IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 January 2023, accepted 13 February 2023, date of publication 16 February 2023, date of current version 23 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3246126

RESEARCH ARTICLE

Blockchain-Based Swarm Learning for the Mitigation of Gradient Leakage in Federated Learning

HUSSAIN AHMAD MADNI[1], RAO MUHAMMAD UMER[2,3], AND GIAN LUCA FORESTI[1], (Senior Member, IEEE)

[1]Department of Computer Science and Artificial Intelligence, University of Udine, 33100 Udine, Italy
[2]Institute of AI for Health (AIH), Helmholtz Munich, 85764 Neuherberg, Germany
[3]Department of Computer Science, University of Engineering and Technology, Lahore 39161, Pakistan

Corresponding author: Hussain Ahmad Madni (hamadnig@gmail.com)

This work was supported in part by the PSD-AI Project at the University of Udine, Italy.

ABSTRACT Federated Learning (FL) is a machine learning technique in which collaborative and distributed learning is performed, while the private data reside locally on the client. Rather than the data, only gradients are shared among all collaborative nodes with the help of a central server. To ensure the data privacy, the gradients are prone to the deformation, or the representation is perturbed before sharing, ultimately reducing the performance of the model. Recent studies show that the original data can still be recovered using latent space (i.e., gradient leakage problem) by Generative Adversarial Network and different optimization algorithms such as Bayesian and Covariance Matrix Adaptation Evolution Strategy. To address the issues of data privacy and gradient leakage, in this paper, we train deep neural networks by exploiting the blockchain-based Swarm Learning (SL) framework. In the SL scheme, instead of sharing perturbed or noisy gradients to the central server, we share the original gradients among authenticated (i.e., blockchain-based smart contract) training nodes. To demonstrate the effectiveness of the SL approach, we evaluate the proposed approach using the standard CIFAR10 and MNIST benchmark datasets and compare it with the other existing methods.

INDEX TERMS Blockchain, data privacy, federated learning, gradient leakage, model privacy, Swarm Learning.

I. INTRODUCTION

A large amount of representative dataset is required for a robust and generalized deep learning model. Modern devices generate a large amount of such data, which is the best fit for deep learning models. However, sensitivity and confidentiality are important for user's data that should not be exposed by the deep learning models. For example, a language model is used to improve the text writing and speech recognition, whereas an image model is used to select the photos and videos. A single entity, such as a hospital, school, organization, and individual user, often does not produce enough data to train such models with efficacy. In traditional model training methods, the data have been collected from all entities producing homogeneous data and kept on a central server for training. However, due to emerging privacy concerns [1], [2], Federated Learning (FL) is introduced to mitigate the concerns about data sharing [3]. FL has evolved due to its advantages for the need of today's complex machine learning problems such as continuous learning, hardware efficiency, and data diversity, especially the data produced from multiple locations and resources for a decentralized training process [4], [5]. Unlike traditional methods, FL is considered a privacy-preserving mechanism [6], [7], where the training is conducted in the decentralized and collaborative manner. Moreover, the gradients are shared with a central server, while

The associate editor coordinating the review of this manuscript and approving it for publication was Diego Oliva.

the data reside at the local nodes [8]. However, recent studies [1], [9], [10], [11], [12], [13], [14] show that FL is concerned about the confidentiality and privacy of data where gradients are vulnerable to leakage.

Although, much effort has been put into the defenses against gradient leakage as demonstrated in [15], [16], [17], [18], the problem of gradient leakage still persists in the FL environment. Gradients' leakage is exploited to invert the shared gradients that reproduces the original data on the central server. Initially, a common Differential Privacy (DP) method [19] was introduced to protect gradients. In DP, gradients are clipped, and Gaussian noise is added before sharing with the central server. Subsequently, some sophisticated defense mechanisms have been introduced, as given in [2], [5], [16], [17] to mitigate the problem of gradient leakage. Recently, Li et al. [10] proposed a method to recover the original data from shared gradients.

To solve the problem of gradient leakage in decentralized training, a sophisticated method is required to ensure the data privacy of all training nodes. In this paper, we exploit a newly introduced Swarm Learning (SL) [20] framework based on blockchain networking and edge computing to solve the gradient leakage problem. We use SL for the collaborative training of participating clients. It uses blockchain technology to secure the data and model parameters of each participant based on smart contracts among communicating clients. SL performs better than FL as it aggregates the original gradients of local update obtained from each participant. Moreover, it ensures the privacy of data and model by sharing the data with clients registered through smart contracts. The proposed method mitigates the problem of gradient leakage in FL without sacrificing the overall performance of the model. The structure of SL is similar to FL except that the central server is eliminated. SL leverages blockchain technology to develop a secure and private peer-to-peer network in which every new participant is registered through a smart contract. In SL, all participating nodes train the local model with their local data, and share gradients with the authenticated nodes. These gradients are aggregated by a randomly selected sentinel node that averages the gradients with the Federated Average (FedAvg) algorithm [3], which has also been used as different variations in current FL models [16], [17], [21], [22], [23], [24], [25]. Multiple training schemes with different orientations of data and model, are shown in Fig. 1.

