Fed-SHARC: Resilient Decentralized Federated Learning based on Reward driven Clustering

Renuga Kanagavelu*†, Chris George Anil‡§, Yuan Wang*†, Huazhu Fu*¶, Qingsong Wei*†, Yong Liu*†, Rick Siow Mong Goh*†

*Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A*STAR), Singapore

‡ Vellore Institute of Technology, Chennai, India

†{renuga_k, wang_yuan, wei_qingsong, liuyong, gohsm}@ihpc.a-star.edu.sg

¶hzfu@ieee.org; §chrisgeorgeanil@gmail.com

***Abstract*—Federated Learning (FL) is an attractive machine learning paradigm that facilitates collaborative machine learning at the decentralized level and produces insightful results. However, in cases where the participating clients have non-IID (non independent and identical data) distributions, it produces sub-optimal results. Furthermore, gradient leakage attacks have been shown in federated learning to be capable of leaking confidential information from global model parameters, hence posing a severe risk to user privacy. In this work, we develop a method called Fed-SHARC, a Secure Hierarchical model Aggregation by clustering in a decentralized federated learning framework that addresses data heterogeneity issues to achieve good performance while guarding the framework against gradient leakage attacks. Secure model aggregation occurs at two stages in this hierarchical approach. Phase-1 involves exploiting differential privacy to aggregate the models of the reward-driven clustered clients. On the other hand, giving each participant with heterogeneous data the same privacy budget will have a significant impact on the performance. As an effective multi-participant budget allocation technique, we suggest an efficient weighted noise injection policy that modifies the privacy budget based on the data distributions of clients. Phase-2 involves selecting a leader from each cluster's highly rewarded clients to take part in the multi-party computation enabled inter-cluster secure model aggregation. Experiments conducted on three benchmark public datasets demonstrate the effectiveness of the proposed Fed-SHARC in terms of privacy and performance. Furthermore, we verify its resilience against gradient leakage attacks.**

***Index Terms*—Decentralised Federated Learning, Reward-driven Clustering, Multi-Party Computation, Weighted noise injection policy, Gradient leakage Attack.**

I. Introduction

Federated learning (FL) [1] facilitates distributed collaborative learning without revealing the original training data, which contributes to General Data Protection Regulation(GDPR) compliance [2]. The participants in centralized federated learning communicate their local model gradients to a server, which then combines all of the received models into a single global model and sends it back to all of the participants. By learning a shared model jointly, the participants benefit from collaborative learning without disclosing the real data. A server used for aggregation could have a single point of failure [3] and high communication costs [4]. There is a chance that this will experience unplanned downtime. To address this problem, a decentralized FL framework [5] was developed. Decentralized federated learning (DFL) is a fully decentralized learning system that operates without a central server and solely depends on local gradient information exchange between participants and their neighbours.

Decentralised Federated learning performs well when client data sets are evenly distributed (i.e., Independently and Identically Distributed (IID)) [6]. On the other hand, when the data are not dispersed uniformly, or non-IID, DFL has challenges. In non-IID datasets, the skewness of the distribution results in diverging gradients. When these diverse models are averaged throughout training rounds, they fail to yield a useful model, and lowers the performance of the shared global model. In addition, recent research studies show that federated learning is vulnerable to gradient leakage attacks (DLG) [7]–[9], which a prospective adversary can acquire participant data from shared model parameters due to the lack of a trust mechanism. Thus, to prevent model parameters from leaking, privacy-preserving techniques in FL have been introduced. Existing federated learning systems use either secure multiparty computation (SMPC) [10], [11], which divides model parameters into secret-shares and shares them among multiple participants, resulting in high communication overhead with the increasing number of participants and also vulnerable to inference attacks, or differential privacy (DP) [12], [13], in which data privacy is preserved by augmenting the data with noise. A privacy budget is a critical component in determining noise level and degree of protection. Severe utility damage will result from existing differentially private approaches [14], [15] that allot the same amount of privacy budget to each participant. Thus, in order to enhance the utility, an effective budget allocation mechanism is required. Aside from that, Dataset condensation is emerging as a privacy-preserving option in FL [31].

