

A Lightweight and Secure Deep Learning Model for Privacy-Preserving Federated Learning in Intelligent Enterprises

Reza Fotohi[✉], Fereidoon Shams Aliee[✉], and Bahar Farahani[✉]

Abstract—The ever-growing Internet of Things (IoT) connections drive a new type of organization, the Intelligent Enterprise. In intelligent enterprises, machine learning-based models are adopted to extract insights from data. Due to these traditional models' efficiency and privacy challenges, a new federated learning (FL) paradigm has emerged. In FL, multiple enterprises can jointly train a model to update a final model. However, firstly, FL-trained models usually perform worse than centralized models, especially when enterprises' training data is non-IID (Independent and Identically Distributed). Second, due to the centrality of FL and the untrustworthiness of local enterprises, traditional FL solutions are vulnerable to poisoning and inference attacks and violate privacy. Thirdly, the continuous transfer of parameters between enterprises and servers increases communication costs. Therefore, to this end, the FEDANIL+ model is proposed, a novel, lightweight, and secure Federated Deep LeArning Model that includes three main phases. In the first phase, the goal is to solve the data type distribution skew challenge. Addressing privacy concerns against poisoning and inference attacks is given in the second phase. Finally, to alleviate the communication overhead, a novel compression approach is proposed that significantly reduces the size of the updates. The experiment results validate that FEDANIL+ is secure against inference and poisoning attacks with better accuracy. In addition, in terms of model accuracy (13%, 16%, and 26%), communication cost (17%, 21%, and 25%), and computation cost (7%, 9%, and 11%) improvements over existing approaches. The FEDANIL+ code is available on GitHub¹.

Index Terms—Privacy-preserving, Non-IID, Blockchain, Communication Efficiency, Federated Learning (FL).

I. INTRODUCTION

THE Internet of Things (IoT) consists of multiple interconnected computing devices and mechanical and digital machines exchanging data with other IoT devices and the cloud. The rapid growth of IoT and cloud computing and the growing volume of data in enterprises have created a big data ecosystem. In this ecosystem, vast volumes of data from different sources are seamlessly integrated and shared

Reza Fotohi is with the Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran 1983969411, Iran (e-mail: r_fotohi@sbu.ac.ir).

Fereidoon Shams Aliee (Corresponding Author) is with the Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran 1983969411, Iran (e-mail: f_shams@sbu.ac.ir).

Bahar Farahani is with the Cyberspace Research Institute, Shahid Beheshti University, Tehran 1983969411, Iran (e-mail: b_farahani@sbu.ac.ir).

© 2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

¹<https://github.com/rezafotohi/FedAnilPlus>.

among stakeholders. Since sharing and outsourcing data to cloud centers risks breaching privacy and security, a promising technology, Federated learning (FL), has emerged [1], [9].

FL is a cutting-edge, secure, distributed machine learning (ML) technology that collaboratively trains a shared deep learning model using heterogeneous data from various clients. All client data remains private in FL, and only the updated parameters are sent to the central server [2]. This approach bypasses centralized data collection, thereby enhancing security and privacy. Consequently, the trained FL models in local clients must meet several criteria: Achieve better accuracy on non-IID datasets, demonstrate robustness and high resistance to inference and poisoning attacks, and maintain lower communication costs. Thus, this paper addresses the following three significant problems:

- *Non-IID*: Since enterprises collect training data based on their usage patterns and local environments, *data type distribution skew* often occurs among them. This skew can negatively affect a final global model's accuracy and convergence speed [3]–[6].
- *Privacy concern*: The FL-based technique keeps raw data from local enterprises private. It only shares updated gradient information with the server. However, FL does not guarantee adequate privacy and is vulnerable to poisoning and inference attacks. Model and data poisoning attacks pose significant threats to FL because they aim to degrade the global model's accuracy. Therefore, the attacker injects fake samples into the training dataset in data poisoning. Furthermore, model poisoning attacks manipulate updated parameters, hindering optimization and leading to higher test error rates. In inference attacks, adversaries infer the local sensitive data via the global model parameters to leakage privacy [3]–[6].
- *Communication costs*: FL-based techniques can generate many parameters when building and updating a model. Exchange of these parameters to the server can cause high communication overhead [3], [5], [6].

Therefore, the FEDANIL+ model has been proposed to overcome the above three challenges. The main contributions of this research are unfolding as follows:

- *Non-IID*: In the FEDANIL+ model, the non-IID challenge is addressed by the heterogeneity in data type distribution skew among different enterprises. To this end, a cosine similarity (CS) and affinity propagation (AP)-based clustering approach is proposed. Therefore, using

correct clustering based on these two techniques alleviates the heterogeneity between the local models. Then, the aggregation process can be appropriately performed on homogeneous models in each cluster without reducing the final model's accuracy.

- *Privacy concern:* Three steps in FEDANIL+ address privacy concerns. First, to defend against data poisoning attacks, enterprises whose *CS* of the updated gradient vector falls outside the specified range between two thresholds are ignored. If this occurs suspiciously n times, the enterprise is removed from the global model update aggregation process. Second, collude attacks are prevented based on the consortium blockchain and by randomly selecting enterprises. Finally, the third step utilizes the Cheon-Kim-Kim-Song (CKKS) *Fully Homomorphic Encryption (FHE)* technique and the blockchain consortium to prevent membership inference attack (MIA), reconstruction, and model poisoning attacks. In this step, local enterprises use *CKKS-FHE* to encrypt the *CHs index vector*, while the local models aggregate without decrypting the model parameters on the server side.
- *Communication costs:* In FEDANIL+, a two-step compression method is proposed to solve this challenge. In the first step, k-medoids [8]-based quantization is used to overcome the communication costs by entropy reduction in the local gradient vector. The main focus of the quantization technique is to remove gradients that have outliers, noise, or are out of range relative to the gradients of each cluster. The second step uses the entropy coding method in FEDANIL+. In this step, fewer bits are assigned to gradients with more repetition, and vice versa; more bits are assigned to gradients with less repetition.

The other sections are structured as follows. Section II briefly reviews the preliminary used. The related work is discussed in Section III. In Section, IV, the details of the FEDANIL+ are discussed. The convergence analysis is brought in Section V. In Section VI, experiment results are evaluated. Finally, Section VII brings the conclusions and future work.

II. PRELIMINARIES

The basic preliminaries adopted in FEDANIL+ are reviewed in this section. The key symbols used are listed in Table I.

A. Cosine Similarity

According to (1), this technique measures the similarity of two specified non-zero vectors by calculating the angle between two vectors. The output values belong to the interval $[-1,1]$. These two values obtained by calculating the angle between two vectors mean that if the output value shows -1 , it means that the two texts are less similar to each other, but if the output value shows 1 , it means which two texts are more similar to each other [9].

$$\text{Similarity}(x, y) = \cos(\theta) = \frac{x * y}{\|x\| * \|y\|} = \frac{\sum_{i=1}^n x_i * y_i}{\sqrt{\sum_{i=1}^n (x_i)^2 * \sum_{i=1}^n (y_i)^2}}. \quad (1)$$

TABLE I
KEY NOTATIONS.

Notation	Definition
$F^k(\omega)$	The loss function for enterprise k
$F^S(\omega)$	The loss function for the server
DS_k	Dataset for enterprise k
ω^k	The local enterprise k model
$\omega^S, \mathbb{G}\mathbb{I}$	Global model, Global iteration
$\omega(\tau)^k$	The local model τ of enterprise k
$\omega(\tau)^S$	The global model τ of server
$\bar{\omega}_r$	Averaged model in round r
$\Delta c, \chi^k$	Chosen enterprises, Malicious enterprise k
r / R	Current / Total communication round
$\nabla \omega, \omega, n$	Gradient, Weight, Total enterprises
θ^k	Angle between local and global model
Ψ, \aleph	CH and their index, Delay of local enterprise
Υ	Initial gradients using the CH index
τ, \mathcal{M}	Selected model; ($\tau \in \mathcal{M}$), Models vector
κ	Correctly predicted samples
ι	Total samples in the validation dataset

In (1), x and y represents the vector x and y respectively. Each x_i and y_i represents an element in these vectors. The angle between two vectors (x, y) is denoted by θ .