We exploit the SL framework with blockchain technology to mitigate the gradient leakage problem in FL. In the proposed training scheme, a blockchain communication protocol provides a secure means of sharing gradients among authenticated nodes. The overall process of the proposed SL scheme consists of the following steps: (1) A node is registered and authenticated with the smart contract and acquires the model for training. (2) Each node performs local training with its local data and shares the gradients with other nodes (i.e., sentinel node). (3) The sentinel node performs the average aggregation of the gradients received from all nodes and broadcasts it to all nodes. (4) An individual participating node receives updates and evaluates its local model. (5) A global model is obtained after enough communication rounds among all the nodes.

II. RELATED WORKS

A. GRADIENT LEAKAGE IN FEDERATED LEARNING

Initially, FL was considered a privacy preserving framework, but recent work [11] shows that FL is vulnerable to inference attacks. Deep Leakage from Gradients (DLG) [11] approach has been proposed to demonstrate that the original data is reconstructed from the gradients. The dummy gradients are generated with the dummy inputs and labels. Finally, the distance between the dummy and the real gradients is minimized to recover the original data. In a recent study, the Generative Gradient Leakage (GGL) [10] method has been proposed that uses the latent space of the Generative Adversarial Network (GAN) learned from the public dataset. The latent space helps compensate for the loss of features during gradient degradation due to defense implementation. Moreover, adaptive loss and gradient-free optimization methods have been adapted to solve the problem of non-linearity. Another work [1] investigated that the input data, to the fully connected layer in the deep learning model, are reconstructed from gradients. It is also possible to recover the input images in the batches as proposed in the GradInversion [9]. In the GradInversion method, a batch of 8 to 48 images of ImageNet [26] is reconstructed for a larger network such as ResNet50 [27].

B. DEFENSE AGAINST GRADIENT LEAKAGE IN FEDERATED LEARNING

Recently, a number of methods have been proposed to solve the problem of gradient leakage in FL. It has been investigated that the class-wise data representation produced by local updates is the main cause of gradient leakage. The defense, named Soteria [16], has been proposed to solve this problem. In this method, the representation of the data is perturbed by maintaining the convergence of the FedAvg [3] and FL model. However, the accuracy is still effected by the perturbed representation. This work has been extended by [17], which addresses the problem of deterioration of the FL model in exchange for defense implementation. This method is based on the Bounded Local Update Regularization (BLUR) and the Local Update Sparsification (LUS), which limits the norm of local update before applying DP [19]. It improves the model performance, but gradients are still degraded, which eventually affect the performance of global model. In another method proposed in [5], a recurrent language model is trained that maintains differential privacy when applying the FedAvg [3] algorithm. The proposed method ensures the privacy of user-partitioned data for deep models especially Long Short-Term Memory (LSTM) language models. In [28], a method of secure aggregation in FL is proposed in which on-device Gaussian noise is added to data to ensure the distributed DP. In this method, Gaussian noise with a threshold

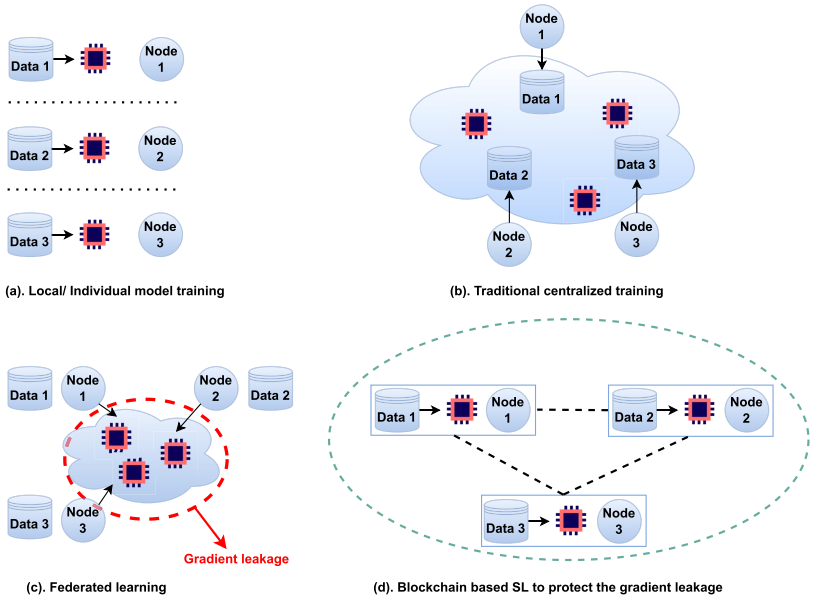

FIGURE 1. Overview of different training schemes: (a) Individual and independent training of a node with the local model and data. (b) Traditional model training with a single collection of data from multiple resources. (c) Federated Learning environment showing collaborative, distributed, and decentralized learning with multiple training nodes that share vulnerable gradients with the central server. (d) Model training, using blockchain-based SL to mitigate the issue of gradient leakage. In SL, the gradients are shared with only authenticated nodes.

is added to the gradients after a local update, before sending them to the central server for aggregation. A voting-based Differentially Private Federated Learning (DPFL) method has been proposed in [29], in which despite of gradient-averaging, voting process is established among the labels of each local model to decrease the communication cost. In this method, secure multi-party communication is implemented on the basis of voting scores.