With the goal of enhancing performance and resilience, this paper proposes an effective solution, Fed-SHARC, to these problems by using a hierarchical mechanism to enable secure model aggregation. The proposed method groups clients with similar data distributions into clusters. The clusters are trained individually and concurrently after being divided. A secure intra-cluster model is obtained at the first level of the hierarchy by securely aggregating the local models of all the clients that belong to the same cluster utilizing differential privacy. We improve the performance of differential privacy by proposing

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00115

a novel weighted noise injection strategy that will adjust the privacy budget based on the client data distributions. In the proposed method, rewards will be given to clients based on their contribution in intra-cluster model training. More rewards will be given to clients who contribute more in their intra-cluster model training. The most rewarded client from each cluster is chosen to act as the cluster leader. The selected cluster leaders work together and serve as a representative at the second level of hierarchy to complete the MPC enabled inter-cluster model aggregation and produce the global model.

We summarize the contributions of our work below:

- We propose Fed-SHARC, a resilient decentralized federated learning framework that improves performance and robustness through reward-driven clustering and secure hierarchical model aggregation.
- We propose a weighted noise injection strategy, an effective multi-participant budget allocation approach that adjusts the privacy budget according to the client data distributions.
- We demonstrate the effectiveness of our Fed-SHARC through extensive experiments on public datasets. The performance results demonstrate the effectiveness of Fed-SHARC and is not significantly impacted by cluster size.
- Furthermore, we carried out the DLG attack and our approach is shown to successfully resist this attack.

II. Related Work

The research work presented in [16] demonstrates Clustered Federated Learning (CFL), an approach that is applicable to general non-convex objectives, not requiring the number of clusters to be known in advance, and does not require any alterations to the FL communication protocol. The work in [16] also raises a privacy concern in FL by showing how weight updates can be used to determine client data similarity. As a result, these methods might not be appropriate for user-intensive applications. The authors also suggest [17] ways to increase the robustness of a CFL-based architecture in a byzantine environment. However, it is computationally inefficient because it takes several communication rounds to entirely segregate all in congruent random clients. The Iterative Federated Clustering Algorithm (IFCA) [18], which the paper proposes, alternately estimates the cluster identities of the users and optimises model parameters for the user clusters via gradient descent and poses a potential risk as the system still requires clients to provide estimations of their cluster IDs to the central server.

The research presented in [19] suggests a novel multi-center aggregation approach for Federated Learning that concurrently determines the best user and centre matching while simultaneously learning several global models from non-IID user data. In [20] , the authors proposed a hierarchical clustering step (FL+HC) to distinguish client clusters based on how similar their local updates are to the global joint model. However, it depends on a computationally challenging method known as iterative pairwise distance calculation between all clusters. The study [21] also proposes an algorithm for choosing clients in each cluster based on an auction, which tries to address the imbalance in resource consumption brought on by client selection at random. However, auction approach is unsuitable for FL applications on a broad scale. The research work presented in [22] introduces a MPC enabled Federated Learning system named CE-Fed. In particular, the suggested CE-Fed is a mechanism that creates a committee for the purpose of aggregating models with a small number of members and does so by aggregating the global model exclusively among those members rather than across all participants. In order to achieve model aggregation for FL while protecting privacy, the work presented in [23] suggests using Multi-Party Computation (MPC). The paper suggests creating a two-phase system to solve this issue by electing a small committee and making MPC-enabled model aggregation services available to more participants via the committee. However, the committee members are chosen at random.

As opposed to the aforementioned CFL approaches, which take into account each client's participation in FL training, our proposed Fed-SHARC method performs global model aggregation, utilizing a chosen cluster leaders to increase accuracy, decrease communication costs and preserving the privacy by having secure hierarchical model aggregation.

III. Fed-SHARC Framework

In this section, we describe the proposed Fed-SHARC framework as shown in Figure 1, with the objective of improving performance and resilience.

There are several phases involved in implementing the proposed Fed-SHARC framework. (1) Clustering/grouping mechanisms based on client data distributions and proximity to locations; (2) Secure hierarchical model aggregation that involves differentially private Intra-cluster model aggregation with weighted noise injection policy, in which the local models of all FL clients in the same cluster are securely aggregated using DP to form an intra-cluster model; and (3) Inter-cluster model aggregation, where the selected reward-based cluster leaders, jointly aggregate inter-cluster models in a secure way using MPC to generate the overall global model.