The FEDANIL+ model to measure the similarity between gradient vectors adopts the *CS* technique.

B. Consortium Blockchain

In a consortium blockchain, some aspects of enterprises are exposed, while others are private. Consensus methods are controlled by predefined nodes. A consortium blockchain is managed by multiple enterprises; Therefore, no single force here has a concentrated result [10].

C. Homomorphic Encryption (HE)

It is a type of encryption where calculations are performed on encrypted data without initial decryption. Also, the results of the calculations will be encrypted. Formally, such a scheme will be homomorphic if it satisfies (2) [2]:

$$E(m_1) * E(m_2) = E(m_1 * m_2) \quad \forall m_1, m_2 \in M. \quad (2)$$

In (2), Messages and a homomorphic operation are denoted by M and $*$, respectively. The main homomorphic operation is described by four main algorithms *KeyGen*, *Enc*, *Dec*, and *Eval*. These algorithms are listed separately below:

- 1) $\text{KeyGen}(1^\lambda) \rightarrow (p_k, s_k)$: It gets λ as the input security parameter and generates a public key p_k and a private key s_k .
- 2) $\text{Enc}(p_k, m) \rightarrow c$: It gets p_k and m as public key and message, respectively, and generates c as cipher text.
- 3) $\text{Dec}(s_k, c) \rightarrow m$: It gets s_k and c as private key and cipher, respectively, and gives m as a message.
- 4) $\text{Eval}(p_k, F, c_1, c_2, \dots, c_n) \rightarrow c^*$: The public key p_k takes as input an allowed evaluation function F and computes the cipher texts c_1 through c_n and evaluates to $F(c_1, \dots, c_n)$ if the following holds true: $\text{Dec}(s_k, \text{Eval}(p_k, F, c_1, c_2, \dots, c_n)) = F(m_1, \dots, m_n)$ where (c_1, \dots, c_n) is the encrypted message (m_1, \dots, m_n) .

Informally, the security parameter λ represents the difficulty of breaking the encryption key. Generally, an $m \in M$ message can be an integer string or another type of encryption [2]. In FEDANIL+, the *CKKS Fully homomorphic encryption (CKKS-FHE)* has been leveraged to encrypt local models.

III. RELATED WORK

This section will introduce recent approaches based on privacy-preserving, communication-efficient, non-IID data, and data skews on FL.

In [11], an approach called pFedV is proposed to address feature distribution skew. It modifies the last layer for feature extraction before classification layers, creating variable distribution feature maps instead of compressing the input. In [12], bias among local models is corrected by calibrating the logits to solve label distribution skew. Specifically, in FedBalance, the weak learner is trained locally, and its logits reflect the model's learning ability, which is fully influenced by locally unbalanced data. Merging the logits of the two models reduces the misclassification of minority classes and avoids overlearning of majority classes. In [13], the main skewed task is divided into multiple unskewed (balanced) sub-tasks for quantity distribution skew. Then, the representation of the original task is reconstructed using feature extractors for unskewed sub-tasks. In [14], local models' aggregation and learning operations are performed without access to private data to alleviate privacy concerns. The superiority of the proposed framework over the previous related approaches has been proven in various types of non-IID data distributions in the real world, such as time-skew, quantity-skew, scene-skew, and feature-skew. In [15], to deal with non-IID data skewness, local clients are divided into several groups, and instead of individual models, group models are trained for local clients. The clustering criterion is based on EARTH MOVER'S DISTANCE to group clients with similar data distribution by measuring the similarities of their models so that each group performs its respective local training in each round.

In [16], a compression framework called sparse ternary compression (STC) is proposed, which has low communication overhead. In [17], a Clustered FL (CFL) is proposed. The clustering is done between the clients in CFL. All the clients are grouped in homogenous clusters based on the similarity criterion. A secure aggregation-based method (RFA) is proposed in [18] to prevent poisoning attacks. In [3], an FL-based averaging method (FEDAVG) is proposed, which updates and constructs the global model by using a random selection of clients. FEDPROX emerged as an enhanced version of FEDAVG, designed to handle non-IID data and improve global model efficiency using Euclidean distance [19]. Another notable algorithm is FEDADAM, introduced for adaptive server optimization, which ensures model convergence despite heterogeneous data [20].

In summary, the FEDANIL+ model differs from other related approaches in the following areas:

- According to [6], existing approaches to heterogeneity typically address only one or two aspects, such as label skew, feature skew, temporal skew, and quantity skew,

without considering data type skew. However, FEDANIL+ specifically addresses heterogeneity from the perspective of data type.

- In the compression step, in the existing approaches, both the vector of the initial gradients and the Cluster Heads have been encoded using Huffman Coding. But in FEDANIL+ it is done separately: In this way, the initial gradient vector is coded with Adaptive Huffman coding (AHC), and the cluster head vector is encrypted with CKKS-FHE and finally, recorded in the blockchain.
- In existing methods, K-Means is used for Quantization based on clustering, which has a weakness in cluster head selection and is not sensitive to noisy data. But in FEDANIL+, it is based on K-Medoids, which are sensitive to noisy data and remove noisy data before clustering.

IV. FEDANIL+ MODEL

This section introduces the FEDANIL+ model with the following four phases.

A. Overview

This section first describes the use of blockchain and then the operations repeated in a global model round in FEDANIL+.

- Selecting a simple miner on the blockchain to send initial models to local clients.
- Random selection of local clients by the simple miner.
- Initialization and registration of CKKS-FHE, p_k and s_k parameters in the blockchain for selected local clients.
- Download and update the initial models by local clients and then upload them to the blockchain.
- Decentralized gradient aggregation to address the single-point-of-failure server problem.

In FEDANIL+, a round of global iteration involves five steps. Each step will be explained in detail below.

- 1) **INITIALIZATION.** Three initial models are created by the central enterprise and uploaded to the blockchain. Then, the parameters of CKKS-FHE, p_k , and s_k are set so that local enterprises can download these to update the models.
- 2) **SELECTION OF SIMPLE AND LEADER MINERS.** In FEDANIL+, miners play a crucial role in facilitating heavy blockchain operations. There are two types of miners in the proposed model: Simple and leader miners. The simple miner sends the initial models to local enterprises and, on the server side, evaluates their integrity. The goal of the leader miner is to perform the aggregation operation of the verified local models and update the global model. The simple miner with the highest reward is chosen as the leader miner.
- 3) **RANDOM SELECTION OF ENTERPRISES.** In this step, the random selection of enterprises prevents collusion attacks among them. By randomly selecting enterprises for each round of training the global model, malicious enterprises cannot predict which ones will be chosen. Consequently, enterprises are unlikely to coordinate collusion attacks with potential partner-enterprises, as

the selection process disrupts any patterns that could facilitate such collusion.

- 4) **LOCAL MODEL TRAINING.** First, each local enterprise downloads the initial global model from the blockchain using a simple miner. They then train these models in parallel, adapting them based on their respective data types. Next, each enterprise encodes the vector Υ using *AHC* according to (10), then according to (11), the vector Ψ is encrypted by *CKKS-FHE*. Finally, it uploads both encoded/encrypted gradient vectors to the blockchain.

In the FEDANIL+ to formulate the FDL model, the enterprise k data sample is represented as (X_k, Y_k) , where Y_k is the labels and X_k is its features. The loss function is defined individually for each enterprise in the training steps. As long as the model is not converged, reducing the loss function is performed. Therefore, in the FEDANIL+, the dataset of local models is shown as $DS_1, DS_2, DS_3, \dots, DS_n$, where the variable n represents the total enterprises. The loss function for the dataset DS_k ($k \in n$), in each local enterprise, is defined, via $F^k(\cdot)$ as (3):

$$F^k(\omega(\tau)) \triangleq \frac{1}{|DS_k|} \sum_{k \in DS_k} F(E(\omega(\tau)^k), X_k, Y_k) \quad (3)$$

$$F(\omega(\tau)) \triangleq \frac{\sum_{k \in DS_k} F(E(\omega(\tau)^k), X_k, Y_k)}{|\cup DS_k|} = \frac{\sum_{k=1}^n |DS_k| F^k(E(\omega(\tau)^k))}{|\cup DS|}. \quad (4)$$

In (4), DS_k related to the FEDANIL+ shows the size of the dataset of enterprises' local models. To specify the entire dataset for all enterprises according to the relation $|DS| \triangleq \sum_{k=1}^n |DS_k|$, and $DS_{k_1} \cap DS_{k_2} = \emptyset$ for $k_1 \neq k_2$. $F(\omega(\tau))$ is on the datasets of all enterprises $DS_1, DS_2, DS_3, \dots, DS_n$.

