proc. 6th IEEE/ACM Int. Conf. Big Data Comput., Appl. Technol.*, 2019, pp. 53–58.
[34] F. Chen, Z. Dong, Z. Li, and X. He, "Federated Meta-learning for recommendation," 2018, *arXiv:1802.07876*.
[35] K. Muhammad et al., "FedFast: Going beyond average for faster training of federated recommender systems," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 1234–1242.
[36] H. Sun, Y. Chen, K. Sha, and Y. Wu, "Poster: Ensemble federated edge learning for recommender systems," in *Proc. IEEE/ACM 7th Symp. Edge Comput. (SEC)*, 2022, pp. 291–292.
[37] G. Salloum and J. Tekli, "Automated and personalized nutrition health assessment, recommendation, and progress evaluation using fuzzy reasoning," *Int. J. Human-Comput. Stud.*, vol. 151, Jul. 2021, Art. no. 102610.
[38] G. Salloum and J. Tekli, "Automated and personalized meal plan generation and relevance scoring using a multi-factor adaptation of the transportation problem," *Soft Comput.*, vol. 26, no. 5, pp. 2561–2585, 2022.
[39] P. Li, T. Li, X. Wang, S. Zhang, Y. Jiang, and Y. Tang, "Scholar recommendation based on high-order propagation of knowledge graphs," *Int. J. Semant. Web Inf. Syst. (IJSWIS)*, vol. 18, no. 1, pp. 1–19, 2022.
[40] J. Xiao, X. Liu, J. Zeng, Y. Cao, and Z. Feng, "Recommendation of healthcare services based on an embedded user profile model," *Int. J. Semant. Web Inf. Syst. (IJSWIS)*, vol. 18, no. 1, pp. 1–21, 2022.
[41] Y. Liu, L. Zhang, N. Ge, and G. Li, "A systematic literature review on federated learning: From a model quality perspective," 2020, *arXiv:2012.01973*.
[42] B. Pejó, "The good, the bad, and the ugly: Quality inference in federated learning," 2020, *arXiv:2007.06236*.
[43] D. C. Verma, G. White, S. J. Julier, S. Pasteris, S. Chakraborty, and G. H. Cirincione, "Approaches to address the data skew problem in federated learning," in *Proc. Def. + Commer. Sens.*, 2019, pp. 1–16.
[44] G. George and A. M. Lal, "A personalized approach to course recommendation in higher education," *Int. J. Semant. Web Inf. Syst. (IJSWIS)*, vol. 17, no. 2, pp. 100–114, 2021.
[45] N. Bouacida and P. Mohapatra, "Vulnerabilities in federated learning," *IEEE Access*, vol. 9, pp. 63229–63249, 2021.
[46] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Gener. Comput. Syst.*, vol. 115, pp. 619–640, Feb. 2021.
[47] M. Asad, A. Moustafa, and C. Yu, "A critical evaluation of privacy and security threats in federated learning," *Sensors (Basel, Switzerland)*, vol. 20, no. 24, p. 7182, 2020.
[48] T. D. T. Nguyen and M. T. Thai, "Preserving privacy and security in federated learning," 2022, *arXiv:2202.03402*.