A. Clustering Mechanism

As depicted in Figure 1 , the proposed server-less decentralized (peer-to-peer) Clustered Federated Learning framework enables direct communication between clients inside clusters without the use of a central server or coordinator. The proposed clustering mechanism is explained as follows: Initially clients are assigned to a random cluster. Our suggested mechanism will reshuffle them into various groups based on the cosine similarity of their local gradients after a few training rounds. Along with that, the distance among clients is taken into consideration for grouping. The following equation is used to calculate the cosine similarity [24] $Cos\_sim(i,j)$ between the two clients, C_i and C_j.

$$Cos\_sim(i,j) \triangleq \frac{< \Delta w_t^{(c_i)}, \Delta w_t^{(c_j)} >}{||\Delta w_t^{(c_i)}||\,||\Delta w_t^{(c_j)}||} \tag{1}$$

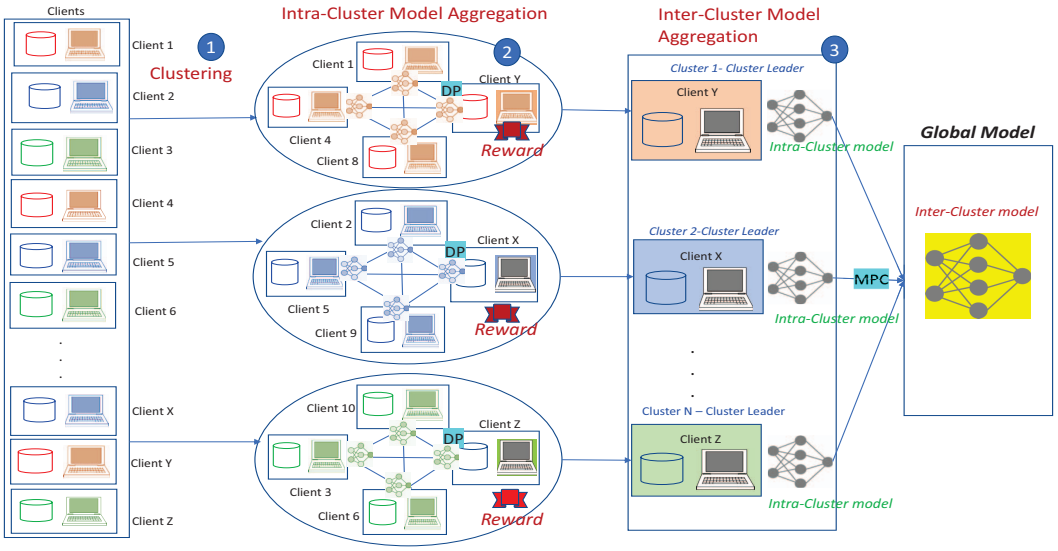

Fig. 1: Fed-SHARC Framework

In the equation 1, $\Delta w_t^{(c_i)}$, $\Delta w_t^{(c_j)}$ represents the parameter used for model updates of the clients C_i and C_j respectively.

B. Differentially private intra-cluster local model aggregation with weighted noise injection policy

The proposed Fed-SHARC aims to cooperatively train a global model with numerous clients. The goal of the training procedure is to gradually minimize the loss function f_{loss}.

$$w_{optimal} = min f_{loss}(w^k, X^k) \tag{2}$$

where $w_{optimal}$ represents the optimal global model parameters; w represents the global model parameters; w^k represents local model parameters; X^k represents the local data.

To achieve high performance and minimal communication costs, the model aggregation is carried out in a hierarchical manner. It consists of two phases: DP enabled Intra-cluster model aggregation and MPC enabled Inter-cluster model aggregation with rewarded cluster leaders.

Each client in the same cluster begins the training process by building their models locally using local datasets. After completing the local training, the cluster's clients within the same cluster work together to cooperatively learn the intra-cluster model. Although the model parameters are shared with the other clients, there is a potential that a malicious user may intercept them and hack into the private information. In order to prevent this, we use DP (ϵ,δ) mechanism. This is achieved by adding quantitative random Gaussian noise to the model parameters following the local training, which is defined as follows [25] :

$$G_\sigma F(x) \triangleq f(x) + N(0, S^2\sigma^2) \tag{3}$$

where S is a sensitivity of function f and $N(0, S^2\sigma^2)$ is the normal distribution with mean 0 and standard deviation $S\sigma$. The level of privacy provided by a differentially private mechanism is controlled by the privacy parameter or privacy budget ϵ.

