Addressing Non-IID Data and Model Heterogeneity in Federated Learning: A Novel Local Feature Extraction and Classification Scheme

Qingyun Wei, Richard Jiang and Ahmed Bouridane

***Abstract*—Federated learning, an innovative decentralized learning paradigm, has garnered considerable attention for its capacity to train on disparate data sources while upholding privacy. However, the challenges stemming from non-IID data distribution and model heterogeneity necessitate sophisticated solutions. In response, we propose a groundbreaking framework known as Personalized Federated Learning (PFL). Central to our approach is the bifurcation of each user's model into two distinct strata. Firstly, a foundational, lower-dimensional feature extraction layer is crafted to align with and be regulated by a shared representation. This is coupled with a local, higher-dimensional feature classification layer. This segregation ensures consistent feature extraction across the federation while accommodating personalized classification tasks. A pivotal aspect of our methodology is the incorporation of homomorphic encryption, which safeguards the confidentiality of both data and model parameters during the learning process. Our method's robustness and effectiveness are validated through empirical evaluations conducted on CIFAR-10 and MNIST datasets. Supplementary ablation studies and efficiency assessments further underscore the reliability and computational merits of our design. In summary, our findings position the PFL approach as an innovative stride in federated learning, distinguished by its unwavering commitment to privacy through the integration of homomorphic encryption.**

***Index Terms*—Personalized Federated Learning, Homomorphic Encryption, Non-IID Data, CKKS**

I. Introduction

A. Federated Learning

RECENTLY, artificial intelligence technology based on big data has developed rapidly in many fields such as medical care, finance, and automatic driving. Many large-scale and complex tasks have been successfully solved with the assistance of big data and artificial intelligence. Traditional machine learning algorithms require the trainer to first collect user data, and obtain an artificial intelligence model through intensive training on the massive user data collected. This traditional training process introduces many issues, such as data security and privacy protection. In traditional machine learning methods, data is stored in a central server. When the server is attacked or the data is leaked, then the user's data will be lost or polluted. At the same time, if the uploaded user data involves sensitive information of personal privacy, the amount and willingness of users to share data will also be reduced. With the development of artificial intelligence, this phenomenon will bring serious isolated data island problem [1].

In order to solve the contradiction between isolated data islands and the development of artificial intelligence, Google [2] proposed the concept of federated learning in 2017. Federated learning is a distributed machine learning method. Its core idea is to allow model training and parameter update on distributed devices while protecting data privacy, and no data is shared between devices. Federated learning realizes distributed machine learning through the joint participation of a server and multiple clients. Wherein, the client may be a personal terminal device, or a plurality of different enterprises. The client performs local training of the model and obtains the trained model parameters. The server is responsible for integrating some or all of the model parameters trained by the client, and then synchronizing the aggregated model to the client for the next round of iterative training. At the same time, this distributed computing method distributes the computing burden to client devices, reducing the computing and storage pressure on the central server.

1) Formalization of Federated Learning: Assume there are N participants (denoted as $P_1, P_2, \ldots, P_N$) and a central server (denoted as S). The federated learning process can be divided into the following step:

1) **Initialization:** The central server S initializes the global model parameters θ_0
2) **Participant Selection:** In each round of iteration, a subset of participants is selected as the current participants. Denote it as $C_t \subseteq \{P_1, P_2, \ldots, P_N\}$
3) **Model Distribution:** The central server S sends the current global model parameters θ_t to the current paricipants C_t.
4) **Participant Training:** Each participant $P_i \in C_t$ receives the model parameters θ_t and performs local model training using ther own local dataset. Participants compute gradients based on their local data and update the model parameters using the optimization algorithm: $\theta^i_{t+1} \leftarrow f(\theta_t, D_i)$ where D_i is the local dataset of P_i, $f(\cdot)$ is the optimization algorithm.
5) **Model Aggregation:** The central server S collects the updated model parameters θ^i_{t+1} from all participants and performs model aggregation. This aggregation operation is typically a simple average: $\theta_{t+1} = \frac{1}{|C_t|}\Sigma_{i \in C_t}\theta^i_{t+1}$.

This work was supported in part by the UK EPSRC under Grant EP/P009727/1, and the Leverhulme Trust under Grant RF-2019-492. (Correspondent author: Richard Jiang, e-mail: r.jiang2@lancaster.ac.uk).

Qingyun Wei and Richard Jiang are with Lancaster University, Bailrigg, Lancaster, Lancashire, UK, LA1 4WA

Ahmed Bouridane is with Centre for Data Analytics and Cybersecurity (CDAC), University of Sharjah, Sharjah, UAE.

Manuscript received Dec 7th, 2023; revised xxx xxx, 2024.

6) **Repeat Iterations:** Repeat steps 2-5 until the predetermined number of iterations or convergence conditions are reached.

By formalizing the federated learning process in this way, federated learning enables collaborative model training and parameter sharing among multiple participants without directly sharing their data. This most basic federated learning algorithm is called FedAvg [2].

2) Risks in Federated Learning: Although federated learning avoids the problem of direct data leakage, there are still a large number of security risks of indirect privacy leakage:

1) **Privacy may be leaked during the exchange of intermediate parameters.** The raw data in the training process may be inferred and exposed to third parties during the transfer process. [3] found that according to the intermediate parameters of federated learning, it can be inferred whether the content of the record comes from a certain member.
2) **There are risks in directly publishing the model trained by federated learning.** Federated learning is essentially a distributed machine learning technology. In the case of excessive pursuit of the accuracy of the model training set, this may make the model remember sensitive information during the training process. Through repeated testing of the resulting model interface, the attacker can infer the specific data of the training set and leak private information [4].
3) **Unreliable clients may bring malicious attacks.** Unconditional trust in the client may bring more serious hidden dangers, such as carefully uploaded harmful information will seriously damage the model effect, and even affect the data privacy protection of other clients [5].

In summary, federated learning without protection still faces serious security risks. How to further protect the intermediate parameter information in the federated learning process is an important research content to promote the application of federated learning in practical scenarios. Therefore, the current hot research content focuses on improving the privacy of federated learning intermediate parameter transfer through algorithms.

B. Homomorphic Encryption

Encryption algorithms play a very important role in data security [6]. Sensitive information can be converted into a form that cannot be directly understood through encryption algorithms to protect the confidentiality, integrity and reliability of data. Encryption algorithms have important applications in fields such as identity verification and secure communication. However, traditional encryption algorithms need to decrypt the encrypted data before performing calculations or operations on the encrypted data, and then re-encrypt after the calculation is completed. Such a complex encryption and decryption process will bring huge calculations, which will make it impossible to proceed smoothly in some scenarios that require a lot of calculations. In this context, cryptographers have developed a special encryption technology called homomorphic encryption.

Homomorphic encryption allows computational operations to be performed in an encrypted state without first decrypting the data [7]. Homomorphic encryption enables common mathematical operations such as addition and multiplication to be performed on data in an encrypted state, providing a secure way of data sharing. Compared with traditional encryption algorithms, homomorphically encrypted data continues to remain encrypted, providing a higher level of data privacy protection, making cloud computing, federated learning and other scenarios possible.

Homomorphic encryption can be divided into partial homomorphic encryption, somewhat homomorphic encryption and fully homomorphic encryption according to the types and times of supported ciphertext operations:

1) **Partially Homomorphic Encryption (PHE):** PHE allows a single kind of operation on encrypted data, such as addition or penalty, and the number of operations is unlimited, but not simultaneously. Common partial homomorphic encryption algorithms include additive homomorphic encryption and multiplication homomorphic encryption [8], [9].
2) **Somewhat Homomorphic Encryption (SHE):** SHE only supports a limited number of addition and multiplication operations. It is a weakened version of fully homomorphic encryption with less overhead, but it is also easier to implement. The most common SHE is leveled fully homomorphic encryption (leveled-FHE) [10], but due to the bounded depth, it can only support a limited number of operations, so leveled-FHE is not suitable for training deep neural networks.
3) **Fully Homomorphic Encryption (FHE):** FHE is an encryption algorithm based on ideal lattices theory [11]. Any algorithm on the ciphertext is supported, and the number of calculations is not limited. FHE is often accompanied by a complex bootstrapping process, which can bring huge overhead. At present, improved FHE has been proposed one after another, and these studies are devoted to reducing noise and improving efficiency.

1) Formalization of Fully Homomorphic Encryption: A fully homomorphic encryption system consists of four core algorithms:

1) **Key Generation Algorithm** $KeyGen(1^\lambda) \rightarrow sk$: Generate the keys needed for encryption and decryption sk. For simplicity, we assume here that the encryption key is equal to the decryption key.
2) **Encryption Algorithm** $Enc(sk, m \in 0, 1^{|M|}) \rightarrow ct$: Encrypt the original text m into ciphertext ct.
3) **Decryption Algorithm** $Dec(sk, ct) \rightarrow m$: Decryption the ciphertext ct into original text m.
4) **Operation Algorithm** $Eval(F, ct_1, \ldots, ct_l) \rightarrow \hat{ct}$: Combine the l ciphertexts, pass through a binary logic circuit F, and finally get the combined ciphertext $\hat{ct}$, making: $Dec(sk, \hat{ct}) = F(m_1, \ldots, m_l)$.

It can be seen that the last operation algorithm $Eval$, is the core of fully homomorphic encryption that distinguishes it from other homomorphic encryption. In actual scenarios, we need to provide encrypted data samples $ct_1, \ldots, ct_l$, convert

the process of model analysis and prediction into a binary logic circuit F, and then use $Eval$ to obtain the final prediction result.

2) Limitations of Fully Homomorphic Encryption: Although fully homomorphic encryption (FHE) technology has important privacy protection and secure computing features, it also has some limitations. 1) The computational complexity of fully homomorphic encryption is very high [12]. Adding and multiplying encrypted data consumes a lot of computing resources and time. 2) The ciphertext after a fully homomorphic encryption is usually much larger than the original plaintext data, which will require more resources and bandwidth when storing and transmitting the ciphertext [13]. 3) At present, homomorphic encryption has limitations on dynamic data support [14]. If the data set changes, it needs to recalculate and generate ciphertext, which brings overhead.

Researchers are currently working to improve the technique and increase the efficiency and scalability of fully homomorphic encryption. The limitations of fully homomorphic encryption can be further solved by optimizing algorithms, hardware acceleration or combining other privacy protection technologies.