To calculate the loss function by the server, all local enterprises send their loss function along with their dataset size to the server. The purpose of training enterprises models is to calculate $F(\omega(\tau))$ to obtain the minimum function $F(\omega(\tau))$ which is shown in (5):

$$\omega(\tau)^S = \min \left\{ F(\omega(\tau)) \right\}. \quad (5)$$

In the FEDANIL+ based on (6), the Stochastic Gradient Descent (SGD) with momentum is used to optimize the weight of enterprise models. Therefore, each local enterprise shows its local parameters as $\omega(\tau)_r^k$ where $r = 0, 1, 2, 3, \dots, R$ and the variable r represents the rounds of local models. All enterprise parameters are initialized at $r = 0$. At $r \geq 1$, the local model's update is performed on the variable $\omega(\tau)_r^k$.

$$E(\omega(\tau)_r^k) = E(\omega(\tau)_{r-1}^k) - \eta \nabla F^k(E(\omega(\tau)_{r-1}^k)). \quad (6)$$

In (6), the variable η represents the learning rate. $\nabla F^k(E(\omega(\tau)_{r-1}^k))$ is the encrypted gradients, $E(\omega(\tau)_{r-1}^k)$, for the function F . The variable $E(\omega(\tau)_r^S)$, the global parameter is computed by the server using the aggregation operation of all locally updated models of local enterprises ($E(\omega(\tau)_r^k)$). Also, its volume of data is shown as the weight shown in (7):

$$\omega(\tau)_{\sim r} = \frac{\sum_{k=1}^n |DS_k| E(\omega(\tau)_r^k)}{|DS|}. \quad (7)$$

- 5) **MODEL AGGREGATION AND BLOCK GENERATION.** The leader miner, acting as the aggregation server, aggregates the locally encrypted gradient vectors that have passed the *CS* and *AP* steps. Subsequently, the averaged global model is recorded in a new block, enabling the next round of training to be executed by Δc . This process is repeated until the model achieves better model accuracy.

B. Addressing the non-IID

In this subsection, the challenge of Data Type distribution skew is addressed according to the following step.

DATA TYPE DISTRIBUTION SKEW. In this step, to solve data type distribution skew, a Personalized FL (PFL) based clustering approach based on *AP* and *CS* is proposed. In PFL, updating and building the models differs from that in FL. In PFL, several models are used, unlike FL. Therefore, each local enterprise trains various global models on their dataset and then sends it to the remote server for aggregation. In FEDANIL+, this step aims to create different homogenous clusters based on the distribution of different data types to solve the data type distribution skew challenge. Hence, employing the *CS* technique as described in (8), the distance between $\omega(\tau)_r^k$ and $\omega(\tau)_{r-1}^S$ is computed and stored in a list. Subsequently, utilizing the *AP* outlined in (9), the cluster members are identified based on the *CS* list. As a hyperparameter, the *AP* algorithm does not require the pre-defined total clusters.

$$\begin{aligned} \theta/CS(E(\omega(\tau)_r^k), E(\omega(\tau)_{r-1}^S)) &= \\ \frac{\langle \Delta E(\omega(\tau)_r^k), \Delta E(\omega(\tau)_{r-1}^S) \rangle}{\|\Delta E(\omega(\tau)_r^k)\| * \|\Delta E(\omega(\tau)_{r-1}^S)\|}. \end{aligned} \quad (8)$$

In (8):

- θ/CS : Similarity percentage of two models
- $\Delta E(\omega(\tau)_r^k)$: The enterprise k model.
- $\Delta E(\omega(\tau)_{r-1}^S)$: Updated parameters of the previous round of the server.

$$ClusterList[1 \dots M] = AP\left(\theta(\tau)_r^{k=1}, \dots, \theta(\tau)_r^{k \in \Delta c}\right); \quad (9)$$

In (9):

- $ClusterList[1 \dots M]$: Created clusters.
- AP : Determining cluster members.

Finally, FEDAVG is executed for each cluster, and the global model is built. Since the data type distribution in each cluster

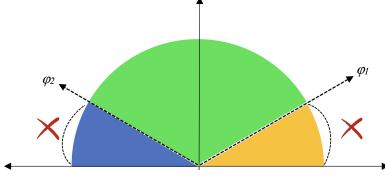


Fig. 1. φ_1 and φ_2 in the FEDANIL+ model.

group is homogenized using *AP* clustering. As a result, the FEDAVG aggregation algorithm will have the same performance as it does on homogeneous data. The comprehensive handling of non-IID data and the associated heterogeneous procedure is outlined in Algorithm 1.

Algorithm 1 Non-IID Data in the FEDANIL+

```

1: procedure HETEROGENEOUS ▷ Sect. IV(B)
2:   DATA TYPE SKEW();
3:    $\mathcal{M} = [CNN, ResNet50, GloVe];$ 
4:   for each  $\omega(\tau)^k \in \mathcal{M}$  in parallel do
5:     for  $r = 1$  to  $R$  do
6:       for each  $k \in \Delta_c$  in parallel do
7:          $\theta(\tau)_r^k = CS(E(\omega(\tau)_r^k), E(\omega(\tau)_{r-1}^S));$ 
8:         ClusterList[1... $\mathcal{M}] = AP(\theta(\tau)_r^{k=1}, \theta(\tau)_r^{k=2}, \dots, \theta(\tau)_r^{k \in \Delta_c});$ 
9:          $E(\omega(\tau)_r^S) \leftarrow FedAvg(ClusterList[1... $\mathcal{M}]);$ 
10:      end for
11:    end for
12:  end for
13: end procedure$ 
```

C. Addressing the Privacy-preserving

This section prevents inference and poisoning attacks in FEDANIL+.

STEP 1: DATA POISONING ATTACK PREVENTION. On the server side, following the computation of the *CS* between $E(\omega(\tau)_{r-1}^S)$ and $E(\omega(\tau)_r^k)$, the state of the local models undergoes verification based on a specific condition. The data poisoning attack performs the data poisoning operation using the revealed statistical distribution. Therefore, the desired model is ignored if the angle between two vectors is outside the two thresholds. If this value does not fall between φ_1 and φ_2 (as indicated by the green range in Fig. 1) for five consecutive rounds, the enterprise is classified as malicious and discarded before aggregation process.

According to Fig. 1, the angle range is between 0 and 180 degrees. The cosine similarity technique changes the continuous interval from +1 to -1 in this range. Using two threshold ranges φ_1 and φ_2 , these three color ranges are separated as follows:

- **YELLOW.** In this range, due to the small cosine angle between the prior round global model and the current local model, the probability of the model being poisoned is high, and therefore, it is discarded.
- **BLUE.** By injecting noise into the training data of enterprises, adversaries attempt to induce dissimilar behavior compared to the global model. This results in a large angle (indicating low similarity) between these gradient vectors. Consequently, the local models falling within this range are also excluded from further participation in

the model aggregation process to alleviate data poisoning attacks.

- **GREEN.** Finally, the local models, $E(\omega(\tau)_r^k)$, which form an angle, $\theta(\tau)_r^k$, within the green range (i.e., between $\varphi_1 = -0.7$ and $\varphi_2 = +0.7$) with the prior global model, $E(\omega(\tau)_{r-1}^S)$, exhibit a noise-free and normal local model. When a local model is deemed safe by the *CS*, it can be forwarded to subsequent steps.