Algorithm 1 Intra-cluster Model Aggregation with weighted noise injection policy

Input: C_i - Cluster List ; n_i- No. of clients in cluster; m - Neighbouring clients;
Function{**Intra-Cluster.Model.aggregation(C**$_i, n_i, m$)}
for $a \in [1, C_i]$ **do**
  for $b \in [1, n_i]$ **do**
    for $c \in [1, m]$ **do**
      $T \leftarrow local\_training$
    end for
    for $T \in [1, n_i]$ **do**
      Perform clipping
      Add noise σ to the locally-trained model
      share it with 'm' neighbours in the cluster
    end for
  end for
  $W \leftarrow Aggregated\_Model$
end for
return W

1) Weighted noise injection policy: Budget for privacy is a crucial consideration when assessing the degree of security and noise level. The FL clients with non-IID data distributions will perform worse and the model's final convergence will be significantly impacted if the same amount of noise $N(0, S^2\sigma^2)$ is added to their model parameters. To tackle this problem, we design a *weighted noise injection policy* by combining clipping and noise injection, where the data perturbation level is adjusted according to the data scale weights owned by different clients. When applying weighted noise injection policy , clients can add $N(0, Q*S^2\sigma^2)$ where Q represents the data scale weights. Each client aggregates peer clients' private

models to obtain the intra-cluster model.

C. Inter-cluster global model aggregation with rewarded cluster leaders

The trained intra-cluster models from several clusters are ultimately combined to create the final inter-cluster global model. We present the reward-based client selection technique, where one client from each cluster is selected based on the contribution to their intra-cluster training. We take into consideration MPC to safely aggregate the inter-cluster model aggregation since adding more noise (DP) may result in too much noise in the aggregated results, resulting to poor privacy-utility trade-offs.

1) Rewarding Mechanism: The primary objective of the rewarding mechanism is to provide an incentive to the best performing client from each cluster during each communication round. In our protocol, we assign a reward to the best performing client in each cluster, which is considered as the cluster leader of that cluster. In order to choose the cluster leader (best performing client) for a particular cluster, we use the cosine similarity as a performance metric. So, the client closest to the cluster centre is chosen as cluster leader. Since we test our model with a classification problem the cosine similarity is used as a means of measuring the similarity of the output with the expected value. This cosine similarity is a widely accepted metric to use for classification problems. Each cluster will have at most only one cluster leader. The cluster leader from each cluster who has been assigned the higher reward will be selected to take part in the inter-cluster model aggregation. The cluster with higher incentives can take part in the global training more frequently, accelerating convergence and enhancing accuracy.

Following the selection of all cluster leaders, the model parameters of each cluster leader is securely aggregated using MPC to obtain the global model. Each cluster leader splits its model parameters equally into several secret shares. The number of splits depends on the number of cluster leaders in a FL at a given instance. Each cluster leader keeps one share and sends the rest to all the other neighbouring cluster leaders in the FL. This process continues till all the cluster leaders complete the exchange of their shares to all the neighbouring cluster leaders. The cluster leaders then train their respective models on these aggregated shares which then becomes the new global model.

The global model is then sent back to all the cluster leaders. The cluster leaders in turn send it to its cluster's clients, for further training. This is a continuous process and converges to an optimal solution after few global rounds. Thus, the proposed architecture effectively safeguards individual model updates throughout the learning process. Assume the FL system contains G number of clusters. Then, there are G number of cluster leaders. The total number of messages denoted as $Total\_Inter\_cluster\_msgs$ exchanged in the inter-cluster model aggregation among all cluster leaders is given by