C. BLOCKCHAIN FOR FEDERATED LEARNING

Blockchain technology provides a secure network and communication protocol. A method proposed in [30] gives a framework for FL using blockchain and Zero-Knowledge Proofs (ZKPs) to preserve the privacy of data. A similar method is proposed in [31] which uses blockchain technology and data can be accessed through access rights. Another similar method proposed in [32] uses blockchain with Hyperledger Fabric to build a secure platform for FL. In this method, local updates are encrypted using Homomorphic Encryption (HE) to protect the data shared with central server. A method named platform-free proof of federated learning (PF-PoFL) [33] has been proposed recently, which provides a secure outsourcing of AI task. In this method, a privacy-preserving algorithm is designed using blockchain for model training in FL.

D. SWARM LEARNING FOR DECENTRALIZED TRAINING

The SL framework [20] was proposed for data confidentiality and decentralized clinical machine learning. It was used for the decentralized model training with private clinical data from different hospitals. Later, it has been used in [34] to detect cancer disease in the same scenario of private and decentralized pathology image data. SL has been applied to solve the problem of decentralized clinical data that cannot be shared for machine learning tasks. We exploit the SL framework for the gradient leakage problem in FL.

III. PROPOSED METHODOLOGY

In the proposed approach, multiple nodes (clients) mutually train a model with the help of a randomly selected sentinel node that is responsible for the aggregation of the gradients. Unlike FL, each node trains a local model with its own local data, and shares the gradients with the authenticated node through a blockchain communication strategy, as shown in Fig. 2. A randomly selected sentinel node is responsible for the aggregation and broadcasting of gradients. Finally, each node evaluates its local model with the gradients received from the sentinel node. Thus, each node sends and receives updates recursively for a specific number of communication rounds until a given limit is reached. If there are N nodes participating with the full data D, in the training process, and a node $i \in N$ has a local data d_i and model m, then the loss of an individual node is expressed as $f_i(m, d_i)$. The objective of the aggregated model is to minimize the aggregated loss of all training nodes, which is formulated as:

$$\min_{m \in R^d} \{f(m, D) = \sum_{i=1}^{N} \frac{s_i}{N} f_i(m, d_i)\} \quad (1)$$

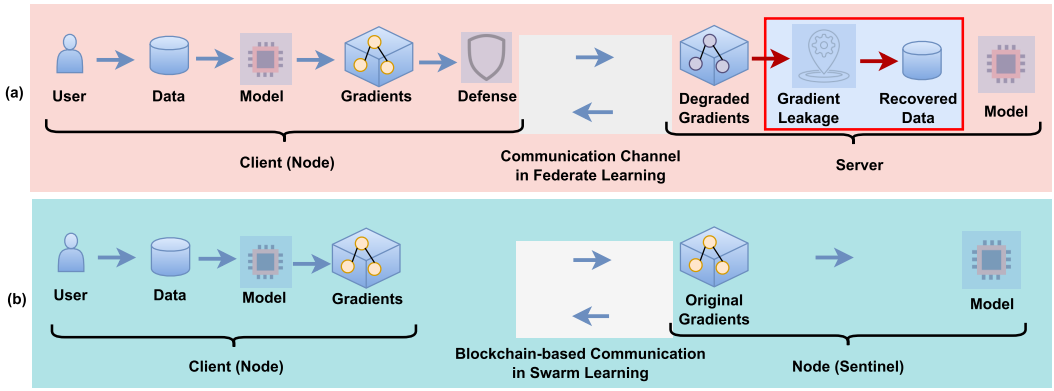

FIGURE 2. Process flow of traditional Federated Learning and newly introduced Swarm Learning. (a). Federated Learning shares perturbed gradients obtained from individual client with the central server. (b). Swarm Learning shares original gradients with the authenticated sentinel node using the blockchain-based secure communication protocol.

where s_i denotes the samples of the node $i \in N$. There are multiple communication rounds among the participating training nodes and the loss is calculated by the average of all communication rounds. An update on the sentinel node for the communication round r is induced as:

$$\Theta^{r+1} = \sum_{i=1}^{N} w_i \Theta_i^r \tag{2}$$

where w_i is the aggregated weight and Θ_i^r represents the local model of the node i.