STEP 2: COLLUDE ATTACK PREVENTION. Based on consortium blockchain and random selection of local enterprises, collusion attacks have been prevented. In this model, since $E(\omega(\tau)^S)$ represents an average of the behaviors of $E(\omega(\tau)_r^k) \in \Delta_c$, selected enterprises must alter their behavior to influence the global model with their attack, termed a collude attack. As the global model serves as an immutable reference and is directly influenced by enterprises, non-local enterprises can potentially bias this model. Therefore, this process is expected to safeguard enterprises from colluding attacks to a significant extent. Moreover, since not all enterprises receive updates of the global model in FEDANIL+ and the simple miner randomly selects among enterprises, collude attacks can be effectively deterred through random enterprise selection.

STEP 3: PREVENTION OF MIA, RECONSTRUCTION, AND MODEL POISONING ATTACKS. To reduce these attacks in FEDANIL+ model, CKKS-FHE and consortium blockchain techniques have been integrated. It should be noted that the server in FEDANIL+ is honest but curious.

A model poisoning attack seeks to manipulate local parameters. Hence, it's imperative for FEDANIL+ to prevent the exposure of updated parameters during the aggregation process and global model update. Consequently, in FEDANIL+, leveraging CKKS-FHE can effectively prevent the disclosure of local parameters, enabling servers to execute the requisite computations for aggregation and global model creation. On the other hand, the leader miner aggregates and updates the global model in the consortium blockchain, which has limited access to the public.

In a MIA, the intruder scrutinizes the global model to ascertain whether a specific sample exists within the training dataset. This is done through questions and answers from a trained machine-learning model. At first glance, it seems that to defend against this attack, the model architecture should be hidden from the attackers. However, hiding the model architecture in FL models is impossible because servers and clients follow a model with the same architecture. In fact, in a MIA where the attacker has the architecture used to train the real data, this attacker can, with a fake dataset and, by accessing the parameters of the model update, be able to infer the real training data so that the privacy violation occurs in enterprises.

In a reconstruction attack, the honest-but-curious server can access local model parameters. Since in the proposed model, all the parameters are encrypted by CKKS-FHE, this attack cannot compare the fake data output with any valid source to violate the privacy of the local model's training data by identifying the content of these parameters.

Therefore, to avoid revealing the parameters, the gradients of each enterprise based on two vectors, Ψ and Υ according to (10) and (11) in the blockchain are recorded. Specifically, first, according to (10), the vector Υ is encoded using *AHC*. According to (11), the vector Ψ is encrypted by *CKKS-FHE*, and finally, both vectors are recorded in the blockchain.

$$En(\rho[\]) \leftarrow Encoding(\rho[\], AHC). \quad (10)$$

$$E(CH[\]) \leftarrow Encrypt(CH[\], p_k). \quad (11)$$

In (10) and (11):

- $\rho[\]$, $En(\rho[\])$: Υ , Vector of encoded Υ .
- AHC , p_k : Adaptive Huffman coding, The public key.
- $CH[\]$, $E(CH[\])$: Ψ , Vector of encrypted Ψ .

D. Addressing the Communication costs

This section describes the gradient compression phase, which includes the following two main steps in detail.

STEP 1: QUANTIZATION. The quantization technique is used to reduce the entropy in the gradient vector. This technique is based on clustering. Specifically, the *K-Medoids* is used to minimize the number of gradients in the gradient vector, which improves the *K-Means* algorithm. In *K-Medoids*, which operates through iterative repetition, all datasets are segmented into subgroups called clusters. In these clusters, each gradient belongs to only one cluster. For a better understanding, an example of the *Quantization* process is given in Fig. 2.

(1): Fig. 2 shows a $4 * 4$ matrix containing an enterprise's initial weights. The goal is to minimize the gradients so that similar and close gradients are placed in a cluster.

(2): Then, by selecting Medoids on the members of each cluster, a gradient is recorded as *Cluster Head (CH)* in the Ψ .

(3): Using the Ψ vector, a new matrix called Υ is defined, which includes the index of *CHs* (instead of real gradients). The vector Ψ is encrypted using *CKKS-FHE*. Encryption protects *CH* gradients against inference and poisoning attacks in the training and aggregation phase. On the other hand, the Υ vector is delivered to the *Lossless Entropy Encoding* step to perform the last step of compression. The reason why *CKKS-FHE* is not applied to this vector is that this vector after compression by *Lossless Entropy Encoding* only contains the number of repetitions of the index related to *CHs*. Therefore, on the server side, no information is revealed by decoding this vector except the number of repetitions of indexes corresponding to *CHs*.

In addition, for calculating Ψ in the *K-Medoids*, the total clusters are automatically calculated using the *Silhouette index* [21], and the clustering operation is categorized into K separate clusters. In the FEDANIL+, the frequency of optimal clusters was estimated to be $K = 5$ based on the *Silhouette index*. Also, the *Quantization* operation is shown in (12).

$$K\text{-Medoids}(\rho[\]) = \left\{ \forall_{\rho[i]} \in \Delta c, \text{ let } \rho[i] \leftarrow i \in CH[\] \right\}. \quad (12)$$

In (12):

Initial gradients using the Cluster Heads index				Cluster Heads and their index		Initial gradients			
3	0	2	1	3:	1.80	2.24	-0.75	1.10	0.04
1	1	0	3	2:	1.16	0	0	0.45	2.32
0	3	1	0	1:	0.04	-1.35	1.80	0.07	-1.45
3	1	2	2	0:	-0.75	1.64	-0.11	1.28	1.16
				(3)	(2)	(1)			

Fig. 2. Quantization in the FEDANIL+.

- $\rho[i]$, $CH[\]$: Υ , *CHs* vector.

As a result, by using this *Quantization*, we can greatly reduce the entropy of gradients by clustering and displaying gradients using their *CHs*.

STEP 2: LOSSLESS ENTROPY ENCODING. This step's purpose is to decrease the bits used to transfer the local enterprise's gradient vector. This operation employs *AHC*. The purpose of *AHC* is gradients lossless compression. This algorithm is a type of entropy encoding algorithm. In *AHC*, characters are displayed with a fixed number of bits (0, 1). It allocates fewer bits to frequently used parameters to minimize the required bits and more bits to less frequently used parameters. As described in STEP 1, the Quantization output will include two main vectors Ψ and Υ . Therefore, in this step, only the Υ vector is encoded using *AHC* and then recorded in the blockchain. On the server side, this vector is decoded by the leader miner, and the gradient vector is reconstructed. *AHC* coding is given in (13):

$$\left\{ \forall_k \in \Delta c, \text{ let } Encoding(\rho[\], AHC) \right\}. \quad (13)$$

The comprehensive handling of communication costs and the associated compression procedure is outlined in Algorithm 2.

Algorithm 2 Communication costs in the FEDANIL+

```

1: procedure COMPRESSION
2: | | Download  $E(\omega(\tau)_r^S)$  for  $k \in \Delta c$ ; ▷ Sect. IV(D), Step 1
3: | | QUANTIZATION();
4: | |  $CH[\ ] \leftarrow K\text{-Medoids}(\rho[\ ])$ ;
5: | |  $E(\omega(\tau)_r^k) \leftarrow K\text{-Medoids}(\rho[\ ])$ ;
6: | | for each  $CH[i] \in E(\omega(\tau)_r^k)$  do in parallel
7: | | | for each  $Cluster\_Mem[j] \in CH[i]$  in parallel do
8: | | | |  $\rho[j] \leftarrow \{i \in CH[\ ]\}$ ;
9: | | | end for
10: | | | |  $E(\omega(\tau)_r^k) \leftarrow \rho[j]$ ;
11: | | end for
12: | | LOSSLESS ENCODING(); ▷ Sect. IV(D), Step 2
13: | | for each  $k \in \Delta c$  in parallel do
14: | | |  $Encoding(\rho[\ ], AHC)$ ;
15: | | |  $Encryption(CH[\ ], CKKS - FHE)$ ;
16: | | end for
17: | | Uploadk → S(ρ[ ], CH[ ]);
18: end procedure

```

V. CONVERGENCE ANALYSIS

To analyze the FEDANIL+ model convergence, it is assumed that $F(\omega)$ is non-convex and have two assumptions, which are detailed in [4]:

ASSUMPTION 1(β -SMOOTHNESS). Assuming $\nabla F^k(\omega)$ is β smoothness, therefore, $\|(\nabla F^k(\omega) - \nabla F(\omega_*))\| \leq \beta \|(\omega^k - \omega_*)\|$, where $\forall \omega^k, \omega_* \in \mathbb{R}^d$. Where β is a positive constant.