$$Total\_Inter\_cluster\_msgs = G * (G \times (G-1)) \times 2 \quad (4)$$

The pseudo code of the inter-cluster model aggregation is shown in Algorithm 2

Algorithm 2 Inter-cluster secure Model Aggregation

Input: C_i - Cluster List ; n_i- No. of clients in cluster; $\hat{C}_i$ Selected cluster leaders;
Function{**Inter-Cluster.Model.aggregation(C**$_i, n_i, \hat{C}_i$)}
for $a \in [1, C_i]$ **do**
  for $b \in [1, n_i]$ **do**
    for $c \in [1, \hat{C}_i]$ **do**
      $T \leftarrow weights\_from\_ClusterLeaders(\hat{C}_i)$
      $W \leftarrow MPC\_Enabled\_GlobalModel\_Aggregation$
    end for
    for $T \in [1, b]$ **do**
      Split the intra-cluster trained model into s secret shares
      share it with $\hat{C}_i$ neighbours
    end for
  end for
  for $a \in [1, m] \& a \neq x$ **do**
    send s to neighbour nodes
    Receive s from neighbour nodes
    $S = sum\_the\_secretshares$
  end for
  $W \leftarrow Aggregated\_Global\_Model$
end for
return W

In order to aggregate the intra-cluster models of all clusters in a distributed manner, the proposed protocol chooses a small group of clients to serve as cluster leaders. As a result, there is a significant reduction in the communication cost.

IV. Experimental Analysis

As a machine learning library, we use TensorFlow 2.0 [26] to assess the performance of our suggested Fed-SHARC framework. It is a free and open-source machine learning and artificial intelligence software library. NetworkX is a Python toolkit [27] for graph and network analysis that is used to create FL clusters in a customized way. MNIST [28], CIFAR-10 [29], and the Fashion-MNIST dataset [30] were three public data sets that we took into consideration. We investigate the performance and effectiveness of IID (Independent and identically distributed) and non-IID datasets in our research. As opposed to the non-IID distribution, which first sorts the data by label before partitioning it across the clients so that each party has a set number of labels and there is no overlap between samples from different clients, the IID distribution involves shuffling the data across all of the clients. Our proposed method is capable of supporting any number of clusters. For model aggregation, we employ FedAvg. In our experiments, we measure performance in terms of accuracy.

A. Accuracy

The network is simulated using Python NetworkX. The following environment is used in experiments: Local-Single Server: Each party uses a single local server running Ubuntu 18.04 with an Intel(R) Xeon(R) CPU i7-7700 V8 (3.60GHz) and 32GB of RAM.

TABLE I: Accuracy Comparisons with existing methods

Method	MNIST IID	MNIST nonIID	CIFAR-10 IID	CIFAR-10 nonIID	Fashion-MNIST IID	Fashion-MNIST nonIID
Fed-SHARC	99.46	98.55	70.68	52.62	93.65	90.08
DFL+MPC	98.0	97.38	65.42	50.83	90.56	88.19
Traditional FL	99.15	98.24	61.22	46.14	90.61	88.81

TABLE II: Accuracy of Fed-SHARC with varying Clip values

Dataset	C=0.5	C=1.3	C=2	C=4	C=6	C=8
MNIST	97.6	98.55	98.53	98.46	98.32	98.12
CIFAR-10	46.6	47.83	52.25	50.64	50.7	50.3
FMNIST	88.01	88.19	88.50	90.01	88.93	88.63

TABLE III: Accuracy of Fed-SHARC with varying noise values

Dataset	σ=1.1	σ=1.3	σ=1.5	σ=1.8
FMNIST-DP (without noise injection policy)	82.5	81.23	80.87	80.04
FMNIST-Fed-SHARC (with noise injection policy)	90.06	89.0	87.83	86.65

The effectiveness of our proposed Fed-SHARC in IID and non-IID distributions is evaluated primarily on the basis of accuracy. For evaluation, we take into account 4 clusters with a total of 10 clients each. All clients participated in intra-cluster local model training, whereas only a small number of cluster leaders—4 in total, one for each cluster—participate in inter-cluster global model aggregation. We evaluate Fed-SHARC's efficacy against the following baseline schemes: Traditional FL, where clients are chosen at random to participate in the training, and decentralized FL with MPC support (DFL+MPC), where all clients participated in the training and communicated their secret shares with one another in order to rebuild the overall joint global model. The accuracy for IID and non-IID distributions are shown in Table I. The proposed Fed-SHARC scheme performs better than that of other baseline methods.This is due to the fact that our proposed technique completely takes into account the local data distribution when choosing clients, resulting in an average increase in the size of the training samples and a higher convergence rate. The DFL+MPC and Fed_AVG schemes did not take into account the local data distributions of various clients when sampling clients, and the distribution of the locally chosen data set varies over rounds, which has an adverse effect on the convergence of the global model.