C. Aims and Objectives

This section outlines the two aims of this project and their motivations. The non-iid nature of distributed data and the heterogeneity of models pose major challenges to its effectiveness, and the privacy protection of intermediate parameters in federated learning also needs to be resolved urgently. This paper proposes a novel Personalized Federated Learning (PFL) scheme that aims to address these challenges. Specifically, we aim to address the following two key issues:

1. A reasonable federated learning algorithm to deal with the non-iid nature of distributed data and the heterogeneity of the model.

In federated learning, participants usually have different data distributions and characteristics. This makes the training of the model challenging, because traditional centralized learning methods usually assume that the data is independent and identically distributed. Therefore, the first aim is to develop reasonable federated learning algorithms to cope with non-IID of distributed data and heterogeneity of models. By considering data variance and feature heterogeneity, we can better model and utilize distributed data, improving the effect and accuracy of federated learning.

Objectives:

- Develop federated learning algorithms that adapt to distributed data to fully utilize information from parties with different data distributions.
- Considering the heterogeneity of the model, design a federated learning algorithm that can effectively handle different features and structures.
- Optimize the model aggregation strategy so that it can better fuse model updates from different parties.

2. Reliable homomorphic encryption federated learning mechanism to solve the privacy protection of intermediate parameters of federated learning.

Federated learning involves the exchange and update of model parameters among multiple parties, which may involve the risk of leakage of sensitive information. To solve this problem, the second goal is to design a reliable homomorphic encrypted federated learning mechanism that ensures privacy protection during parameter updates. Homomorphic encryption technology allows calculations to be performed on encrypted data without decryption, thus protecting the privacy of sensitive information. By introducing a reliable homomorphic encryption scheme, we can protect private information such as model parameters and gradients among participants, and enhance the security and privacy of federated learning.

In summary, the objectives of our research include: 1) Introduce a robust homomorphic encryption scheme to protect the privacy of model parameters and gradients transmitted between parties; 2) Ensure computational efficiency during homomorphic encryption and improve real-time and scalability of federated learning.

D. Work Overview

The structure of this paper is outlined as follows: In section 2, subsequent to the Introduction, we survey pertinent literature encompassing three key domains: federated learning for non-IID data, algorithms associated with homomorphic encryption, and the amalgamation of federated learning with homomorphic encryption; Section 3 delineates the methodology underpinning our proposed personalized federated learning algorithm. This chapter also elaborates on the datasets employed and the specific experimental settings adopted; Section 4 presents our empirical findings, encapsulated in a series of figures and tables. The experimental scope encompasses comparative analyses with prior studies and an examination of model convergence; Section 5 provides a comprehensive discussion of the results, drawing out salient insights pertinent to the research question and candidly addressing any methodological limitations. Concluding, section 6 synthesizes the overarching findings and contributions of this project, while also positing avenues for future research.

II. Related Work

A. Personalized Federated Learning for Non-iid Data

In traditional application scenarios, data is stored in a central server, so centralized machine learning training can obtain the overall information of all data. However, in a distributed scenario, the data is only stored locally, so it will cause the problem of inconsistency in the distribution of data. Therefore, general federated learning methods face many challenges.

First, general federated learning has poor convergence on highly heterogeneous data. This is because multiple rounds of synchronization on non-iid local data will cause client drift. Taking the FedAvg algorithm mentioned in Section 1.1 as an example, when learning on non-iid data, the accuracy will be significantly reduced. This process can be explained as shown in Figure 1. FedAvg model parameter updates move towards the mean value of the client model parameters. When the data are IID, the average model is close to the global optimum w^* because it is equidistant from the local optimum w_1^* and w_2^*.

However, when the data is non-IID, the distance between the global optimum w^* and the local optimum are not equal. In the figure w^* is closer to w_2^*. Therefore, the averaged model w_{avg}^* cannot converge to a true global optimum w^*. This will bring convergence problems to the training process of FedAvg.

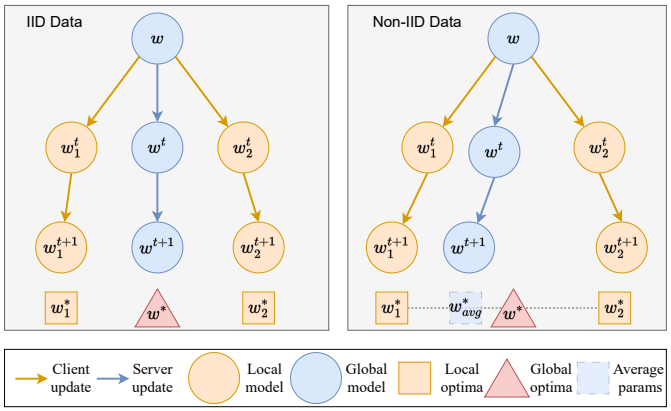


Fig. 1. Illustration of client drift in FedAvg.

In addition, general federated learning lacks personalized solutions. FedAvg trains a single globally shared model to fit the averaged clients. When the data distribution of individual clients is significantly different, it will be difficult to obtain personalized solutions for specific situations. For example, people from different regions may have different dietary preferences. If you want to use federated learning to train a recipe application, you need to have more targeted predictions for each user to further meet the needs of users.

1) Scaffold: In order to deal with this kind of problem, a new federated optimization algorithm **Scaffold** is proposed [15]. **Scaffold** introduces the server control variable $\boldsymbol{c}$ and the client control variable $\boldsymbol{c}_i$, which contain the update direction information of the model. The relationship between the two control variables is:

$$\boldsymbol{c} = \frac{1}{N}\sum \boldsymbol{c}_i \tag{1}$$

In each round of communication, the server-side parameters $(\boldsymbol{x}, \boldsymbol{c})$ are sent to the selected client S. Each selected client initializes its local model as $\boldsymbol{y}_i \leftarrow \boldsymbol{x}$, and then performs local updates:

$$\boldsymbol{y}_i \leftarrow \boldsymbol{y}_i - \eta_l(g_i(\boldsymbol{y}_i) + \boldsymbol{c} - \boldsymbol{c}_i) \tag{2}$$

where η_l is the local step-size, $g_i(\boldsymbol{y}_i)$ is the unbiased stochastic gradient of $\boldsymbol{y}_i$. After the K local updates are completed, the local control variable $\boldsymbol{c}_i$ also needs to perform two update options:

$$\boldsymbol{c}_i^+ \leftarrow \begin{cases} \text{Option I.} & g_i(\boldsymbol{x}), \text{ or} \\ \text{Option II.} & \boldsymbol{c}_i - \boldsymbol{c} + \frac{1}{K\eta_l}(\boldsymbol{x} - \boldsymbol{y}_i) \end{cases} \tag{3}$$

Option I may be more stable than Option II, but II is less computationally expensive. After the local control variables are updated, the global model is updated:

$$\begin{aligned} \boldsymbol{x} &\leftarrow \boldsymbol{x} + \tfrac{\eta_g}{|S|}\textstyle\sum_{i\in S}(\boldsymbol{y}_i - \boldsymbol{x}) \\ \boldsymbol{c} &\leftarrow \boldsymbol{c} + \tfrac{1}{N}\textstyle\sum_{i\in S}(\boldsymbol{c}_i^+ - \boldsymbol{c}_i) \end{aligned} \tag{4}$$

Or aggregate the updated model directly:

$$\begin{aligned} \boldsymbol{x} &\leftarrow \tfrac{\eta_g}{|S|}\textstyle\sum_{i\in S}\boldsymbol{y}_i \\ \boldsymbol{c} &\leftarrow \tfrac{1}{N}\textstyle\sum_{i\in S}\boldsymbol{c}_i^+ \end{aligned} \tag{5}$$

When the local control variable $\boldsymbol{c}_i$ is always 0, the update formula becomes:

$$\boldsymbol{y}_i \leftarrow \boldsymbol{y}_i - \eta_l g_i(\boldsymbol{y}_i) \tag{6}$$

That is, **Scaffold** degenerates into FedAvg. By adding a correction term $\boldsymbol{c} - \boldsymbol{c}_i$ in the local model update formula, **Scaffold** overcomes the gradient difference and effectively alleviates the client drift problem.

2) FedProx: In FedAvg, although the increase in the number of local iterations can reduce the communication cost, too many iterations may make some devices with insufficient computing power unable to complete the training. At the same time, a large number of iterations can easily make the local model of the device deviate from the global model, affecting global convergence. This kind of heterogeneity is also called device heterogeneity, that is, there are differences in communication and computing capabilities between devices, which may cause updates between different devices to be out of sync. Therefore, **FedProx** [16] is proposed to solve this heterogeneous problem.

Compared with FedAvg, **FedProx** allows rough minimization of local subproblems in different ways in different devices and different local epochs according to the device's available system resources. This is not to directly discard devices with insufficient computing power, but to alleviate the problem of device heterogeneity. What's more, a proximal term is added to the local subproblem. The regularization item subtracts the global model of the previous round, so that the local update will not deviate too much from the global model:

$$\min_w h_k(w; w^t) = F_k(w) + \frac{\mu}{2}||w - w^t||^2 \tag{7}$$

We update the parameters of the local model:

$$w_k^{t+1} \approx argmin_w h_k(w; w^t) \tag{8}$$

Then the server aggregates the w as :

$$w^{t+1} = \frac{1}{K}\sum_{k\in S_t} w_k^{t+1} \tag{9}$$

where K is the subset of devices. When $\mu = 0$, the local solver selects SGD optimization, different devices use the same local epoch in each round of global update, and **FedProx** degenerates into FedAvg.

In general, **FedProx** alleviates data heterogeneity and improves the stability of global model convergence by adding a proximal term. At the same time, device heterogeneity is alleviated through tolerating partial work.

3) Ditto: In order to improve the personalization effect of federated multi-task learning, it is necessary to improve the fairness and robustness of federated learning at the same time. Fairness means that local models of different devices have the same performance. Robustness refers to preventing malicious nodes from sending arbitrary updates to the server to corrupt the training phase. Previous studies often only consider fairness or robustness alone, and improving fairness comes at

the expense of robustness. Improving robustness may filter out some rare but valuable update parameters, reducing fairness. Therefore, **Ditto** [17] is proposed to improve both fairness and robustness. Specifically, **Ditto** proposed to apply multi-task learning to solve this difficulty.