We exploit SL using blockchain communication protocol to mitigate the gradient leakage problem with a mutually secure contract among nodes. Each participating node in the training process is registered, authorized, and well-known to the network. With the use of the blockchain communication protocol, a new participant needs to enroll with a smart contract, then obtain the training model, and continues the process until it synchronizes with the existing training process. A swarm application interface (i.e., API) is used to merge and exchange model parameters before each training cycle. At an individual node, SL is divided into the application layer and middle-ware in which application layer provides models, while middleware provides the blockchain, Swarm API, and the machine learning platform. We can formulate the output gradients y obtained from an individual node as follows.

$$y = M(x) \tag{3}$$

where $M(x) = \nabla\mathcal{L}(f(x), c)$ is the calculation of gradients from the loss $\mathcal{L}$ generated by the input x and lable c in the model f. Note that in FL, most of the recent methods use a degrading factor ε added to the gradients along with some transformation τ if needed, before sharing these gradients with the central server. In general, this process is expressed as follows:

$$y = \tau(M(x)) + \varepsilon \tag{4}$$

Thus, (4) is also applicable to the proposed approach only without any transformation τ and addition factor ε. The final formulation for the gradients to be averaged at the sentinel node is given below.

$$y = \nabla\mathcal{L}(f(x), c) \tag{5}$$

In the proposed approach, original gradients are protected and shared with authenticated participants that perform better than noisy and perturbed gradients in FL.

IV. EXPERIMENTS AND RESULTS

We conduct a substantial number of experiments to assess the performance of individual and SL model. In the experiments, we use a well-known ResNet18 pre-trained model [27] and CNN-2 model [35] for the classification task. We setup four nodes (clients) having a part of the dataset for the model training environment of blockchain-based Swarm Learning.

A. DATASET

In the experiments, we use a standard CIFAR10 [36] and MNIST [37] datasets, which are publicly available. The CIFAR10 dataset contains 60,000 color images of 32 × 32 size, which are distributed as 50,000 training samples and 10,000 testing samples. This data set consists of 10 classes each having 6000 instances. The MNIST dataset contains a total of 70,000 grayscale images of numbers from 0 to 9 with the size of 28 × 28, which is distributed as 60,000 training and 10,000 testing samples.

For both data sets, we divide all training data equally into four training nodes using different Dirichlet distribution settings (Dir(α)) with different values of α to make the data non-independent and individually distributed (non-IID) as used in [17], [38]. The smaller value of α leads to more heterogeneity in data and vice versa. Heterogeneity in the data results in class imbalance that eventually affects the performance of the model. For an individual node, each model uses fourth part of the whole training set. Test sets of both datasets are used for the evaluation of each individual node as well as SL model.

TABLE 1. A comparison of the proposed approach with the baseline methods.

Model	Distribution (α)	DDGauss [28] (ICML-2021)	DP-FedAVG [5] (ICLR-2018)	BLUR + LUS [17] (CVPR-2022)	AE-DPFL [29] (NIPS-2022)	Ours
CNN-2	0.1	53.55 ± 1.12	53.84 ± 1.04	58.95 ± 0.95	55.79 ± 0.86	$\mathbf{59.22 \pm 0.47}$
	1	58.28 ± 0.96	58.67 ± 0.85	63.74 ± 0.70	60.00 ± 0.57	$\mathbf{66.85 \pm 0.61}$
	10	62.43 ± 0.77	62.25 ± 0.71	65.34 ± 0.52	63.93 ± 0.45	$\mathbf{67.26 \pm 0.51}$
	100	63.80 ± 0.69	63.73 ± 0.64	66.05 ± 0.45	64.51 ± 0.32	$\mathbf{68.93 \pm 0.28}$
ResNet18	0.1	59.37 ± 1.04	59.73 ± 0.96	64.50 ± 0.88	63.11 ± 0.65	$\mathbf{66.48 \pm 0.26}$
	1	63.84 ± 0.89	63.49 ± 0.81	67.27 ± 0.62	65.80 ± 0.51	$\mathbf{71.70 \pm 0.19}$
	10	65.85 ± 0.72	65.64 ± 0.69	68.96 ± 0.54	67.62 ± 0.42	$\mathbf{73.17 \pm 0.17}$
	100	66.74 ± 0.63	66.58 ± 0.60	69.42 ± 0.47	68.39 ± 0.35	$\mathbf{73.08 \pm 0.07}$

TABLE 2. A comparison of the trained model with the proposed blockchain-based SL and individual nodes.