We examine the impact of changing the clipping bound between 0.5 and 8. The Fed-SHARC findings are depicted in Table II . Fed- SHARC achieves good accuracy on MNIST when c= 1.3 , CIFAR-10 when C= 2 and FMNIST when C=4 by combining gradient clipping with weighted noise injection to per-example gradients. This is because the L2 norm of the gradients changes for different datasets, leading to variable optimum clipping bound settings. The best accuracy can only be achieved with a properly set clipping bound, as an improperly set clipping bound will degrade performance because it will affect both noise variance and gradient information retention. Next, we assess the effects of different noise levels (1.1, 1.3, 1.5, 1.8) with the clipping bound C = 4 . Table III compares the performance of Fed-SHARC with the federated learning without weighted noise injection policy . The per-example gradients' Gaussian noise content is adjusted by the noise scale. The outcomes demonstrate that training performance will suffer from excessive noise addition. we set σ = 1.1 by default in Fed-SHARC, although a smaller noise scale might increase accuracy performance across all datasets.

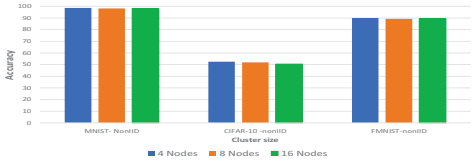

Fig. 2: Impact of varying Cluster size

B. Impact of varying cluster size

We study the impact of the varying cluster size in the framework. The accuracy for MNIST data set over communication rounds with variable cluster size is shown in Figure 2. Our analysis revealed that accuracy is unaffected by the quantity of nodes in each cluster. Faster convergence rate and accuracy were achieved by using our suggested Fed-SHARC technique, which chooses an effective cluster leader for global model aggregation that contributed more training samples.

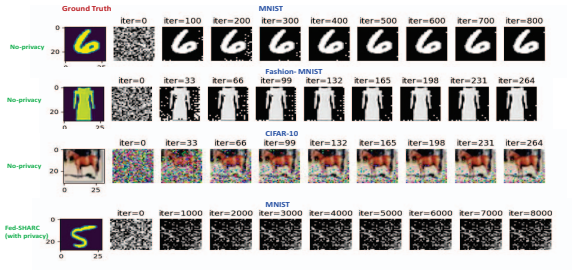

Fig. 3: Resilient to Gradient Leakage

C. Gradient Leakage Attack Resiliency

We measure the resiliency of Fed-SHARC against gradient leakage attacks using MNIST ,CIFAR-10 and and Fashion-MNIST datasets. we carried out the DLG attack [9] on arbitrary images on the above datasets. We use initial random weights and biases with a broader range of values as a random seed to generate the dummy data. In the client local model, the dummy data is fed. Using L2, the gradient loss between

the dummy data's gradient and the real gradient from the client local training is calculated. The dummy data is adjusted and the the same process is repeated until the dummy data gradient matches with the real gradient. Figure 3 provides a visualization comparison of Fed-SHARC with FL without privacy under gradient leakage attack using MNIST dataset, CIFAR-10, Fashion-MNIST datasets. It has been observed that the gradient leakage attacks succeed at the 100^{th} attack iteration for MNIST dataset, 33^{th} attack iteration for Fashion-MNIST dataset, 66^{th} attack iteration for CIFAR-10 dataset in the FL without privacy. The gradient leakage attack fails in Fed-SHARC even after 8000 iterations. We observe that federated learning with no privacy is vulnerable to gradient leakage attacks, while Fed-SHARC can effectively mitigate gradient leakages attack. We use the root mean square deviation of the difference between the reconstructed input and the ground truth to quantify the resiliency. While FL without privacy has a value of 0.2094, Fed-SHARC has a value of 0.7654. The highest value denotes resilience against gradient leakage attacks.

V. Conclusions

In this study, we proposed and implemented an effective secure hierarchical model aggregation by reward-driven clustering. It enables many clients to jointly develop a machine learning model while preserving their privacy and resilience to gradient leakage attacks. Our Fed-SHARC method aggregates the models accurately while protecting privacy. The proposed Fed-SHARC with weighted noise injection policy improves accuracy and resiliency of FL when compared to the uniform privacy budget allocation. We demonstrated the effectiveness of our proposed Fed-SHARC method on several datasets. With extensive evaluation, we show that the Fed-SHARC approach outperforms with high resilience against gradient leakage attacks while offering competitive accuracy performance.