Ditto's global objective is to aggregate the local models participating in the training, and all current aggregation methods can be applied, such as FedAvg, FedProx, etc. The global goals are as follows:

$$\min_{w} G(F_1(w), \ldots, F_K(w)) \tag{10}$$

where $F_k(w)$ is the local objective function and $G(\cdot)$ is the aggregation function. **Ditto**'s local objective function is as follows:

$$\begin{array}{c} \min_{v_k} h_k(v_k; w^*) := F_k(v_k) + \frac{\lambda}{2}||v_k - w^*||^2 \\ w^* \in arg\min_w G(F_1(w), \ldots, F_K(w)) \end{array} \tag{11}$$

Ditto adds a regularization term to the original local objective function, where v_k is the personalized model of device k, and w^* is the global model.

Since malicious nodes will destroy the training of the global model, the effect of simply applying the global model to heterogeneous devices may be very bad; while benign nodes cannot train a better model relying solely on a small amount of local data. **Ditto** makes a trade-off between the personalized model and the global model through the hyperparameter λ. The larger the λ, the closer the personalized model v_k is to the global model w^*. The smaller the λ, the more the personalized model v_k deviates from the poisoned global model w^*. Each device finds its own personalized model between the global model and the local model by adjusting the value of λ. This improves robustness and fairness at the same time.

4) Other PFL Methods: In addition to the classic methods mentioned above, many other model-based personalized federated learning methods have been proposed in recent years. **FedCL** [18] considers elastic weight integration using a continuous learning domain in a regularized local loss function. When these parameters are transferred to the client, a penalty step is performed on the client to prevent important parameters of the global model from being changed when adapting the global model and the client's local data. This mitigates the difference in weights between the global and local models and preserves the knowledge of the global model.

FL-MOON [19] based on contrastive learning is proposed. The goal of **FL-MOON** is to reduce the distance between the representations learned by the local model and the global model, and to increase the distance between the representations learned between a given local model and its previous local model. distance between. By alleviating weight ambiguity and speeding up fusion, this emerging approach enables each client to learn a representation close to the global model. It also accelerates learning by encouraging local models to improve on the previous version.

MAML [20] is a meta-learning algorithm that aims to improve a learning algorithm by exposing it to various subtasks. **Fed-ML** [21] maps the meta-testing step to the personalization process of federated learning by mapping the meta-learning step in MAML to the FL global model training process. Due to the similarity in the formulation of meta-learning and federated learning algorithms, meta-learning techniques can be applied to improve the global training process and enable fast personalization on the client side.

FedMD [22] is a FL framework based on transfer learning and knowledge transfer for clients to design independent models using their own private data. Before the FL training and knowledge transfer stage, a model pre-trained on a public dataset is first used for transfer learning. Each client then fine-tunes the model on its private data. In personalized federated learning, domain adaptive transfer learning technology is usually used to reduce the domain gap between the source domain and the target domain and improve the degree of personalization. Similar approaches have many applications in the field of medical federated learning [23], [24].

B. Algorithms for Fully Homomorphic Encryption

Fully homomorphic encryption (FHE) can effectively protect privacy and provide computable capabilities during the transfer of intermediate parameters in federated learning. This section will introduce the three main stages of the current development of fully homomorphic encryption.

1) FHE Based on Ideal Lattices: The method of fully homomorphic encryption was first proposed by Gentry [11]. Its core idea is to first construct a somewhat homomorphic encryption, which can homomorphically calculate circuits of a certain depth. Then compress the decryption circuit so that it can homomorphically compute its own enhanced decryption circuit, and obtain a homomorphic encryption scheme that can be bootstrapped. Finally, based on the assumption of cycle safety, the bootstrap operations are executed in an orderly manner, and a scheme that can homomorphically compute arbitrary circuits is obtained, that is, FHE. At the same time, based on the ICP assumption on the ideal lattice, combined with the assumption of sparse subsets and cycle safety, Gentry gives a specific implementation plan.

[25] used the principal ideal lattice and introduced batch processing technology to realize a preliminary scheme of fully homomorphic encryption. A batch version of this method allows encryption of a pack of plaintext vectors into a single ciphertext using the Chinese remainder theorem. [26] further optimized this algorithm to allow simultaneous processing of multiple messages. [27] reduce the bit complexity of refreshing the ciphertext, and their algorithm can be applied to different FHE schemes.

The ideal lattice-based FHE scheme is the first-stage FHE algorithm, which is based on a mathematical structure that is difficult to implement effectively. Realized value in industry is limited, but opens up the field of FHE research.

2) FHE based on LWE and RLWE: LWE is also known as the fault-tolerant learning problem. It is necessary to find a set of coefficients so that the linear combination of a set of base vectors approaches the target vector infinitely. The size of the noise error is used to define how close we need to be to the target vector. RLWE is the in-ring version of the LWE problem. In a polynomial ring $R_q = \mathbb{Z}_q[x]/f(x)$, each element is a polynomial. Each operation is equivalent to

operating on multiple elements, that is, it can encrypt multiple bits of plaintext at one time. Compared with LWE, RLWE can greatly improve the efficiency to meet actual needs.

Using bootstrapping techniques, [10], [28] introduced two FHE schemes based on LWE and RLWE problems and cycle safety assumptions. These works started the second generation of FHE programs, in which a symmetric scheme based on LWE, called BV.

Ref. [29] proposed a method for defining hierarchical fully homomorphic schemes that avoid computationally expensive bootstrapping techniques. On this basis, they defined the **BGV** scheme. This new approach makes the scheme applicable to real-world scenarios, thereby attracting increasing interest from the research community. The authors introduce two variants of the BGV scheme (one based on the LWE assumption and the other on the RLWE assumption). The RLWE-based BGV scheme described in Scheme 1 is more efficient than the LWE scheme, which is implemented in the widely used FHE library HElib.

BFV [30] is a method of extending the fully homomorphic construction scheme based on LWE to the scheme based on RLWE. The BFV scheme is one of three schemes implemented in Microsoft's Simple Encrypted Arithmetic Library (SEAL), which allows modular arithmetic to be performed on encrypted integers. Another modification to the BFV scheme was proposed by [31]. This scheme studied encoding methods that transform real input data into a polynomial that is an element of the message space of the RLWE scheme. Both [31] and [32] achieved a reduction in error growth compared to the original version of the BFV scheme, and thus, both are able to evaluate circuits with higher multiplicative depth.

In BGV and BFV schemes, each ciphertext contains an error that grows with each homomorphic operation. To avoid decryption failures, the errors must be below a certain threshold. This implies a trade-off between the level of safety and the margin of error affecting the choice of parameters, which is specific to each use case. Such parameter selection requires a sophisticated study of error growth, which has motivated several research efforts [33], [34]. With these optimizations, their BFV variants have better noise growth than BGV for all plaintext moduli. However, their BFV variants are only faster than BGV in small plaintexts, while BGV is still faster in intermediate and large plaintexts.

3) The third generation of FHE: The **GSW** [35] scheme proposes a different approach to perform homomorphic operations, introducing an approximate eigenvector approach that eliminates the requirement for key and modulo switching techniques. This new technique reduces the error growth introduced by homomorphic multiplication to a small polynomial factor. This is a unique aspect relative to previous schemes such as BGV or BFV, for which the final error grows with a quasi-polynomial factor, the RLWE version of which is given by [36].

The main disadvantages of the GSW scheme are high communication cost (ciphertext is large relative to the corresponding plaintext) and computational complexity. To reduce computational overhead, various optimizations have been proposed to improve the bootstrapping process. Specifically, [37] propose a new bootstrap algorithm **AP** that treats decryption as an arithmetic function rather than a Boolean circuit.

Hiromasa et al. optimized the work of Alperin et al. and constructed a FHE scheme [38] that supports homomorphic matrix operations. Ducas and Micciancio proposed a ring variant **FHEW** scheme [39] of the AP bootstrap technique. They introduce a new approach to homomorphic computation. In this work, they also employed a complex FFT, enabling the scheme to achieve the fastest Fourier transform possible. This set of optimizations makes the bootstrapping process of GSW's scheme faster than that of BGV. Afterwards, Chillotti et al. improved the Ducas-Micciancio results and used a different bootstrapping technique, that proposed by Gama, Izabachène, Nguyen, and Xie (GINX), a scheme commonly referred to as **TFHE** [40]. Specifically, for binary keys, TFHE is faster than FHEW, while for higher key sizes, FHEW outperforms TFHE in runtime. In terms of memory, TFHE has a smaller bootstrap key than FHEW.

4) CKKS: **CKKS** [41] is an algorithm for approximate computational homomorphic encryption proposed in recent years. Its specific construction is based on the BGV scheme, but it can also rely on other existing homomorphic schemes. Unlike the previous homomorphic encryption algorithm, where the decryption result is exactly the same as the plaintext, the goal of the CKKS algorithm is to perform approximate calculations. This does not deviate from the requirements, because most operations in real life are faced with real numbers (complex numbers), and operations on real numbers (complex numbers) often only need to retain a part of the effective digits. In addition, the error is allowed and the accuracy limit is relaxed, so that compared with other homomorphic schemes based on the LWE/RLWE problem, CKKS has greatly simplified the details and greatly improved the computational efficiency.

The CKKS scheme includes seven processes of initialization, key generation, encryption, decryption, addition, multiplication, and rescaling. The key steps are the multiplication and rescaling of the two ciphertexts. If the direct product of the ciphertext is to be similar to the product of the plaintext, then the product of the decrypted plaintext must also be similar. Consider first the product of two decrypted values:

$$\begin{aligned} &\mathrm{Dec}(c) \cdot \mathrm{Dec}\left(c'\right) \bmod q \\ &= (c_0 + c_1 \cdot s) \cdot (c'_0 + c'_1 \cdot s) \bmod q \\ &= c_0 \cdot c'_0 + (c_0 \cdot c'_1 + c'_0 \cdot c_1) \cdot s + c_1 \cdot c'_1 \cdot s^2 \bmod q \\ &= d_0 + d_1 \cdot s + d_2 \cdot s^2 \bmod q \end{aligned} \tag{12}$$

It can be seen that if the polynomial is directly used to decrypt, the ciphertext multiplied multiple times requires a higher-order decryption polynomial, and the decryption process cannot be unified. Therefore, CKKS proposes to calculate the auxiliary key evk:

- Given two ciphertexts $c, c' \in \mathcal{R}_q^2$, compute $(d_0, d_1, d_2) = (c_0c'_0, c_0c'_1 + c_1c'_0, c1c'1) \mod q$.
- Output $c_{mult} \leftarrow (d_0, d_1) + \left\lfloor P^{-1} \cdot d_2 \cdot evk \right\rceil \mod q, c_{mult} \in \mathcal{R}_q^2$, where $\lfloor \rceil$ represents the rounding operation.