Model	Distribution (α)	Node 1	Node 2	Node 3	Node 4	SWARM
CIFAR10						
ResNet18	0.1	55.18 ± 0.26	52.12 ± 0.92	49.78 ± 0.51	52.21 ± 0.99	$\mathbf{66.48 \pm 0.26}$
	1	61.25 ± 0.15	59.87 ± 0.19	58.97 ± 0.25	59.92 ± 0.24	$\mathbf{71.70 \pm 0.19}$
	10	65.74 ± 0.62	64.27 ± 0.66	65.48 ± 0.44	64.22 ± 0.54	$\mathbf{73.17 \pm 0.17}$
	50	64.85 ± 0.15	65.08 ± 0.24	65.33 ± 0.22	64.84 ± 0.32	$\mathbf{73.15 \pm 0.21}$
	100	65.34 ± 0.20	65.55 ± 0.13	65.42 ± 0.22	64.84 ± 0.36	$\mathbf{73.08 \pm 0.07}$
CNN-2	0.1	46.76 ± 1.02	46.10 ± 0.92	48.04 ± 0.12	48.30 ± 0.86	$\mathbf{59.22 \pm 0.47}$
	1	55.59 ± 0.56	53.11 ± 0.56	53.97 ± 0.26	54.33 ± 0.19	$\mathbf{66.85 \pm 0.61}$
	10	58.28 ± 0.92	56.29 ± 0.73	59.93 ± 0.85	60.27 ± 0.70	$\mathbf{67.26 \pm 0.51}$
	50	59.86 ± 0.20	56.96 ± 0.15	58.78 ± 0.35	59.48 ± 0.43	$\mathbf{67.73 \pm 0.18}$
	100	59.92 ± 0.26	56.03 ± 0.60	62.07 ± 0.19	60.54 ± 0.25	$\mathbf{68.93 \pm 0.28}$
MNIST						
ResNet18	0.1	89.37 ± 0.20	90.74 ± 0.13	87.13 ± 0.14	87.35 ± 0.17	$\mathbf{97.97 \pm 0.31}$
	1	93.55 ± 0.26	91.83 ± 0.14	94.13 ± 0.18	94.53 ± 0.27	$\mathbf{98.65 \pm 0.11}$
	10	96.00 ± 0.69	96.00 ± 0.42	95.77 ± 0.73	95.95 ± 0.39	$\mathbf{98.88 \pm 0.18}$
	50	95.14 ± 0.99	95.97 ± 0.12	95.05 ± 0.12	95.18 ± 0.23	$\mathbf{99.25 \pm 0.07}$
	100	96.17 ± 0.45	95.93 ± 0.43	95.93 ± 1.14	95.99 ± 0.68	$\mathbf{99.72 \pm 0.09}$
CNN-2	0.1	95.18 ± 0.26	95.95 ± 0.06	94.51 ± 1.12	96.92 ± 0.80	$\mathbf{97.83 \pm 0.25}$
	1	97.30 ± 0.92	95.29 ± 0.72	97.36 ± 0.24	98.49 ± 0.22	$\mathbf{97.98 \pm 0.13}$
	10	98.60 ± 0.38	96.58 ± 0.20	98.72 ± 0.43	98.98 ± 0.24	$\mathbf{98.13 \pm 0.10}$
	50	99.10 ± 0.21	98.93 ± 0.27	98.94 ± 0.16	99.04 ± 0.10	$\mathbf{99.26 \pm 0.20}$
	100	99.00 ± 0.25	98.91 ± 0.29	99.08 ± 0.14	98.88 ± 0.14	$\mathbf{99.33 \pm 0.11}$

B. TECHNICAL DETAILS

The proposed approach is implemented with PyTorch. For individual model training, all experiments are performed with Ubuntu 22.04 having NVIDEA GeForce RTX 3090 with 24 GB memory and an i7-8700 CPU with 50 GB memory. For the SL and blockchain environment, we set up four nodes each having Ubuntu 22.04 LTS and i7-8700 CPU with 50 GB memory.

SL environment requires an HPE license server [39], SL repository [40], Secure Shell (SSH) and the docker platform. In our work, we use Linux (Ubuntu 22.04 LTS) as an operating system (OS) where the HP license server is installed on the sentinel node and the participating nodes are connected through SSH and share the learning with authentication certificates. In our experiments, 1 Swarm Network (SN) node, 1 Swarm Operator (SWOP) node, 4 Machine Learning (ML) nodes, 4 Swarm Learning (SL) nodes, and a Swarm Learning Command Line Interface (SWCI) are used for the collaborative and decentralized model training.

C. EVALUATION METRICS

We evaluate the proposed approach with the accuracy as evaluation metric. In the training process, we have imbalance classes for the training set, but balance classes in the test set used for evaluation. Thus, we consider the commonly used accuracy metric for evaluation, and the comparison with other existing methods.