ASSUMPTION 2. Assuming $F^k(\omega)$ in selected local enterprises be locally convex. Hence, the following formula will hold for $F^k(\omega)$. $F^k[(\varphi\omega + (1-\varphi)\omega_*)] \leq [\varphi F^k(\omega) + (1-\varphi)F(\omega_*)]$, $\forall \omega, \omega_* \in \mathbb{R}^d$, $\varphi \in [0, 1]$ and φ is a positive constant, and distance for both ω and ω_* at a radius $ri > 0$. In the next discussion, we prove the convergence of the FEDANIL+ model weight parameter ω^k in training local models.

THEOREM 1. For β as a constant, if $\eta \leq \frac{1}{\beta}$, then there exists $\|(F^S(\omega_{r+1}) - F^S(\omega_*))\| \leq \|(F^S(\omega_r) - F^S(\omega_*))\|$, where $F^S(\omega_r)$ represents the loss function of the aggregated global model. ω_r and ω_* denote the regular model and the optimized model at round r On the server side, respectively.

PROOF. Following the discussion, the proof of the $\|F^S(\omega_r) - F^S(\omega_*)\|^2$ is given.

$$\begin{aligned} & \| (F^S(\omega_{r-1}) - \eta \nabla F^S(\omega_{r-1}) - F^S(\omega_*)) \|^2 \\ &= \| F^S(\omega_{r-1}) - F^S(\omega_*) \|^2 - 2\eta \nabla (F^S(\omega_{r-1})^{\text{G}\ddagger} F^S(\omega_{r-1})) \\ &\quad - (F^S(\omega_*)) + \eta^2 \|\nabla F^S(\omega_{r-1})\|^2 \\ &\leq \| F^S(\omega_{r-1}) - F^S(\omega_*) \|^2 - \eta \frac{\|\nabla \omega_{r-1}\|^2}{\beta} + \eta^2 \|\nabla F^S(\omega_{r-1})\|^2 \\ &= \| F^S(\omega_{r-1}) - F^S(\omega_*) \|^2 - \eta \left(\frac{1}{\beta} - \eta \right) \|\nabla F^S(\omega_{r-1})\|^2. \end{aligned}$$

In the end, the following relation is output:

$$\|F^S(\omega_r) - F^S(\omega_*)\|^2 \leq \|F^S(\omega_{r-1}) - F^S(\omega_*)\|^2.$$

COROLLARY 1(ANTI-INFERENCE AND ANTI-POISONING). If the number of normal enterprises exceeds the number of intruder enterprises, the proposed anti-poisoning and anti-inference approach converges to a model that mirrors the normal enterprise models.

PROOF. To prove the aforementioned Corollary, let's clarify with specific scenarios. In Scenario 1, all local models are normal. In Scenario 2, all local models are adversaries. adversary enterprises launch inference and poisoning attacks to disrupt the model's accuracy and privacy. The FEDANIL+ model will converge in both scenarios 1 and 2. However, the direction of the final model obtained from Scenario 1 and Scenario 2 will differ. The global models for Scenarios 1 and 2 are denoted as $\omega^{S(\text{normal})}$ and $\omega^{S(\text{intruder})}$, respectively. The final model obtained after the last round will be a model between $\omega^{S(\text{normal})}$ and $\omega^{S(\text{intruder})}$. If the μ variable represents the percentage of intruder enterprises, the final aggregated model after the r^{th} round is denoted as (14):

$$\omega^S = [(1-\mu) * \omega_r^{S(\text{normal})}] + [\mu * \omega_r^{S(\text{intruder})}]. \quad (14)$$

According to (14), if the majority of local models are normal (i.e., $\mu \approx 0.2$), then the global model will closely resemble $\omega^{S(\text{normal})}$. In this case, μ effectively pulls the aggregated model towards the normal enterprises' model, and the proposed anti-poisoning and anti-inference mechanisms mitigate the impact of intruder enterprises in each round.

Next, we elaborate on the aggregated local model's convergence with the same distribution. In the FEDANIL+ model based on federated learning, there are n local enterprises. The dataset of enterprises is denoted by DS_1, DS_2, \dots, DS_n , each possessing a different data distribution p^ϱ ($\varrho = 1, 2, \dots, n$).

Assuming that stochastic gradients $SG(.)$ are unbiased with a distinct probability distribution in each round, i.e., $\mathbb{E}[SG^\varrho(\omega_r)] = \nabla F^\varrho(\omega_r)$.

THEOREM 2. In the FEDANIL+ model, following the addressing of the non-IID challenge (as discussed in Section IV-B), it is assumed that the set of uploaded weight parameters is selected from datasets with the same distribution p^ϱ . We can establish the following relation compared to FEDAVG:

$$\mathbb{E}\|\omega_r^\varrho - \omega_*^\varrho\|^2 \leq \mathbb{E}\|\bar{\omega}_r - \omega_*^\varrho\|^2, \quad (15)$$

where ω_*^ϱ represents the optimized weight with p^ϱ distribution to fit the dataset. ω_r^ϱ is the received local model with p^ϱ distribution. And finally, $\bar{\omega}_r$ defines the FEDAVG uniform global model in round r .

PROOF. Utilizing induction, we can prove the result. First, we include the following two relationships:

$$\bar{\omega}_{r=1} = \omega_0 - \eta \nabla \bar{S}G_{r=1}, \quad (16)$$

$$\omega_{r=1}^\varrho = \omega_0 - \eta \nabla SG_{r=1}^\varrho, \quad (17)$$

After the first round on the server side, FEDAVG gradients are displayed with $\bar{S}G_{r=1}$ and FEDANIL+ gradients with $SG_{r=1}^\varrho$. The execution of the first round of SGD WITH MOMENTUM with FEDAVG is given in (16) and the execution of the first round of SGD WITH MOMENTUM with FEDANIL+ is given in (17). Therefore, according to (16) and (17), we can conclude (18):

$$\mathbb{E}\|\omega_{r=1}^\varrho - \omega_*^\varrho\|^2 \leq \mathbb{E}\|\bar{\omega}_{r=1} - \omega_*^\varrho\|^2. \quad (18)$$

After this, it is assumed that (19) is correct in r^{th} round, we will have:

$$\mathbb{E}\|\omega_r^\varrho - \omega_*^\varrho\|^2 \leq \mathbb{E}\|\bar{\omega}_r - \omega_*^\varrho\|^2. \quad (19)$$

Now, using the (16) and (17), we check the round $(r+1)^{\text{th}}$ and then the (20) is obtained:

$$\mathbb{E}\|\omega_r^\varrho - \eta \nabla \bar{S}G_r - \omega_*^\varrho\|^2 \leq \mathbb{E}\|\bar{\omega}_t - \eta \nabla SG_r^\varrho - \omega_*^\varrho\|^2, \quad (20)$$

And finally, the (21) can be expressed:

$$\mathbb{E}\|\omega_{r+1}^\varrho - \omega_*^\varrho\|^2 \leq \mathbb{E}\|\bar{\omega}_{r+1} - \omega_*^\varrho\|^2. \quad (21)$$

When the optimized parameters of local models are correctly compatible with the dataset pattern of local enterprises, we call it convergence. The convergence has been accepted if the training parameters undergo a constant and unchanged process. The ultimate goal of convergence in non-IID data is to converge each non-IID data set to the optimal model individually. Therefore, if the loss function $F(\omega)$ approaches 0 continuously, the trained model converges to the optimal model.

VI. EXPERIMENTS

This section compares the performance of the FEDANIL+ model against six well-known baseline methods: FEDAVG [3], FEDPROX [19], FEDADAM [20], STC [16], CFL [17], and RFA [18].