Then we can see the effect of the patch item:

$$\begin{aligned}&\left\langle P^{-1}\cdot d_2\cdot evk, sk\right\rangle \bmod q\\&=P^{-1}d_2b'+\left(P^{-1}d_2a'\right)\cdot s \bmod q\\&=\left(P^{-1}d_2\left(-a's+e'+Ps^2 \bmod PQ\right)+\left(P^{-1}d_2a's\right)\right) \bmod q\\&=\left(s^2\right)d_2+\left(P^{-1}d_2e'\right) \bmod q\\&\approx s^2\cdot d_2 \bmod q\end{aligned} \tag{13}$$

It can be seen that the result is similar to the quadratic term. In addition, it can be seen that the role of the large number P, d_2e' is a non-negligible number, but by introducing P, the error is reduced.

CKKS multiplication can introduce problems of scale growth and error accumulation. Therefore, a rescaling technique is introduced in CKKS:

- For a given ciphertext $c \in \mathcal{R}_q^2$ and a new modulus $q' < q$, output $c_{rs} \leftarrow \lfloor (q'/q)\cdot c \rceil \mod q', c_{rs} \in \mathcal{R}_{q'}^2$

After rescaling, the approximation error can become a linear growth instead of an exponential growth, which can maintain the accuracy sought. The CKKS algorithm can support a wider range of calculations, adapt to different data sizes and computational complexity, and is of great significance in the field of homomorphic encryption.

C. Federated Learning with Homomorphic Encryption

Federated learning with homomorphic encryption algorithm is a direction of privacy protection in federated learning. Encrypting intermediate parameters or gradients through encryption algorithms can further improve the privacy protection ability of federated learning. The comparison between traditional federated learning and federated learning with homomorphic encryption is shown in Figure 2.

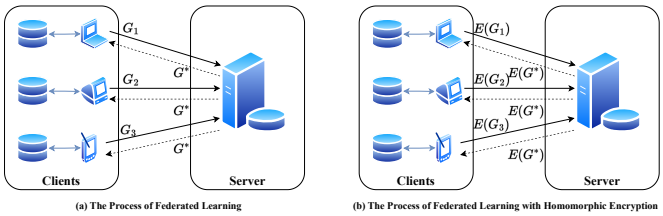


Fig. 2. Comparison between FL and FL with HE. G_1, G_2, G_3 represent for each participants' gradients. G^* represents for the gradients after aggregation. $E(\cdot)$ represents for the Homomorpic Encryption algorithm.

Ref. [42] proposed a deep machine learning scheme using LWE-based additive homomorphic encryption on the gradient to protect privacy, and theoretically analyzed its expansion multiple. Since privacy-preserving deep learning may not be as attractive as ordinary deep learning in terms of accuracy, this article focuses on optimizing the accuracy of encrypted federated learning on classification tasks. However, this solution does not solve the problem of high communication costs.

Ref. [43] proposed a privacy-enhanced data collection scheme for deep learning, using federated learning to reduce the amount of data uploaded to the cloud and protect data privacy. So there are still privacy threats that can happen in federated learning. Ref. [44] proposed an efficient and privacy-enhanced federated learning scheme for industrial artificial intelligence, improving BGV using HE scheme and DP. The scheme is non-interactive within each aggregation and provides high privacy protection levels for both local and shared parameters.

Ref. [45] used Paillier homomorphic encryption [9] in federated learning. Experiments show that the deviation between the model accuracy of PFMLP training and the original model accuracy is less than 1%. However, when the number of encrypted objects is large enough, the Paillier scheme will bring a large communication cost. Considering the computational overhead of homomorphic encryption, the paper uses an improved Paillier algorithm to increase the training speed by 25-28%.

On the basis of FedAvg, [46] proposed a homomorphic encryption federated learning scheme through CKKS. Compared with the Paillier algorithm, the computational efficiency of CKKS has been greatly improved.

Compared with previous work, we propose a novel Personalized Federated Learning (PFL) scheme aimed at addressing these challenges. Our approach splits each user's model into two components: a local low-dimensional feature extraction component, which is regularized to a shared representation, and a high-dimensional feature classification component, which remains local to the user. This unique design allows consistent feature extraction across federation while supporting personalized classification tasks. At the same time, we use CKKS as a homomorphic encryption algorithm to improve the efficiency of federated learning while ensuring accuracy.

III. Methods

A. Methodology

Our proposed methodology is formulated around the optimization objectives and the innovative approach of personalized federated learning. This strategy is meticulously designed to alleviate the inherent issues associated with non-independent and identically distributed (non-iid) data, as well as model heterogeneity inherent to federated learning.

1) Personalized Federated Learning: To address the intricacies of non-iid data and model heterogeneity, we direct our focus towards a paradigm of personalized federated learning. This paradigm bifurcates each user's model into two nuanced components: a localized, low-dimensional feature extraction module, and a high-dimensional feature classification module.

The low-dimensional feature extraction component, while localized, undergoes a regularization towards a communal representation derived from federated amalgamation. This design is calibrated to pinpoint and extract specific features that resonate with local utility. Conversely, the high-dimensional feature classification remains user-centric, facilitating individualized classification undertakings.

The unique aspect of this approach is that while the feature extraction part is local, it is regularized to align with a federated shared representation, thus enabling the extraction of informative features that are consistent across the federation. On the flip side, the feature classification part stays with the

user, enabling them to customize their local model as per the task at hand. Essentially, this strategy fosters a federated learning model that can adapt to different classification tasks, enhancing the personalization of federated learning.

2) Optimization Objective: The optimization objective in our federated learning approach aims to minimize both the composition model loss and the regularization term. This can be formulated as follows:

$$\min_{\omega_k,\rho_k} \mathcal{L}(\omega_k;\rho_k) = \mathcal{L}_{comp}(\omega_k;\rho_k) + \mathcal{R}(\omega_k;\omega_g^*) \tag{14}$$

In this objective function, the composition model loss $\mathcal{L}_{comp}$ is defined as the expected loss over the user-specific data distribution $\mathcal{D}_k$, and is computed as follows:

$$\mathcal{L}_{comp}(\omega_k;\rho_k) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_k}[l(F^k_{\omega_k\diamond\rho_k}(\mathbf{x}),y)] \tag{15}$$

where ω_k and ρ_k represent the feature extraction and feature classification parameters for the k^{th} user respectively, $F^k_{\omega_k\diamond\rho_k}(\mathbf{x})$ denotes the forward pass of the model for the k^{th} user, and $l(\cdot)$ stands for a user-defined criterion.

The regularization term $\mathcal{R}$ is defined as the squared distance between the user-specific feature extraction parameter ω_k and the global feature extraction parameter ω_g^* that minimizes the average composition model loss, scaled by the regularization factor λ:

$$\mathcal{R}(\omega_k;\omega_g^*) = \frac{\lambda}{2}\|\omega_k - \omega_g^*\|^2 \tag{16}$$

where ω_g^* is defined as:

$$\omega_g^* \in \underset{\omega_g}{arg\min} AVG(\{\mathcal{L}_{comp}(\omega_k;\rho_k)\}) \tag{17}$$

The overall goal is to minimize the objective function $\mathcal{L}(\omega_k;\rho_k)$, which balances the trade-off between the locally regularized feature extraction and the personalized feature classification, aiming to improve the overall performance and utility of the federated learning system.

3) Privacy Preservation with CKKS Homomorphic Encryption Scheme: To fortify user privacy and preemptively obstruct potential exfiltration of federated learning models by servers, we have incorporated the CKKS (Cheon-Kim-Kim-Song) homomorphic encryption scheme. The CKKS scheme stands out as a proficient homomorphic encryption method, facilitating computations on encrypted data without the requisite of prior decryption. Its innate capability to seamlessly handle intricate operations involving real or complex numbers earmarks it as the preferred choice for federated learning contexts, wherein model parameter augmentations inherently demand such computations.

Within our framework, user-side locally trained models are encrypted utilizing the CKKS scheme. The ensuing homomorphically encrypted ciphertext is subsequently relayed to the centralized server. Once the server accrues homomorphic ciphertexts from the cohort of users, a methodical aggregation is executed. The aggregated product is then disseminated back to the user nodes.

On acquisition of this aggregated ciphered output, users employ their private decryption keys to retrieve the plaintext global model. This global blueprint, in tandem with our federated learning paradigm, orchestrates the training of the local model. This cyclical sequence perpetuates, concurrently guaranteeing optimal model refinement and staunch preservation of user privacy.

Integrating CKKS into our architecture infuses a robust layer of privacy preservation, aptly addressing the escalating imperatives for secure, confidential machine learning infrastructures. This synergistic amalgamation harnesses the protective prowess of homomorphic encryption with the optimization assets inherent to our federated learning approach.

4) Algorithm: Our federated learning approach can be formulated into an algorithm that iteratively updates the global and local models at each iteration using gradient descent. The detailed algorithm is as Algorithm 1.

Algorithm 1 Privacy-Preserving Federated Learning Optimization with CKKS

Require: Initial global model parameters ω_g^0 and client-specific model parameters ω_k^0, ρ_k^0, learning rate η, number of iterations T, and regularization parameter λ.

for t in $1, 2, \cdots, T$ **do**
  Server randomly samples n clients $\mathcal{I}^t$.
  for k in $\mathcal{I}^t$ in parallel **do**
    # Encrypt locally trained model with CKKS
    $E_k = \text{CKKS.Encrypt}(\omega_k^t, \rho_k^t)$
    # Send encrypted model to server
    Send(E_k)
  end for
  # Server aggregates received encrypted models
  $E_g = \frac{1}{n}\sum_{k\in\mathcal{I}^t} E_k$
  # Server sends aggregated model back to clients
  Send(E_g)
  for k in $\mathcal{I}^t$ in parallel **do**
    # Decrypt aggregated model
    $(\omega_g^{k,t}, \rho_g^{k,t}) = \text{CKKS.Decrypt}(E_g)$
    # Update global model $\omega_g^{k,t}$
    $\omega_g^{k,t} = \omega_g^{k-1,t} - \eta \cdot \nabla\mathcal{L}_{comp}^{(\omega_g^{k-1,t})}(\omega_g^{k-1,t}; \rho_k^{k-1,t})$
    # Update local model ω_k^t, ρ_k^t
    $\omega_k^{t+1} = \omega_k^t - \eta\cdot[\nabla\mathcal{L}_{comp}^{(\omega_k^t)}(\omega_k^t;\rho_k^t) + \lambda\cdot\nabla\mathcal{R}(\omega_k^t;\omega_g^{k,t})]$
    $\rho_k^{t+1} = \rho_k^t - \eta\cdot\nabla\mathcal{L}_{comp}^{(\rho_k^t)}(\omega_k^t;\rho_k^t)$
  end for
  Update the global model $\omega_g^{t+1} = \frac{1}{n}\sum_{k\in\mathcal{I}^t}\omega_g^{k,t}$
end for

Ensure: The final global model parameters ω_g^T and client-specific model parameters ω_k^T, ρ_k^T.