D. COMPARISONS WITH THE STATE-OF-THE-ART METHODS

We are motivated by the recent methods for FL with defenses applied to secure the gradients obtained from local updates of

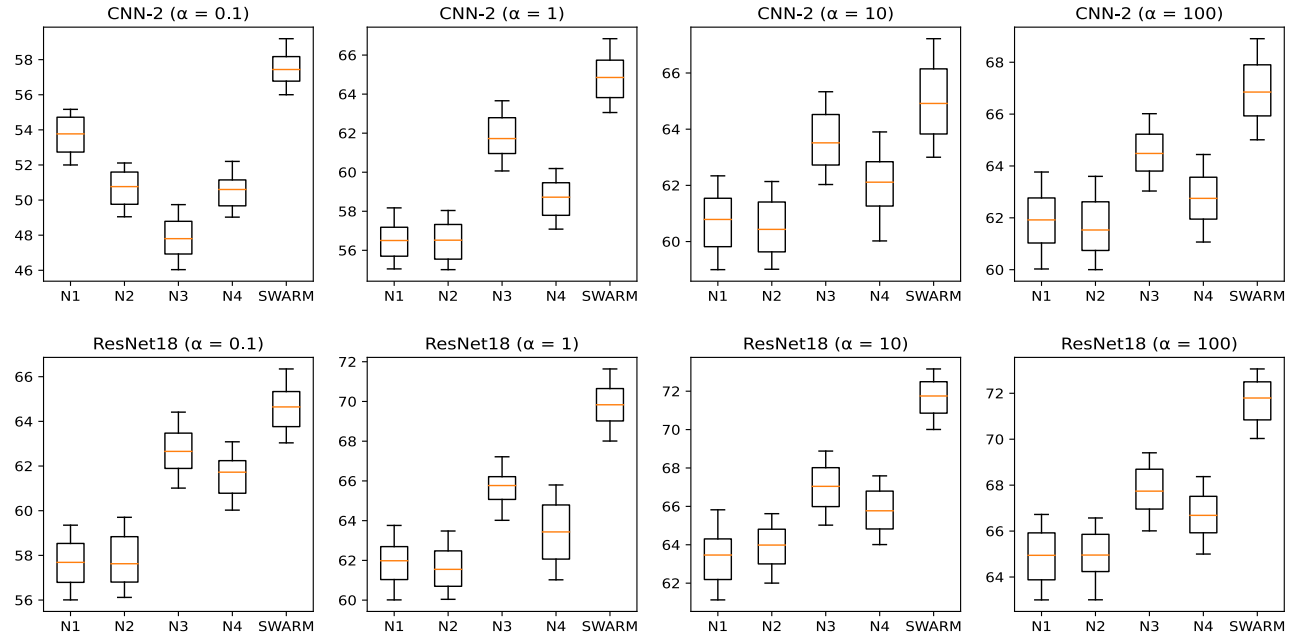

FIGURE 3. Test accuracy of individual and SL model with Non-IID settings of CIFAR10 data using different values of α. CNN-2 and ResNet18 are evaluated on each individual node with their local chunk of data. SL model is obtained by the collaborative training of individual nodes. The model performance increases consistently with a higher α value for both deep models.

individual participating clients. Moreover, there is a trade-off between privacy and model performance in FL when applying some defense, as proposed in the baseline methods. We compare our method with existing baseline methods implementing similar deep learning model in FL with some defense. The existing baseline methods are based on FL with some defense to secure the gradients before sharing with the central server. Table 1 shows the comparison of proposed scheme with the baseline methods [5], [17], [28], [29]. The results of the proposed scheme are compared with different values of α to distribute the non-IID dataset for each individual node. Lower value of α makes the data more heterogeneous, while the higher value leads to homogeneity of the data. Heterogeneity in the data means class imbalances in the training set, whereas homogeneity produces balanced classes in the training set. Thus, the performance of model is increased as the value of α increases. Experiments are repeated four times for each α and we report accuracy with standard deviation (i.e. $mean \pm std$).

It is clear from Table 1 that the proposed scheme consistently outperforms existing state-of-the-art (SOTA) methods for different values of α, and the data distribution. Specifically, using the CIFAR10 [36] dataset, ResNet18 [27] gives better results as compared to CNN-2 [35] model. The proposed scheme also uses the averaging algorithm FedAvg [3] similar to FL. However, it uses blockchain technology to secure the network and protect gradients from leakage.

Unlike FL, Swarm Learning does not require any defense method to degrade or perturb gradients to avoid gradient leakage. However, SL exploits the blockchain protocol to secure the sharing gradients. In the training process of SL, the gradients are shared only with the authenticated nodes through smart contract. The reason for the improved results in SL compared to FL is obvious that transformation and degradation of gradients as in 4, eventually reduces the performance of the FL model. In contrast, in the proposed approach, the original gradients are shared among all the authenticated nodes, which increases the performance of the aggregated model. We argue that our proposed approach is better than the existing FL methods with respect to security, data privacy and model performance.