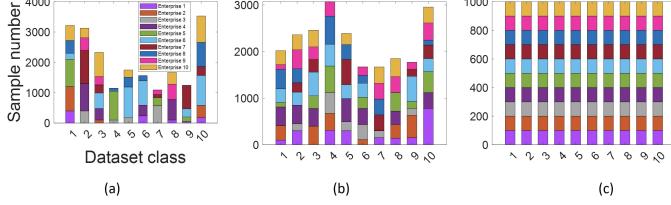


Fig. 3. Non-IID data distribution. Different colors represent different labels, and each column represents an enterprise's data distribution. (a) $\alpha \rightarrow 0.1$; (b) $\alpha \rightarrow 1$; (c) $\alpha \rightarrow \infty$.

TABLE II
HYPERPARAMETERS.

Hyperparameters	Description
<i>SGD with momentum=0.9</i>	Optimizer for CNN
<i>Adam</i> ($\beta_1, \beta_2 = 0.9$)	Optimizer for ResNet50
$K = 5$	Total clusters for <i>K-Medoids</i>
$R = 50$	Total communication rounds
$B_1 = 64$	Training batch size for local models
$B_2 = 128$	Testing batch size for local models
$\mu = 20\%$, $\varepsilon = 30$	Rate of Malicious, Total Epochs
$C = 100$, $\eta = 0.01$	Total enterprises, Learning rate

A. Experimental Setup

The experiments rely on a statistical concept called the Dirichlet distribution (denoted as $\text{Dir}(\alpha)$), which is a probability distribution used for continuous, multi-dimensional data. This distribution is characterized by a parameter α , which must be a positive number greater than zero ($p \sim \text{Dir}(\alpha)$, $\alpha > 0$) [22]. $\text{Dir}(\alpha)$ is used to generate non-IID datasets. The parameter α is utilized to control the level of non-IID data distribution for each enterprise. As illustrated in Fig. 3a, a lower α value leads to a more skewed distribution, resulting in a higher degree of non-IID data. Conversely, a larger α value creates a distribution closer to a uniform one, mimicking IID (Fig. 3c). The FEDANIL+ model utilizes the Dirichlet distribution (with α set to 0.1) to partition the overall dataset. This results in each participating enterprise having a unique distribution of class types within their local dataset. The total samples in each local dataset will also vary. Therefore, the FEDANIL+ model considers one mode for non-IID data called *Data Type skew*, in which the data type differs in different enterprises.

IMPLEMENTATION. Our experiments were conducted on the macOS Ventura operating system using Python 3.10.9. We utilized the PyTorch [23] library to train the models. Performance evaluation of the FEDANIL+ model has been done on the four popular datasets. According to [3], used hyperparameters for FEDANIL+ model are shown in Table II.

DATASETS AND MODELS. To evaluate the FEDANIL+, we will have four diverse non-IID datasets and three machine-learning models according to Table III.

- 1) *Sentiment analysis.* The purpose of the Sent140 [26] is sentiment analysis. A linear model using average GLOVE EMBEDDINGS [27] of tweet words was used. Also, the binary logistic loss was used to train the model. Sent140 has 2 classes. Table III gives the rest of its more details.

TABLE III
THE DATASET USED IN THE FEDANIL+ (SENTIMENT ANALYSIS=S.A, IMAGE CLASSIFICATION=I.C, CHARACTER-LEVEL=C.L).

Dataset	Task	Models	#Train	#Test
Sent140	S.A	GloVe	57K	15K
Fashion-MNIST	I.C	CNN	60K	10K
FEMNIST	C.L	CNN	49K	4.9K
CIFAR-10	I.C	ResNet50	50K	10K

- 2) *Image classification.* The purpose of the Fashion-MNIST [24] is image classification. This dataset contains low-resolution grayscale images designed with a scale of 28*28. Fashion-MNIST has 10 classes. Table III gives the rest of its more details.
- 3) *Handwritten character recognition.* The purpose of the FEMNIST dataset [25] is to analyze handwritten characters. This dataset contains grayscale images designed with a scale of 28*28. FEMNIST has 62 classes. Table III gives the rest of its more details.
- 4) *Image classification.* The purpose of the CIFAR-10 [28] is image classification. This dataset contains color images designed with a scale of 32*32. CIFAR-10 has 10 classes. Table III gives the rest of its more details.

EVALUATION METRICS. The metrics used to evaluate the proposed model are explained in detail below:

- 1) *Accuracy.* In FEDANIL+, a model's performance is measured by how well it predicts on a validation dataset. This is calculated as the number of correct predictions (κ) divided by the total samples (ι) in the validation set, as shown in (22).

$$\text{Accuracy} = \frac{\kappa}{\iota} * 100. \quad (22)$$

- 2) *Communication overhead.* One of the important goals of FEDANIL+ is to achieve the highest compression rate to reduce communication costs. To achieve this goal, the communication cost was calculated using gradient vector compression and reducing the bits required to transfer the local gradient vector of each enterprise to the server. This metric is calculated by (23), (24), and (25).

$$\text{COMM}_{C2S_Side} = \sum_{r=1}^R \left(\sum_{k=1}^{\Delta c} b_{\rho[k]} + b_{CH[k]} \right). \quad (23)$$

$$\text{COMM}_{S2C_Side} = \sum_{r=1}^R \left(b_E(\omega(\tau)^S) \right). \quad (24)$$

$$\text{COMM}_{Total} = \left(\text{COMM}_{C2S_Side} + \text{COMM}_{S2C_Side} \right). \quad (25)$$

In (23) and (24):

- COMM_{C2S_Side} : The communication costs from local enterprises to the server.
- $b_{\rho[k]}$: The number of bits used by each local enterprise in entropy coding (AHC) on the Υ vector.
- $b_{CH[k]}$: The consumed bits of each local enterprise in encryption on the Ψ vector.

- $COMM_{S2C_Side}$: The communication costs from the server to the local enterprises.
- $b_E(\omega(\tau)^S)$: Bits used for global model parameters.
- $COMM_{Total}$: Total number of bits used.

3) *Computation overhead.* The computation overhead of the FEDANIL+ model was calculated separately for the enterprises and server sides. The techniques that inject computation cost for the FEDANIL+ on the enterprise side include: *Quantization*, *Coding*, *Training*, and *FHE*. On the server side, include *AP* and *CS*. The key point is that the *FHE* on the client side has the highest computation, with a complexity of $O(N^2)$. On the server side, the *AP* and *CS* create the most computation. This is because they are done sequentially, one after the other, resulting in a computation overhead of $O(N^2)$. Finally, the highest computation cost is related to the server, which is of the order of $O(N^2)$.

B. Experimental Results

In this section, we train three models on four diverse datasets, Sent140 [26], Fashion-MNIST [24], FEMNIST [25], and CIFAR-10 [28], and evaluated the robustness of the FEDANIL+ model. Then, the FEDANIL+ model was compared and evaluated against other approaches, STC, CFL, RFA, FEDADAM, FEDPROX, and FEDAVG. These comparisons were done according to Fig. 4, Fig. 5, and Fig. 6.

OVERALL ACCURACY. As illustrated in Fig. 4a, Fig. 4b, Fig. 4c, and Fig. 4d, with the increase in the number of enterprises from 40 to 100, the FEDANIL+ accuracy has increased. The first reason for the improved accuracy of the FEDANIL+ in later training rounds is that on the server side, the models from all enterprises are clustered based on the distribution of their *data type*, using clustering techniques like *CS* and *AP*. Then, the aggregation is performed within these homogeneous clusters. Therefore, the *CS* and *AP* techniques aim to cluster the heterogeneous models from the local models. The homogenization of the local models, achieved through the clustering, reduces convergence time and increases global model accuracy. The second reason for the improved accuracy of the FEDANIL+ model is that it helps prevent poisoning attacks. These attacks aim to increase the convergence time and reduce the accuracy of the global model. The reason behind the low accuracy of the CFL approach is that they use an encryption technique, which leads to a significant loss of gradients during the decryption process (referred to as the Lossy data approach). This gradient loss ultimately results in a lower model accuracy. The efficiency of the RFA approach is weak on non-IID data, which leads to low model accuracy. A strong reason for the poor accuracy of the model in the STC method is the use of the gradient compression technique, which has led to the loss of useful gradients. In FEDADAM, FEDAVG, and FEDPROX, the performance against poisoning attack and non-IID data is weak, causing the model to diverge. Therefore, these have caused the model's accuracy to decrease.