Our algorithm has been meticulously tailored to enhance privacy within federated learning, employing the CKKS encryption scheme for this purpose. The algorithm's inputs encompass the initial global model parameters, client-specific model parameters, learning rate, iteration count, and regularization constants. The primary objective of this algorithm is to securely encrypt, transmit, and aggregate model parameters across the server-client nexus, ensuring the utmost privacy of data.

The algorithm proceeds through the following stages:

1) **Client Selection:** At the onset of each iteration, the server randomly selects a subset of clients to participate.

2) **Local Training and Encryption:** For each chosen client, model parameters derived from local training undergo encryption leveraging the CKKS scheme. Subsequently, these encrypted parameters are relayed to the server.
3) **Server-side Aggregation:** Upon receiving the encrypted parameters from the clients, the server executes an aggregation function, producing a singular set of aggregated model parameters. This consolidated parameter set is then dispatched to the clients.
4) **Decryption and Model Updates:** Clients, upon receipt of the aggregated parameters, deploy decryption. Post-decryption, clients update both global and local model parameters in line with the gradient data.
5) **Global Model Parameter Consolidation:** Concluding each iteration, the server calculates the mean of the client's global model parameters to derive the most recent global parameters.

Upon completion, the algorithm yields optimized global model parameters as well as client-specific parameters. Achieved through privacy-centric federated learning, this ensures data privacy while procuring an enhanced global model.

In the proposed algorithmic framework, during each iteration, the server stochastically samples a subset of n clients. For every chosen client, the server embarks on an initial step to refresh the global model parameters. This is accomplished by evaluating the gradient of the composite model loss in relation to these global parameters. Subsequently, the client-specific parameters undergo updates based on the gradients derived from both the composite model loss and the regularization term, in reference to their own unique parameters. Concluding this sequence, the server amalgamates the refreshed global model parameters sourced from the curated selection of clients, effectuating an updated global model. This procedural cadence is maintained across a predetermined span of iterations T. The algorithm's terminal deliverables comprise the meticulously optimized global model parameters alongside the distinct parameters specific to each client.

To elaborate on the structural intricacies of our methodology: every user's model is bifurcated into a pair of distinct components. The first is a local, lower-dimensional feature extraction component, which is meticulously regularized in alignment with a universally shared representation. The second component remains anchored in a higher-dimensional space, dedicated to feature classification, and is retained exclusively at the client-side. This avant-garde configuration engenders a consistent feature extraction matrix across federated domains while concurrently endorsing bespoke classification endeavors at an individualized scale.

B. Datasets, Baselines and Implementation Details

1) Datasets: We carried out experimental studies on two widely used datasets: MNIST [47] and CIFAR-10 [48], both of which are canonical in the realm of machine learning and computer vision.

1) MNIST: The MNIST dataset contains images of handwritten digits from the National Institute of Standards and Technology. It consists of 60,000 training samples and 10,000 testing samples. Each sample is a 28x28 pixel grayscale image representing a single handwritten digit between 0 and 9. This dataset is one of the benchmarks in the machine learning community and is widely used for image classification, pattern recognition, and digit recognition tasks. Its relatively small size makes it ideal for rapid prototyping and algorithm verification.

2) CIFAR-10: The CIFAR-10 dataset was created by the Canadian Institute for Advanced Research. The dataset contains 50,000 training samples and 10,000 testing samples, covering 10 different categories, each category has 5,000 image samples. The images are presented as 32x32 color (RGB) images and cover common objects such as airplanes, cars, birds, cats, deer, dogs, frogs, horses, boats, and trucks. The CIFAR-10 dataset is widely used in algorithm development and performance evaluation in areas such as image classification, object recognition, and image generation.

Our selection of these datasets was predicated on several grounds. Firstly, their prominence and recurrent adoption in the machine learning sphere underscore their credibility. Both MNIST and CIFAR-10 introduce a degree of complexity and diversity, thereby establishing benchmarks for typical image classification conundrums. Secondly, their manageability in terms of size facilitates expeditious experimentation and model validation. Lastly, given their entrenched status in computer vision research, they provide a standardized platform, thereby promoting consistent comparisons and facilitating algorithmic replication across disparate studies.

2) Baselines: We compared our personalized federated learning scheme against four commonly used federated learning algorithms: FedAvg [2], Ditto [17], Scaffold [15], and FedProx [16] mentioned in Section 2.1. These algorithms were selected as baselines due to their representative nature in the field of federated learning. Specifically, we chose to utilize the function library from FedLab [1], which is implemented based on PyTorch, to implement these baseline methods. In order to better compare the model effects, we compared the above federated learning with our model based on MLP and CNN respectively.

3) Metrics: The main metric used to evaluate the performance of the different algorithms is the model's test accuracy on each dataset. Greater accuracy indicates a better-performing model. We also showed the Precision, Recall and F1 Score of the model during the ablation experiment to further measure the classification ability of the model for different categories.

ACC, Precision, Recall, and F1 are commonly used performance metrics to evaluate classification models.

- **ACC:** Accuracy (ACC) measures the overall correctness of the model's predictions. It is calculated as the ratio of correctly classified instances (both positive and negative) to the total number of instances.

$$ACC = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \tag{18}$$

where TP(True Positive) represents the number of correctly classified positive instances, TN(True Negative) represents the number of correctly classified negative

[1] https://github.com/SMILELab-FL/FedLab

instances, FP(False Positive) represents the number of incorrectly classified positive instances, and FN(False Negative) represents the number of incorrectly classified negative instances.

- **Precision:** Precision quantifies the proportion of correctly predicted positive instances out of all instances that are predicted as positive. It focuses on the correctness of positive predictions and is computed as the ratio of true positives to the sum of true positives and false positives.

$$Precision = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{19}$$

where True Positive and False Positive have the same definitions as mentioned above.

- **Recall:** Recall (also known as Sensitivity or True Positive Rate) calculates the proportion of correctly predicted positive instances out of all actual positive instances. It evaluates the model's ability to identify positive instances and is calculated as the ratio of true positives to the sum of true positives and false negatives.

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{20}$$

where True Positive and False Negative have the same definitions as mentioned above.

- **F1 Score:** F1 Score combines precision and recall into a single metric. It provides a balanced measure of a model's performance by taking into account both precision and recall. The F1 score is the harmonic mean of precision and recall and is given by the formula:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{21}$$

where Precision and Recall have the same definitions as mentioned above.

4) Experiments Settings: For the MNIST dataset, our implementation uses two architectures:

1) **MLP:** A multi-layer perceptron model comprising three hidden layers followed by an activation layer.
2) **CNN:** A basic convolutional neural network consisting of two convolutional layers, a single pooling layer, two fully connected layers, and an activation layer.

Given the intricacies of the CIFAR-10 dataset, our experimental design leverages both basic CNN architectures and the more sophisticated AlexNet [49], introduced in 2012 by Krizhevsky, Sutskever, and Hinton, revolutionized deep learning and computer vision with its innovative design. Characterized by its deep eight-layer structure, it introduced the ReLU activation function, dropout for regularization, and overlapping max-pooling. Additionally, AlexNet employed data augmentation strategies for robustness, harnessed dual-GPU training for enhanced computation, applied Local Response Normalization (LRN) for improved layer responses, and advocated for end-to-end training.. Notably, the models are initialized randomly, without any pre-training.

For the encryption processes, we employ Pyfhel, a Python Fully Homomorphic Encryption Library. To bolster computational efficiency, our method integrates the CKKS encryption scheme. The settings for the CKKS scheme include a polynomial degree of 2^{14}, a scale parameter of 2^{30}, and polynomial coefficient digit specifications of $[60, 30, 30, 30, 60]$.

Our entire algorithmic framework is built using Python 3.9 and PyTorch 1.2.1. The training phase leverages stochastic gradient descent (SGD) as the optimization technique. In the context of our federated learning scheme, the last linear layer in both the MLP and CNN is designated for local gradient feedback. In contrast, the gradients of other layers are updated post-global aggregation, which enhances the provision of localized personalization. We employ cross-entropy loss for model optimization, with a learning rate of 1e-2. The training regimen spans 100 epochs.

IV. Results

A. Comparison with Previous Studies & Selection of Hyper-parameters

In our quest to emulate non-IID data scenarios typical of federated learning, we employed a Dirichlet distribution, parameterized by α, to craft skewed data distributions amongst users. It is crucial to note that a diminutive α value signifies an augmented degree of non-IIDness, thereby intensifying the complexity for federated learning algorithms.

Table I elucidates the comparative efficacy of various algorithms across datasets under disparate non-IID magnitudes. A perspicuous trend emerges from the collated data: our scheme consistently outperforms the baseline algorithms on both datasets, across all levels of non-IIDness.

In particular, the superiority of our model becomes increasingly conspicuous as data gravitates towards heightened non-IID characteristics (denoted by diminished α values). This resiliency under duress underscores our model's dexterity in navigating these rigorous terrains. Such proficiency emanates from our model's architectural blueprint: the lower-dimensional feature extraction mechanism is strategically oriented towards a communal representation, facilitating the discernment and appropriation of salient features consistent throughout the federation, even under pronounced non-IID constraints.

In tandem, the local higher-dimensional feature classification part of our scheme allows for individualized classification tasks, which further enhances the adaptability and robustness of our scheme under non-IID conditions. This unique dual-component design of our scheme, which combines federated shared representation with local feature classification, is the key to its superior performance over the baseline algorithms.

Additionally, upon transitioning our classification benchmark from MLP to CNN, our framework consistently maintains its superior performance relative to all baseline models. This robustness underscores that our personalized federated learning algorithm exhibits model-agnostic properties, ensuring optimal results across diverse foundational models.