E. ABLATION STUDY

For the ablation study, we conduct a number of experiments to analyze the effect of data and model learning with the individual and blockchain based SL model. We observe the effectiveness of SL through experiments performed with CNN-2 [35] and ResNet18 [27] models. For the experiments, we use the standard CIFAR10 [36] and MNIST [37] datasets to evaluate the proposed scheme. The experimental results of the individual nodes and SL environment with both deep models trained on both datasets with different value of α, are given in Table 2.

In Table 2, α is the variable that performs the data distribution (i.e., Dir(α)), while Node 1, Node 2, Node 3, and Node 4 are individual clients that locally train the model, each having the fourth part (i.e., local data) of the entire training set with a given value of α. Finally, all the nodes collaboratively train the model in SL environment. When the value of α is lower, performance of the model is decreased due to class imbalance in the local data of individual nodes. Conversely, the performance of the model is increased with the higher value of α because of homogeneity in training data

of individual nodes. It is clear from the results obtained that SL performs better compared to the individual node for both deep networks and datasets.

We further use CIFAR10 [36] dataset distribution with different values of α, and evaluate the CNN-2 [35] and ResNet18 [27] models to check and visualize the performance of individual training and SL training as shown in Fig. 3. The test accuracy is measured against each individual node and the SL model using train data with given α. We conclude two aspects of the results: (1) ResNet18 [27] performs better than the CNN-2 [35] model in both individual and blockchain based SL environment. (2) SL nodes with collaborative training on parts of the data give better results compared to the individual node.

V. CONCLUSION

We train the deep CNN networks by leveraging the advantage of Swarm Learning framework to solve the gradient leakage problem in Federated Learning. During the training phase, we share the gradients of the models with the secure blockchain-based communication strategy among authenticated participating nodes. The experimental results demonstrate the effectiveness of our approach compared to existing methods. The proposed approach mitigates the problem of gradient leakage in collaborative and decentralized training in FL. Moreover, we achieve good results in terms of the accuracy as compared to existing FL-based methods.

Using (2), we perform a naive training and averaging for collaborative SL training. Recently, emerging approaches in FL have been more efficient at improving the model performance. Most recent and efficient averaging methods with privacy-preservation among participants, and data transformation can be explored for the efficient SL merging techniques in future research directions.

REFERENCES

[1] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients—How easy is it to break privacy in federated learning?" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 16937–16947.

[2] G. Kaissis, A. Ziller, J. Passerat-Palmbach, T. Ryffel, D. Usynin, A. Trask, I. Lima Jr., J. Mancuso, F. Jungmann, M.-M. Steinborn, A. Saleh, M. Makowski, D. Rueckert, and R. Braren, "End-to-end privacy preserving deep learning on multi-institutional medical imaging," *Nature Mach. Intell.*, vol. 3, no. 6, pp. 473–484, Jun. 2021.

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

[4] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated-learning-based anomaly detection for IoT security attacks," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2545–2554, Feb. 2022.

[5] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–14.

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

[7] D. Ye, R. Yu, M. Pan, and Z. Han, "Federated learning in vehicular edge computing: A selective model aggregation approach," *IEEE Access*, vol. 8, pp. 23920–23935, 2020.

[8] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *Proc. Neural Inf. Process. Syst.*, 2016.

[9] H. Yin, A. Mallya, A. Vahdat, J. M. Alvarez, J. Kautz, and P. Molchanov, "See through gradients: Image batch recovery via gradinversion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 16337–16346.

[10] Z. Li, J. Zhang, L. Liu, and J. Liu, "Auditing privacy defenses in federated learning via generative gradient leakage," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10132–10142.

[11] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.

[12] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40281–40306, 2022.

[13] D. C. Attota, V. Mothukuri, R. M. Parizi, and S. Pouriyeh, "An ensemble multi-view federated learning intrusion detection for IoT," *IEEE Access*, vol. 9, pp. 117734–117745, 2021.

[14] A. E. Ouadrhiri and A. Abdelhadi, "Differential privacy for deep and federated learning: A survey," *IEEE Access*, vol. 10, pp. 22359–22380, 2022.

[15] Y. Huang, S. Gupta, Z. Song, K. Li, and S. Arora, "Evaluating gradient inversion attacks and defenses in federated learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 7232–7241.

[16] J. Sun, A. Li, B. Wang, H. Yang, H. Li, and Y. Chen, "Soteria: Provable defense against privacy leakage in federated learning from representation perspective," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9311–9319.

[17] A. Cheng, P. Wang, X. S. Zhang, and J. Cheng, "Differentially private federated learning with local regularization and sparsification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10122–10131.

[18] N. Bouacida and P. Mohapatra, "Vulnerabilities in federated learning," *IEEE Access*, vol. 9, pp. 63229–63249, 2021.

[19] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, Q. S. T. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.

[20] S. Warnat-Herresthal, H. Schultze, K. L. Shastry, S. Manamohan, S. Mukherjee, V. Garg, R. Sarveswara, K. Händler, P. Pickkers, N. A. Aziz, and S. Ktena, "Swarm learning for decentralized and confidential clinical machine learning," *Nature*, vol. 594, no. 7862, pp. 265–270, 2021.