COMMUNICATION OVERHEAD. As shown in Fig. 5a, Fig. 5b, Fig. 5c, and Fig. 5d, the total communication cost of the

FEDANIL+ model has been compared and evaluated. While the communication overhead increased for all approaches, the communication cost for the FedAnil+ model was much smaller than the other approaches. In the FEDANIL+ model, due to the use of the *K-Medoids Quantization* and *LossLess Entropy Encoding*, more bits are compressed and just useful gradients are transmitted to the aggregator server. In baseline approaches, by injecting poisoning attacks, the model diverges, and its accuracy decreases. Because the model accuracy is affected by poisoning attacks, local enterprises and servers consume more rounds. As a result, more bits are consumed, and this work has caused a lot of communication overhead for the models. Because the parameter compression operation in the STC approach is performed after the model training steps, the quantization process and the communication cost are not optimized. In the CFL approach, Bipartition is used to find a correct partitioning, which, in addition to having heavy operations, this approach is not based on parameter update compression and has a higher communication overhead. The RFA approach performs poorly in compression operations on non-IID data. Also, due to not using entropy coding, the length of bits is longer for transferring gradients.

COMPUTATION OVERHEAD. As demonstrated in Fig. 6a, Fig. 6b, Fig. 6c, and Fig. 6d, with the increase in number of enterprises, the FEDANIL+ computation cost has increased. All approaches have been compared by increasing the number of enterprises from 20 to 100. The computation cost also increases with the growth of enterprises. Because in the training phase, each local enterprise injects separate computations. This metric is intended to demonstrate the computation cost of the techniques employed. Based on the results presented in Fig. 6, the FEDANIL+ model has low computational requirements due to avoiding heavy computational operations. In contrast, the STC, CFL, and RFA approaches utilized computationally heavy operations, such as sparse ternary encoding, aggregation oracles, and lossy encryption. On the other hand, the FEDANIL+ has a higher computational cost than the FEDAVG due to its use of quantization and homomorphic encryption operations. The low computational of the FEDAVG is because it performs fewer computations on the client side, resulting in low overall computation overhead. While the FEDANIL+ model has low computational overhead, it does not demonstrate better performance compared to the FEDAVG.

THE FEDANIL+ RESISTANCE TO INFERENCE ATTACKS. To calculate the Gradient Matching Loss (GML), the L-BFGS model [29] is adopted as an optimizer in FEDANIL+. The difference between fake adversary-generated and real samples is represented by *GML*. According to [30], any *GML* value greater than 0.15 will not leak any information. Conversely, the smaller the *GML* value is than 0.15, the more information is leaked. As Fig. 7 shows, when the total number of rounds is set to 100, in the FEDANIL+ model and the execution of the 20th round, the value of the *GML* is 0.17. This means the difference between the real and fake parameters loss function is 0.17. Continuing and repeating the number of rounds from 20 to 100, FEDANIL+ has the same *GML* value of 0.17 due to its gradient leakage resistance approach and does not leak any information (**No Leak**). Therefore, this non-leakage of

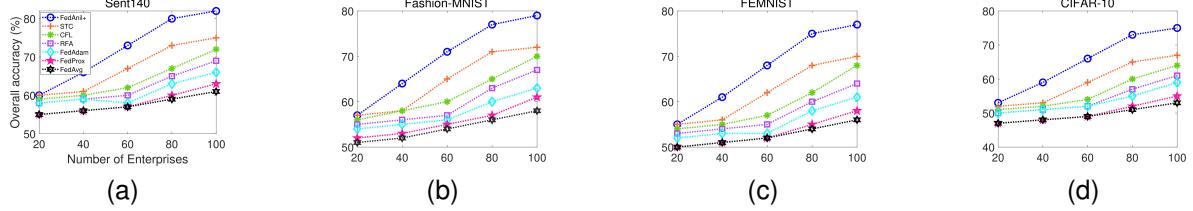


Fig. 4. Comparison of the overall accuracy between FEDANIL+ and existing methods where the ε is fixed at 30 and $R = 50$. ($\mu = 20\%$).

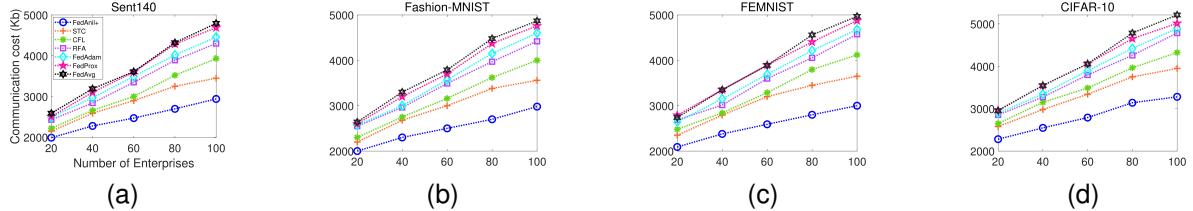


Fig. 5. Comparison of the communication cost between FEDANIL+ and existing methods where the ε is fixed at 30 and $R = 50$. ($\mu = 20\%$).

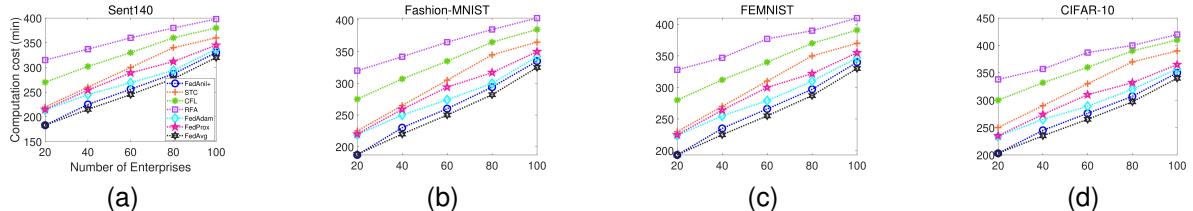


Fig. 6. Comparison of the computation cost between FEDANIL+ and existing methods where the ε is fixed at 30 and $R = 50$. ($\mu = 20\%$).

parameters will be preserved until round 100. In particular, the reason for the robustness of the FEDANIL+ is a *CKKS-FHE* technique, in which the encrypted local enterprise's models are aggregated without decryption on the server side. This makes the intruder not understand the content of the parameters. *GML* value is 0.13, 0.10, and 0.09 for STC, CFL, and RFA methods, respectively, which leaked some image's pixels, and due to the lack of a strong privacy protection approach, this leakage was repeated up to round 100 (**Leak with artifacts**). The *GML* for FedAdam, FedProx, and FedAvg methods have values of 0.04, 0.025, and 0.015, respectively. This means the methods in this *GML* have many parameter matches, and the attacker's dummy data is much closer to the original data. Therefore, these methods have deep leakage and do not preserve the privacy of parameters (**Deep Leakage**). By repeating the number of rounds from 20 to 100, the same deep leakage occurred, and a higher percentage of the real parameters information matches the fake parameters information by the attacker. The *GML* value remains constant from round 20 onwards because none of the approaches could prevent this recovery and leakage due to the lack of a robust privacy protection approach. Thus, the attacker was able to recover more real parameters.