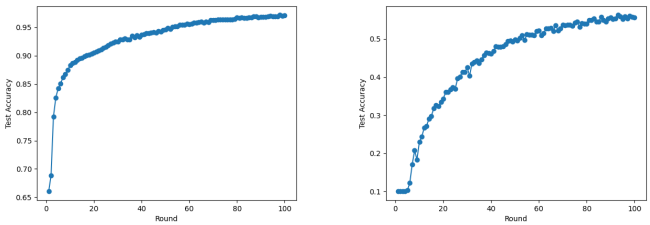
In summation, our empirical analyses unequivocally validate the proficiency of our personalized federated learning framework in adeptly managing non-IID data and inherent model heterogeneity. These experimental results robustly affirm the architectural integrity and distinctive merits of our proposed paradigm.

TABLE I
PERFORMANCE OF VARIOUS ALGORITHMS ON DIFFERENT DATASETS WITH VARYING NON-IID LEVEL

Dataset	Baseline	Algorithms					Non-IID level (a value)
		FedAvg	Ditto	Scaffold	FedProx	Myscheme	
MNIST	MLP	90.31	91.55	91.84	92.06	**92.37**	0.5
		88.84	90.60	90.85	91.21	**91.58**	0.1
		86.94	89.57	89.35	90.15	**90.25**	0.05
	CNN	96.12	97.45	97.55	97.60	**97.92**	0.5
		93.84	96.32	96.02	96.06	**97.62**	0.1
		91.24	94.23	95.33	95.37	**97.01**	0.05
CIFAR-10	CNN	54.20	56.82	57.36	57.75	**58.26**	0.5
		48.83	52.35	52.10	55.94	**56.90**	0.1
		42.55	50.66	51.38	52.05	**52.67**	0.05
	AlexNet	58.37	62.12	62.32	62.48	**65.22**	0.5
		52.60	58.45	58.62	58.76	**61.64**	0.1
		45.89	55.75	55.92	56.00	**59.00**	0.05

B. Convergence Analysis

We rigorously examined the convergence behavior of our algorithm utilizing the CIFAR-10 and MNIST datasets across a span of 100 training iterations, particularly emphasizing high non-IID conditions. Corresponding visualizations are presented in Figure 3a (MNIST) and Figure 3b (CIFAR-10).



(a) MNIST (b) CIFAR-10

Fig. 3. Test accuracy over 100 training rounds on MNIST and CIFAR-10 datasets.

For CIFAR-10, our model's test accuracy started from a low value of 0.1001 and gradually improved to a final accuracy of 0.5566 after 100 rounds of training, indicating a steady convergence. The increase in test accuracy was relatively slow in the initial stages (rounds 1-20), but it accelerated during the middle stages (rounds 21-60), and then slowed down again towards the end (rounds 61-100). This is a typical convergence pattern, where the model makes significant improvements in the middle stages after warming up in the initial stages, and then cools down towards its optimal solution in the final stages.

Contrastingly, for the MNIST dataset, our model's test accuracy started at a higher value of 0.6601 and improved to 0.9712 after 100 rounds of training. Unlike the CIFAR-10 results, the convergence on MNIST was quite rapid in the initial stages and remained relatively stable throughout the training process, reflecting the simpler and less diverse nature of the MNIST dataset compared to CIFAR-10.

The difference in the convergence behaviour between the two datasets can be attributed to the difference in their complexity and diversity, with CIFAR-10 being a more challenging dataset due to its color images and more diverse classes. In scenarios of pronounced non-IID conditions, CIFAR-10's learning trajectory is further complicated by potential imbalances and variances in client data distributions.

Nevertheless, notwithstanding these inherent challenges, our proposed model exhibited unwavering and methodical convergence across both datasets. Such robustness attests to its adaptability across a spectrum of learning environments, encapsulating marked non-IID scenarios.

To encapsulate, our algorithm manifests solid convergence metrics across both MNIST and CIFAR-10, registering consistent enhancements in test accuracy throughout training phases. This unequivocally substantiates the prowess of our model across diverse learning milieu, particularly in challenging non-IID contexts, emphasizing its convergent reliability.

C. Ablation Studies

We present the results of ablation studies to investigate the contribution of each component in our model. We choose Simple CNN as the baseline model and perform ablation experiments with different versions of the model as follows:

- **Method (a)**: This version does not set up Personalized Learning. That is, all global models and local models update all parameters.
- **Method (b)**: This version does not set the regularization term. That is, the regularization formula is removed from the optimization objective.
- **Method (c)**: This version is not encrypted by CKKS. The aim is to verify the effect of CKKS on performance.
- **Ours**: This version is the entire model we proposed.

Specifically, for Method (a), both client and server execute concurrent parameter updates during the implementation phase. This synchronicity is established by judiciously manipulating the filters associated with the stochastic gradient descent (SGD) optimizer in the underlying codebase. For Method (b), during the implementation, the regularization coefficient, denoted as Lambda (λ), is precisely set to zero within the loss function. This configuration ensures the model operates without any regularization constraints. For Method

TABLE II
THE RESULTS OF ABLATION STUDIES ON TWO DATASETS. CKKS REPRESENTS WHETHER TO PERFORM CKKS HOMOMORPHIC ENCRYPTION. PERSONALIZATION REPRESENTS WHETHER TO PERSONALIZE. REGULARIZATION REPRESENTS WHETHER TO PERFORM REGULARIZATION. THE NON-IID LEVEL IS SET TO 0.5.

Dataset	Methods	Personalization	Regularization	CKKS	ACC	Precision	Recall	F1 Score
MNIST	(a)	-	✓	✓	95.37	94.65	95.20	94.93
	(b)	✓	-	✓	94.95	95.80	95.33	95.57
	(c)	✓	✓	-	97.88	97.25	**98.17**	97.78
	Ours	✓	✓	✓	**97.92**	**98.55**	96.29	**97.86**
CIFAR-10	(a)	-	✓	✓	53.82	50.25	55.59	52.30
	(b)	✓	-	✓	54.80	56.76	54.56	55.28
	(c)	✓	✓	-	**58.56**	58.75	56.32	57.36
	Ours	✓	✓	✓	58.26	**59.36**	**56.70**	**58.54**

(c), Contrary to employing any homomorphic encryption strategies, this method opts for transparent model parameter management. As such, all parameters remain unencrypted and are directly communicated between the server and the client. Each method elucidates distinct facets of our model's behavior and all model parameters are directly exposed and transmitted between the server and the client.

In comparative experiments, we control the chosen method through command-line parameters. Different combinations also correspond to different schemes in ablation experiments. We trained for 100 epochs until the model converged. When the model converges, record the results of the model on the test set. The results of the comparison between the different methods are shown in Table II.

Impact of Personalization. A comparative analysis between our method and Method (a) elucidates a pronounced improvement across both datasets, underscoring the efficacy of our personalized federated learning approach in addressing non-IID data challenges. This enhancement is consistently observed across both datasets.

Influence of Regularization. A juxtaposition of our method with Method (b) reveals discernible advancements in our approach across both datasets. Specifically, metrics such as ACC and F1 showcase the most pronounced augmentations. This underscores the potency of the regularization term integrated into our objective function in ameliorating the classification performance, especially in the context of non-IID data distributions. Analogously, this result is uniformly witnessed across both datasets.

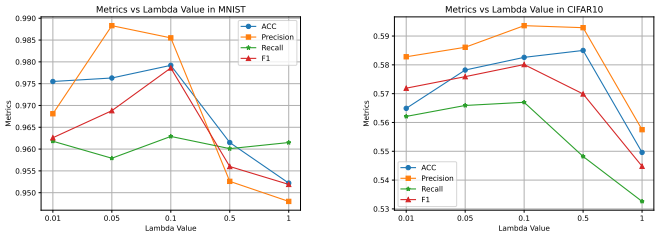
Effect of CKKS. By comparing ours with Method (c), we can see that the effects of the two are similar. On the MNIST dataset, the recall of Method (c) is higher, while on the CIFAR-10 dataset, the acc of Method (c) is higher. This verifies that although CKKS is an approximate homomorphic encryption method, in real scenarios, such errors are almost difficult to detect.

In summary, through meticulous ablation experiments, we systematically delineate the composite influence of the Personalization, Regularization, and CKKS modules within our proposed methodology. The empirical outcomes unequivocally substantiate the logical coherence and effectiveness of our proposed approach.

D. Hyperparameter Influence on Model Performance

To scrutinize the impact of hyperparameters on model the performance, we elected to vary pivotal hyperparameters intrinsic to our method. Specifically, we considered the weight, denoted as λ, of the regularization term. By tuning different values of λ, we aimed to discern its influence on the model's accuracy across two datasets, facilitating an informed decision regarding its optimal value.

Our hyperparameter tuning experiments spanned two datasets: MNIST and CIFAR-10. With the non-IID data level fixed at 0.5, we evaluated several potential λ values, including [0.01, 0.05, 0.1, 0.5, 1]. As a baseline model, we leveraged a simple CNN, chosen due to its impressive performance combined with computational efficiency. Each experiment extended over 100 epochs, culminating once the model converged. Upon convergence, outcomes were noted from the test set, with salient results depicted in Figure 4.



(a) MNIST (b) CIFAR-10

Fig. 4. Metrics for different λ on MNIST and CIFAR-10 datasets. The metrics shown in the figure include ACC, Precision, Recall and F1 Score.

The experimental results attest to the significant influence of the hyperparameter λ on the model's ultimate accuracy. A discernible trend emerges: as λ escalates, the performance metrics generally rise before eventually tapering off. Referencing Figure 4a, optimal values for ACC, Recall, and F1 Score are observed at $\lambda = 0.1$, while Precision peaks at $\lambda = 0.05$. A perceptible decline in all evaluation metrics is evident when λ transitions from 0.1 to 0.5. In contrast, Figure 4b delineates a scenario where Precision, Recall, and F1 Score are maximized at $\lambda = 0.1$, while ACC attains its peak at $\lambda = 0.5$.

Analytically, lower values of λ can predispose the model to overfitting, especially when grappling with skewed datasets.

Conversely, an excessively large λ can hinder the model's training process, impeding its trajectory towards the global optimum.

E. Comparison of computation cost between CKKS and Paillier scheme

To assess the enhanced efficiency of the CKKS encryption method, we juxtapose it against the Paillier scheme [9]. Introduced by Pascal Paillier in 1999, the Paillier homomorphic encryption is a public-key system underpinned by additive homomorphic properties. This facilitates arithmetic on ciphertexts without compromising the confidentiality of the underlying plaintext.