[21] P. Guo, P. Wang, J. Zhou, S. Jiang, and V. M. Patel, "Multi-institutional collaborations for improving deep learning-based magnetic resonance image reconstruction using federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2423–2432.

[22] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proc. Mach. Learn. Syst.*, vol. 2, pp. 429–450, Mar. 2020.

[23] Y. Xia, D. Yang, W. Li, A. Myronenko, D. Xu, H. Obinata, H. Mori, P. An, S. Harmon, E. Turkbey, B. Turkbey, B. Wood, F. Patella, E. Stellato, G. Carrafiello, A. Ierardi, A. Yuille, and H. Roth, "Auto-FedAvg: Learnable federated averaging for multi-institutional medical image segmentation," 2021, *arXiv:2104.10195*.

[24] Q.-V. Pham, M. Zeng, T. Huynh-The, Z. Han, and W.-J. Hwang, "Aerial access networks for federated learning: Applications and challenges," *IEEE Netw.*, vol. 36, no. 3, pp. 159–166, May 2022.

[25] B. Ghimire and D. B. Rawat, "Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8229–8249, Jun. 2022.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[28] P. Kairouz, Z. Liu, and T. Steinke, "The distributed discrete Gaussian mechanism for federated learning with secure aggregation," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 5201–5212.

[29] Y. Zhu, X. Yu, Y.-H. Tsai, F. Pittaluga, M. Faraki, M. Chandraker, and Y.-X. Wang, "Voting-based approaches for differentially private federated learning," in *Proc. Conf. Neural Inf. Process. Syst.*, 2022.
[30] Z. Mahmood and V. Jusas, "Implementation framework for a blockchain-based federated learning model for classification problems," *Symmetry*, vol. 13, no. 7, p. 1116, Jun. 2021.
[31] Z. Mahmood and V. Jusas, "Blockchain-enabled: Multi-layered security federated learning platform for preserving data privacy," *Electronics*, vol. 11, no. 10, p. 1624, May 2022.
[32] J. Sun, Y. Wu, S. Wang, Y. Fu, and X. Chang, "Permissioned blockchain frame for secure federated learning," *IEEE Commun. Lett.*, vol. 26, no. 1, pp. 13–17, Jan. 2022.
[33] Y. Wang, H. Peng, Z. Su, T. H. Luan, A. Benslimane, and Y. Wu, "A platform-free proof of federated learning consensus mechanism for sustainable blockchains," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 12, pp. 3305–3324, Dec. 2022.
[34] O. L. Saldanha et al., "Swarm learning for decentralized artificial intelligence in cancer histopathology," *Nature Med.*, vol. 28, pp. 1232–1239, Apr. 2022.
[35] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečnỳ, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–38.
[36] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.
[37] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[38] P. Guo, D. Yang, A. Hatamizadeh, A. Xu, Z. Xu, W. Li, C. Zhao, D. Xu, S. Harmon, E. Turkbey, B. Turkbey, B. Wood, F. Patella, E. Stellato, G. Carrafiello, V. M. Patel, and H. R. Roth, "Auto-FedRL: Federated hyperparameter optimization for multi-institutional medical image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 437–455.
[39] Hewlett Packard Enterprise. (2015). *HPE Software Center*. [Online]. Available: https://myenterpriselicense.hpe.com/
[40] (2022). *Swarm Learning*. [Online]. Available: https://github.com/HewlettPackard/swarm-learning/

HUSSAIN AHMAD MADNI received the B.S. degree in computer system engineering from the Islamia University of Bahawalpur, Pakistan, in 2014, and the M.S. degree in computer science from COMSATS University Islamabad, Pakistan, in 2018. He is currently pursuing the Ph.D. degree in computer science and artificial intelligence with the University of Udine, Italy. His research interests include artificial intelligence, deep learning, and cybersecurity.

RAO MUHAMMAD UMER received the B.S. degree in computer system engineering from the Islamia University of Bahawalpur, Pakistan, in 2014, and the M.S. degree in computer science from the Pakistan Institute of Engineering and Applied Sciences Islamabad, Pakistan, in 2017, and the Ph.D. degree in computer science and artificial intelligence from the University of Udine, Italy, in 2021. His research interests include deep learning and computer vision.

GIAN LUCA FORESTI (Senior Member, IEEE) received the master's and Ph.D. degrees from the University of Genoa, Italy, in 1990 and 1994, respectively. He has been a Full Professor with the University of Udine, since 2001. He is the Director of the AViReS Laboratory. His research interests include artificial intelligence, autonomous aerial systems, computer vision, machine learning, deep learning, and cybersecurity. He is a IAPR fellow.

Open Access funding provided by 'Università degli Studi di Udine' within the CRUI CARE Agreement