VII. CONCLUSIONS AND FUTURE WORK

This paper proposed a lightweight model named FEDANIL+. In particular, the main innovation of FEDANIL+, in addition to alleviating the privacy concern, is reducing

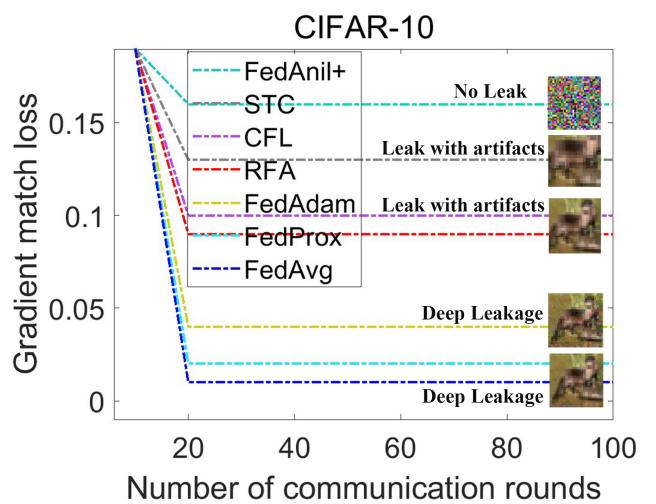


Fig. 7. *GML* between FEDANIL+ and existing methods where the ε is fixed at 30 and $R = 100$. ($\mu = 20\%$).

the model's size and addressing the data type distribution skew. Our simulation results validate that the FEDANIL+ improvements over existing approaches regarding accuracy, communication, and computation overhead. Moreover, the convergence analysis showed that the FEDANIL+ model converges to the set of optimal model parameters. In future work, we will focus on three non-IID data distribution skews,

i.e., Feature, Label, and Data type.

REFERENCES

- [1] N. M. Hijazi et al., "Secure Federated Learning With Fully Homomorphic Encryption for IoT Communications," in IEEE Internet of Things Journal, vol. 11, no. 3, pp. 4289-4300, 1 Feb.1, 2024.
- [2] C. Zhang et al., "Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning," in Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC 2020), 2020.
- [3] B. McMahan et al., "Communication-efficient learning of deep networks from decentralized data," in Artificial intelligence and statistics, 2017: PMLR, pp. 1273-1282.
- [4] R. Fotohi et al., "Decentralized and robust privacy-preserving model using blockchain-enabled Federated Deep Learning in intelligent enterprises," Applied Soft Computing, 111764, 2024.
- [5] R. Fotohi et al., "Federated Learning: Solutions, Challenges, and Promises," In 2022 6th Iranian Conference on Advances in Enterprise Architecture (ICAEEA) (pp. 15-22). IEEE, 2022.
- [6] P. Kairouz et al., "Advances and open problems in federated learning," Foundations and Trends® in Machine Learning, vol. 14, no. 1–2, pp. 1-210, 2021.
- [7] J. H. Cheon et al., "Homomorphic encryption for arithmetic of approximate numbers," in Advances in Cryptology—ASIACRYPT 2017, Hong Kong, December 3–7, 2017, Part I 23, 2017: Springer, pp. 409-437.
- [8] P. Arora and S. Varshney, "Analysis of k-means and k-medoids algorithm for big data," Procedia Computer Science, vol. 78, pp. 507-512, 2016.
- [9] F. Wei et al., "Privacy-preserving implicit authentication protocol using cosine similarity for Internet of Things," IEEE Internet of Things Journal, vol. 8, no. 7, pp. 5599-5606, 2020.
- [10] B. Zhong et al., "Hyperledger fabric-based consortium blockchain for construction quality information management," Frontiers of Engineering Management, vol. 7, no. 4, pp. 512-527, 2020.
- [11] Y. Mou et al., "pFedV: Mitigating Feature Distribution Skewness via Personalized Federated Learning with Variational Distribution Constraints," In Pacific-Asia Conference on Knowledge Discovery and Data Mining, Cham: Springer Nature Switzerland, pp. 283-294, 2023.
- [12] Y. Wang et al., "Federated Skewed Label Learning with Logits Fusion," arXiv preprint arXiv:2311.08202, 2023.
- [13] C. Wang et al., "FedBnR: Mitigating federated learning Non-IID problem by breaking the skewed task and reconstructing representation," Future Generation Computer Systems, vol. 153, pp. 1-11, 2024.
- [14] Y. Pang et al., "Federated Learning for Crowd Counting in Smart Surveillance Systems," in IEEE Internet of Things Journal, vol. 11, no. 3, pp. 5200-5209, 2024.
- [15] J. Shu et al., "Clustered Federated Multitask Learning on Non-IID Data With Enhanced Privacy," in IEEE Internet of Things Journal, vol. 10, no. 4, pp. 3453-3467, 2023.
- [16] F. Sattler et al., "Robust and communication-efficient federated learning from non-iid data," IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 9, pp. 3400-3413, 2019.
- [17] F. Sattler et al., "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," Transactions on Neural Networks and Learning Systems, vol. 32, no. 8, pp. 3710-3722, 2020.
- [18] K. Pillutla et al., "Robust aggregation for federated learning," IEEE Transactions on Signal Processing, vol. 70, pp. 1142-1154, 2022.
- [19] T. Li et al., "Federated optimization in heterogeneous networks," Proceedings of Machine Learning and Systems, vol. 2, pp. 429-450, 2020.
- [20] S. Reddi et al., "Adaptive federated optimization," arXiv preprint arXiv:2003.00295, 2020.
- [21] A. Dudek, "Silhouette index as clustering evaluation tool," In Classification, Data Analysis, and Knowledge Organization, pp. 19-33, 2020.
- [22] T.-M et al., "Measuring the effects of non-identical data distribution for federated visual classification," arXiv preprint arXiv:1909.06335, 2019
- [23] A. Paszke et al., "Automatic differentiation in Pytorch," 2017.
- [24] H. Xiao et al., "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," arXiv preprint arXiv:1708.07747, 2017.
- [25] S. Caldas et al., "Leaf: A benchmark for federated settings," arXiv preprint arXiv:1812.01097, 2018.
- [26] A. Go et al., "Twitter sentiment classification using distant supervision," CS224N project report, Stanford, vol. 1, no. 12, p. 2009, 2009.
- [27] J. Pennington et al., "Glove: Global vectors for word representation," in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532-1543.
- [28] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," 2009.
- [29] D. C. Liu et al., "On the limited memory BFGS method for large scale optimization," Mathematical Programming, vol. 45, 1989, pp. 503-528.
- [30] Y. Cui and J. Zhu, "MChain-SFFL: Multi-Chain Aggregation Privacy-Preserving for Server-Free Federated Learning," IEEE Transactions on Network and Service Management, 2024.



Reza Fotohi is a Ph.D. Candidate in the Faculty of Computer Science and Engineering, Shahid Beheshti University. His research interests include Privacy-Preserving Federated Learning (PPFL). Furthermore, he has published several papers in security domains in highly-ranked journals. His papers have over 1655 citations with a 27 h-index and 33 i10-index. Also, he is recognized as being among the World's Top 2% of Scientists (Stanford University Ranking, 2021 & 2022, [Link](#)). Reza Fotohi has served as a reviewer of several journals, such as IEEE Communications Surveys and Tutorial, IEEE Internet of Things Journal, IEEE Transactions on Aerospace and Electronic Systems, IEEE Transactions on Artificial Intelligence, IEEE Transactions on Cognitive Communications and Networking, IEEE Transactions on Reliability, ACM Transactions on Privacy and Security, Applied Soft Computing, Artificial Intelligence Review, Computers & Security, etc. ([ORCID Link](#)).



Fereidoon Shams Aliee received his Ph.D. in Software Engineering from the Department of Computer Science, Manchester University, in 1996 and his M.S. from the Sharif University of Technology, in 1990. His major interests are Software Architecture, Enterprise Architecture, Service-oriented Architecture, and Software Engineering. He is currently a Professor at the Shahid Beheshti University. Also, Dr. Shams is heading a research group, namely SOEA Lab, [Link](#) at Shahid Beheshti University. ([ORCID Link](#)).



Bahar Farahani received her Ph.D. and Postdoctoral degrees in Computer Engineering from the University of Tehran, and Shahid Beheshti University, respectively. She is an assistant professor at Cyberspace Research Institute, Shahid Beheshti University. She authored several peer-reviewed Conference/Journal papers and book chapters on IoT, Big Data, and AI. Dr. Farahani has served as a Guest Editor of several journals, such as IEEE Internet of Things Journal (IEEE IoT-J), IEEE Transactions on Very Large Scale Integration Systems (IEEE TVLSI), and Elsevier Information Systems. ([ORCID Link](#)).