The core components of the Paillier encryption schema encompass key generation, encryption, decryption, and its homomorphic characteristics, among others. A salient feature of the Paillier scheme is its ability to perform addition on ciphertexts without revealing the plaintext, making it particularly advantageous for safeguarding data privacy and conducting secure computations. Nonetheless, the Paillier method tends to exhibit suboptimal performance metrics. Specifically, its encryption and decryption procedures are computationally intensive, leading to extended processing times. Consequently, real-world applications necessitate a judicious evaluation, balancing security against efficiency, to determine the most suitable encryption strategy.

To substantiate the efficiency claims surrounding the CKKS encryption method, we juxtaposed its performance against the widely-recognized Paillier scheme. Our comparison matrix encompassed several key metrics, including:

- Final accuracy rate.
- Time consumption for addition operations.
- Time taken for the encryption process.
- Decryption time consumption.

For robustness, experiments spanned across 100 iterations, post which an average time consumption metric was derived. The outcomes of this comparative analysis are delineated in Table III.

TABLE III
COMPARISON OF COMPUTATION COST BETWEEN CKKS SCHEME AND PAILLIER SCHEME ON MNIST.

Scheme	Addition	Encryption	Decryption	ACC
CKKS	0.059s	0.011s	0.004s	97.92
Paillier	15.526s	63.183s	22.390s	98.15

From the tabulated results III, we observe discernible differences in the performance metrics of the CKKS and Paillier schemes. Specifically, the CKKS scheme demonstrates shorter durations for addition, encryption, and decryption operations when compared to the Paillier counterpart. While there is a marginal trade-off in accuracy for CKKS, the substantial savings in computational time, facilitated by approximate calculations, accentuate its efficacy. Below, we delineate the key conclusions:

- **Temporal Efficiency:** The data underscores a superior computational efficiency of the CKKS scheme. Its expedited performance in addition, encryption, and decryption processes vis-à-vis the Paillier scheme suggests that for equivalent computational tasks, CKKS completes operations more promptly.
- **Accuracy Discrepancy:** The CKKS scheme recorded an accuracy of 97.92%, a slight diminution from the 98.15% accuracy registered by the Paillier scheme. Given the considerable reduction in computational time offered by the CKKS through its approximative methods, such a diminutive compromise in accuracy can be deemed tolerable.
- **Advantages of the CKKS Scheme:** CKKS's approximate computations significantly curtail the temporal demands of homomorphic encryption, all the while preserving a commendable degree of result accuracy. Furthermore, its rapid addition operations resonate with applications demanding expeditious computations—this could be pivotal for large-scale data analytics and safeguarding privacy in cloud-based systems.

In essence, compared against the Paillier scheme, the CKKS method manifests pronounced temporal efficiencies, accomplishing operations more swiftly, albeit with a slight dip in accuracy. Such characteristics earmark the CKKS framework for prospective applications in voluminous data processing and contexts necessitating rigorous privacy assurances.

V. DISCUSSION

A. Discussion on our Scheme

1) Main Claims: In this project research endeavor, we introduce an innovative personalized federated learning framework, which harnesses the power of homomorphic encryption, specifically the CKKS approximate homomorphic encryption algorithm. The salient features and claims of our proposed methodology are:

- **Improved Privacy Safeguards.**By leveraging advanced homomorphic encryption techniques, the proposed algorithm ensures the utmost confidentiality of user data during the training phase. By initiating the encryption of data prior to its transmission, we effectively mitigate the risks associated with unauthorized data breaches, thereby ensuring the protection of sensitive individual details or proprietary business intelligence.
- **Enhanced Model Personalization.** Recognizing the challenges posed by non-IID data distributions, our approach tailors model adaptation in a two-pronged manner. First, low-dimensional structural features are updated at the global server level. Second, high-dimensional abstract features undergo refinement at the localized client end. This dual process, coupled with deliberate gradient update delays, ensures the model aligns more harmoniously with the inherent distribution of client data during the classification phase, with overall parameter updates consolidated server-side.
- **Optimized Communication and Computation.** A well-documented drawback of homomorphic encryption is its

significant computational overhead. Mindful of this, our choice of the CKKS algorithm prioritizes both communication efficiency during federated learning and computational accuracy, striking a balance that advances the practical applicability of our approach.

In essence, this research embodies a forward-thinking approach to federated learning, intertwining robust data protection, astute model personalization, and resource-efficient computations. It offers a promising pathway for real-world applications where data privacy and model efficacy are paramount.

2) Effectiveness: In Section 4.1, we delineate a comprehensive comparison between our proposed model and preceding methodologies across diverse scenarios. The empirical evidence, as showcased in the provided table, unequivocally underscores the superior efficacy of our framework: it consistently outshines previous models across all tested conditions. Notably, as datasets exhibit heightened non-IID characteristics (evident through smaller α values), the relative performance of our model becomes increasingly conspicuous. This robustness in challenging scenarios emanates from our model's strategic design. Specifically, the scheme's lower-dimensional feature extraction component is meticulously regularized towards a shared representation. This design facet ensures consistent and salient feature extraction throughout the federation, even amidst pronounced non-IID conditions. To elucidate, consider the classification task on the MNIST dataset employing a rudimentary CNN as the baseline model, under a non-IID level of 0.05, our model manifests an impressive accuracy rate of 90.25%—a substantial enhancement of 3.31% over the FedAvg model.

3) Robustness: Furthermore, we undertook an exhaustive evaluation of our framework's responsiveness to varied hyperparameter settings, specifically adjusting the non-IID levels. Simultaneously, cross-evaluations were conducted juxtaposing two archetypal baseline models, MLP and CNN, against our advanced personalized federated learning algorithm. Impressively, across diverse hyperparameter configurations and foundational models, our approach consistently surpassed previous benchmarks, underscoring its intrinsic robustness and adaptability.

Equally compelling, parallel experiments conducted on both the MNIST and CIFAR-10 datasets reaffirmed our algorithm's supremacy. Such congruency in performance across these datasets further attests to the algorithm's broad generalization capabilities. Regardless of the choice of baseline model—be it MLP, a rudimentary CNN, or the more intricate AlexNet—our scheme consistently emerged preeminent.

In summary, these empirical validations conclusively assert our methodology's formidable stability and resilience across an array of baseline models and heterogeneous data distributions.

4) Reasonableness: In Section 4.2, we delve into an intricate analysis of model convergence. A meticulous examination of the ACC curves across both datasets reveals a compelling trend: as the number of epochs escalates, the model's accuracy rate incrementally ascends, eventually stabilizing. Such a trajectory unequivocally demonstrates the efficient convergence of our framework. Furthermore, it substantiates the algorithm's aptitude in adeptly addressing gradient update and aggregation challenges endemic to federated learning.

Subsequently, in Section 4.3, we employ ablation experiments as a rigorous investigative tool to elucidate the influence of individual model components on resultant outcomes. Systematic ablation of the three pivotal modules—Personalization, Regularization, and CKKS—affords a comprehensive insight into their respective contributions. This rigorous examination not only augments the scheme's interpretability but also serves as an empirical validation of the model's structural soundness and the judiciousness of its design choices.

5) Privacy and Efficiency: Within our proposed framework (myscheme), we harness the capabilities of the homomorphic encryption algorithm, ensuring data remains encrypted both during transit and processing. We operate under the widely accepted assumption that the server is 'honest-but-curious', implying that, while it faithfully follows the protocol, there is potential for it to infer information from the uploaded models.

In our architecture, each participant conducts model training locally. Subsequent to this, the model parameters are encrypted using the CKKS scheme prior to transmission to the server. Consequently, the server exclusively receives ciphertexts from participants, limiting its role to performing computations upon these encrypted entities. The aggregated output remains encrypted, which the server dispatches to the participants. Participants then decrypt this output, integrating it into their ensuing local model training iteration.

At no juncture does the server possess the capability to deduce information from these ciphertexts, barring successful cryptographic breaches. It is pertinent to note that prevailing quantum attack methodologies target integer factorization. As of the current state of the art, there are no known quantum attack strategies effective against lattice-based structures. Given that CKKS is rooted in lattice-based homomorphic encryption, it presents a level of resistance to quantum threats.

In summation, our framework is robustly equipped to counter potential inferences by 'honest-but-curious' servers and exhibits characteristics indicative of quantum resistance.

Section 4.4 delves into a comprehensive discourse on model efficiency. Through rigorous experimentation, we juxtapose the performance metrics, particularly the encryption, decryption, and addition times, of both the CKKS and the Paillier encryption schemes. Empirical results suggest that the CKKS homomorphic encryption significantly bolsters our model's operational efficiency with only a marginal compromise on accuracy. Such findings lend credence to our assertion that the federated learning scheme we propose is not only academically robust but also holds immense potential for broader industrial applications.

B. Limitations

While our research delineates a remarkably effective federated learning model, it is essential to acknowledge certain constraints inherent to our approach, mirroring challenges prevalent in many federated learning paradigms.

- **Security Assumptions:** Our model fundamentally rests on the assumption that all participating entities maintain

honest postures. We postulate an "honest-but-curious" server model, implying that while the server will not deviate from the protocol, it might attempt to glean information. Consequently, our scheme might be susceptible to collusion attacks orchestrated between the server and a malicious participant.

- **Experimental Limitations:** Our experimental setup predominantly emulated multiple clients on a singular server. Such a simulation might not capture the nuances of real-world applications, where federated learning efficiency is intrinsically linked to inter-machine communication capabilities. This pivotal dimension remains unexplored in our current analysis.
- **Scope of Experimentation:** Our empirical evaluations, albeit robust, were primarily anchored on relatively elementary datasets like MNIST and CIFAR-10. When transposed to domains characterized by vast data quantities and pronounced distribution disparities—such as healthcare, finance, and autonomous navigation—the algorithm's effectiveness warrants further empirical scrutiny.

Moving forward, our research trajectory will encompass a broader spectrum of experimental validations. Our goal is to fortify the evidence supporting the generalized applicability and industrial potential of our proposed personalized federated learning algorithm.

VI. Conclusions

A. Summary of Achievements

Federated learning is an important research direction for industrial applications of machine learning. Federated learning can improve the privacy of data and reduce the computing load of the central server. What federated learning needs to face is the non-independent and identical distribution of data and the privacy protection of data transmission. Therefore, this project proposes a federated learning scheme with privacy protection to solve non-iid data and model heterogeneity. Specifically, the scheme splits each user's model into two components: a local low-dimensional feature extraction component, which is regularized to a shared representation, and a high-dimensional feature classification component, which remains local to the user. This unique design allows consistent feature extraction across federation while supporting personalized classification tasks. At the same time, we use CKKS as a homomorphic encryption algorithm to improve the efficiency of federated learning while ensuring accuracy.

Our investigative endeavors encompassed an extensive experimental validation. Initially, a juxtaposition with extant methodologies was undertaken, encompassing varying non-IID data levels and leveraging two benchmark models, namely MLP and CNN. This exhaustive analysis, across diverse scenarios, substantiated the supremacy of our approach, rendering unparalleled classification accuracy, thus reaffirming its avant-garde status amongst established algorithms. Such outcomes underscore our framework's prowess in adeptly managing non-IID data intricacies and model diversity.

Furthermore, convergence analysis was pivotal, elucidating the model's ability to seamlessly transfer gradients and attain convergence amidst recurrent client-server interactions. This convergence scrutiny substantiates the model's operational viability, offering a lucid depiction of its convergence trajectory.

Ablation studies, a cornerstone of our research, were executed to discern the efficacy of individual components, especially emphasizing the personalization and regularization aspects. Analyses spanning two datasets unequivocally confirmed the cardinal role of our tailored personalization strategy and regularization entities in accentuating classification outcomes.

Additionally, the encryption efficiency took center stage, as the merits of the CKKS algorithm were showcased. By juxtaposing it against the Paillier framework, we ascertained CKKS's capability in significantly enhancing federated learning's efficiency, with negligible accuracy trade-offs.

In synthesis, through a holistic methodological exposition and rigorous experimental validation, we unerringly address the dual objectives delineated in Section 1.3. Primarily, we instantiate a sophisticated federated learning algorithm adept at navigating the labyrinthine terrains of distributed data's non-IID nature and the multifariousness of models. Subsequently, through the integration of a robust homomorphic encryption modality (CKKS), we ensure an impervious privacy shield for federated learning's intermediary parameters.

B. Gains and Future Work

Throughout this research endeavor, I embarked on an intricate exploration into the multifaceted realms of Federated Learning and Homomorphic Encryption. The intricate complexities and emerging challenges intrinsic to these domains necessitated a comprehensive understanding, particularly against the backdrop of current predicaments within these spheres. Motivated by these challenges, I conceptualized and advanced a pioneering solution tailored to adeptly address them. The rigorous design, execution, and analytical review of numerous experiments not only reinforced the robustness of my proposed methodology but also provided robust empirical validation. Beyond the technical depth, this research venture enhanced my acumen in the scholarly craft of thesis formulation, fortifying my expertise in machine learning and cryptography.

Upon introspection of this research trajectory, I discern several prospects for augmentation and further inquiry:

- **Continual Exploration and Consolidation.** Although the present study has substantially enriched my comprehension of the disciplines, Federated Learning and Homomorphic Encryption persistently evolve, warranting perpetual engagement and knowledge assimilation.
- **Expanding Experimental Horizons.** A salient constraint encountered during this study was the encapsulation of federated learning simulations within an isolated computational environment. Moving forward, there is a pressing imperative to transcend this limitation by orchestrating federated learning across a diverse array of devices, mirroring a more authentic and extensive ecosystem.

- **Application in Practical Domains.** The potential of federated learning extends profoundly into real-world domains. Paramount sectors such as medical diagnostics, autonomous transportation, and fiscal analytics can immensely benefit. However, the intricacies of these real-world challenges demand bespoke algorithmic adaptations and enhancements to ensure the relevance and efficacy of federated learning techniques.

To encapsulate, this research represents not merely an academic culmination but a prologue to an ongoing odyssey towards deeper scholarly introspection, expansive practical implementations, and seminal advancements in Federated Learning and Homomorphic Encryption.

REFERENCES

[1] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[3] C. Song, T. Ristenpart, and V. Shmatikov, "Machine learning models that remember too much," in *Proceedings of the 2017 ACM SIGSAC Conference on computer and communications security*, 2017, pp. 587–601.

[4] K. V. Sarma, S. Harmon, T. Sanford, H. R. Roth, Z. Xu, J. Tetreault, D. Xu, M. G. Flores, A. G. Raman, R. Kulkarni *et al.*, "Federated learning improves site performance in multicenter deep learning without data sharing," *Journal of the American Medical Informatics Association*, vol. 28, no. 6, pp. 1259–1264, 2021.

[5] E. Damiani, D. C. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante, "A reputation-based approach for choosing reliable resources in peer-to-peer networks," in *Proceedings of the 9th ACM conference on Computer and communications security*, 2002, pp. 207–216.

[6] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[7] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys (Csur)*, vol. 51, no. 4, pp. 1–35, 2018.

[8] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE transactions on information theory*, vol. 31, no. 4, pp. 469–472, 1985.

[9] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International conference on the theory and applications of cryptographic techniques*. Springer, 1999, pp. 223–238.

[10] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) lwe," *SIAM Journal on computing*, vol. 43, no. 2, pp. 831–871, 2014.

[11] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 2009, pp. 169–178.

[12] S. Kim, J. Kim, M. J. Kim, W. Jung, J. Kim, M. Rhu, and J. H. Ahn, "Bts: An accelerator for bootstrappable fully homomorphic encryption," in *Proceedings of the 49th Annual International Symposium on Computer Architecture*, 2022, pp. 711–725.

[13] N. Samardzic, A. Feldmann, A. Krastev, S. Devadas, R. Dreslinski, C. Peikert, and D. Sanchez, "F1: A fast and programmable accelerator for fully homomorphic encryption," in *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, 2021, pp. 238–252.

[14] Z. Brakerski and R. Perlman, "Lattice-based fully dynamic multi-key fhe with short ciphertexts," in *Annual international cryptology conference*. Springer, 2016, pp. 190–213.

[15] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International conference on machine learning*. PMLR, 2020, pp. 5132–5143.

[16] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[17] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *International Conference on Machine Learning*. PMLR, 2021, pp. 6357–6368.

[18] X. Yao and L. Sun, "Continual local training for better initialization of federated models," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 1736–1740.

[19] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 10 713–10 722.

[20] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 1126–1135.

[21] Y. Jiang, J. Konečnỳ, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," *arXiv preprint arXiv:1909.12488*, 2019.

[22] D. Li and J. Wang, "Fedmd: Heterogenous federated learning via model distillation," *arXiv preprint arXiv:1910.03581*, 2019.

[23] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 83–93, 2020.

[24] H. Yang, H. He, W. Zhang, and X. Cao, "Fedsteg: A federated transfer learning framework for secure image steganalysis," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1084–1094, 2020.

[25] N. P. Smart and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes," in *International Workshop on Public Key Cryptography*. Springer, 2010, pp. 420–443.

[26] C. Gentry and S. Halevi, "Implementing gentry's fully-homomorphic encryption scheme," in *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 2011, pp. 129–148.

[27] D. Stehlé and R. Steinfeld, "Faster fully homomorphic encryption," in *Advances in Cryptology-ASIACRYPT 2010: 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings 16*. Springer, 2010, pp. 377–394.

[28] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-lwe and security for key dependent messages," in *Annual cryptology conference*. Springer, 2011, pp. 505–524.

[29] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(leveled) fully homomorphic encryption without bootstrapping," *ACM Transactions on Computation Theory (TOCT)*, vol. 6, no. 3, pp. 1–36, 2014.

[30] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," *Cryptology ePrint Archive*, 2012.

[31] H. Chen, K. Laine, R. Player, and Y. Xia, "High-precision arithmetic in homomorphic encryption," in *Cryptographers' Track at the RSA Conference*. Springer, 2018, pp. 116–136.

[32] C. Bootland, W. Castryck, I. Iliashenko, and F. Vercauteren, "Efficiently processing complex-valued data in homomorphic encryption," *Journal of Mathematical Cryptology*, vol. 14, no. 1, pp. 55–65, 2020.

[33] S. Arita and S. Nakasato, "Fully homomorphic encryption for point numbers," in *International Conference on Information Security and Cryptology*. Springer, 2016, pp. 253–270.

[34] A. Jäschke and F. Armknecht, "Accelerating homomorphic computations on rational numbers," in *Applied Cryptography and Network Security: 14th International Conference, ACNS 2016, Guildford, UK, June 19-22, 2016. Proceedings 14*. Springer, 2016, pp. 405–423.

[35] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*. Springer, 2013, pp. 75–92.

[36] A. Khedr, G. Gulak, and V. Vaikuntanathan, "Shield: scalable homomorphic implementation of encrypted data-classifiers," *IEEE Transactions on Computers*, vol. 65, no. 9, pp. 2848–2858, 2015.

[37] J. Alperin-Sheriff and C. Peikert, "Faster bootstrapping with polynomial error," in *Advances in Cryptology–CRYPTO 2014: 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I 34*. Springer, 2014, pp. 297–314.

[38] R. Hiromasa, M. Abe, and T. Okamoto, "Packing messages and optimizing bootstrapping in gsw-fhe," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 99, no. 1, pp. 73–82, 2016.

[39] L. Ducas and D. Micciancio, "Fhew: bootstrapping homomorphic encryption in less than a second," in *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 2015, pp. 617–640.

[40] N. Gama, M. Izabachene, P. Q. Nguyen, and X. Xie, "Structural lattice reduction: generalized worst-case to average-case reductions and homomorphic cryptosystems," in *Advances in Cryptology–EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II 35*. Springer, 2016, pp. 528–558.

[41] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23*. Springer, 2017, pp. 409–437.

[42] Y. Aono, T. Hayashi, L. Wang, S. Moriai *et al.*, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE transactions on information forensics and security*, vol. 13, no. 5, pp. 1333–1345, 2017.

[43] T. Wang, Z. Cao, S. Wang, J. Wang, L. Qi, A. Liu, M. Xie, and X. Li, "Privacy-enhanced data collection based on deep learning for internet of vehicles," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6663–6672, 2019.

[44] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artificial intelligence," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6532–6542, 2019.

[45] H. Fang and Q. Qian, "Privacy preserving machine learning with homomorphic encryption and federated learning," *Future Internet*, vol. 13, no. 4, p. 94, 2021.

[46] F. Qiu, H. Yang, L. Zhou, C. Ma, and L. Fang, "Privacy preserving federated learning using ckks homomorphic encryption," in *International Conference on Wireless Algorithms, Systems, and Applications*. Springer, 2022, pp. 427–440.

[47] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[48] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[49] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.