Acknowledgment

This Research is supported by the RIE2025 Industry Alignment Fund – Industry Collaboration Project (IAF-ICP) (Award No: I2301E0020), administered by A*STAR.

References

[1] McMahan, H. B. and Ramage, D,Federated learning: Collaborative machine learning without centralized training data, April 2017. https://ai.googleblog.com/2017/04/ federated-learning-collaborative.html. Google AI Blog.
[2] General Data Protection Regulation, https://gdpr-info.eu/
[3] P. Vanhaesebrouck, A. Bellet, and M. Tommasi. Decentralized collaborative learning of personalized models over networks. In AISTATS, pages 509–517. PMLR, 2017.
[4] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In Proc. Adv. Neural Inf. Process. Syst., pages 5336–5346, 2017.
[5] C. Hu, J. Jiang, and Z. Wang. Decentralized federated learning:a segmented gossip approach. Proc. 1st Int. Workshop Fed. Mach. Learn. User Privacy Data Confidentiality (FML), 2019.
[6] K.Bonawitz, H. E. (n.d.). Towards Federated Learning at Scale: System Design. in Proc. the Conference on Systems and Machine Learning (SysML), 2019.
[7] W. Wei, L. Liu, Y. Wu, G. Su and A. Iyengar, Gradient-Leakage Resilient Federated Learning, in 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS), DC, USA, 2021 pp. 797-807.
[8] Li, Zhuohang, et al. Auditing privacy defenses in federated learning via generative gradient leakage, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.
[9] Zhu, Ligeng and Liu, Zhijian and Han, Song, Deep Leakage from Gradients, Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019.
[10] D. W. Archer, D. Bogdanov, Y. Lindell, L. Kamm, K. Nielsen, J. I.Pagter,N. P. Smart, and R. N. Wright, From keys to databasesreal-world applications of secure multi-party computation, The Computer Journal,Volume 61, Issue 12, pp 17491771, 2018.
[11] P.Bogetoft, D.L.Christensen, I.Damgrd, M.Geisler, T.P.Jakobsen, M.Krigaard, J.D.Nielsen, J.Pagter, M.Schwartzbach, and T.Toft, Secure multiparty computation goes live,Financial Cryptography and Data Security,pp 325 - 343,2009.
[12] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, Learning differentially private recurrent language models, inICLR, 2018.
[13] R. C. Geyer, T. Klein, and M. Nabi, Differentially private federated learning: A client level perspective,arXiv preprint arXiv:1712.07557,2017.
[14] N. Wang, Collecting and analyzing multidimensional data with local differential privacy,in Proceedings of the IEEE 35th International Conference on Data Engineering (ICDE), pp. 638–649, Macao, China, 2019.
[15] O. Thakkar, G. Andrew, and H. B. McMahan, Differentially private learning with adaptive clipping,2019, http://arxiv.org/abs/1905.03871.
[16] F. Sattler, K.-R. Muller, and W. Samek, Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints, IEEE Transactions on Neural Networks and Learning Systems (TNNLS), pp. 1–13, 2020.
[17] F. Sattler, K.-R. Muller, T. Wiegand, and W. Samek, On the byzantine robustness of clustered federated learning, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020, pp. 8861–8865.
[18] Ghosh, J. Chung, D. Yin, and K. Ramchandran, An efficient framework for clustered federated learning, in Advances in Neural Information Processing Systems, vol. 33. Curran Associates, Inc., 2020, pp. 19586–19597.
[19] M. Xie, G. Long, T. Shen, T. Zhou, X. Wang, and J. Jiang, Multi-center federated learning, arXiv preprint arXiv:2005.01026, 2020.
[20] Briggs, Z. Fan, and P. Andras, Federated learning with hierarchical clustering of local updates to improve training on non-IID data, in Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1–9.
[21] R. Lu, W. Zhang, Q. Li, X. Zhong, A. V. Vasilakos, Auction based clustered federated learning in mobile edge computing system, CoRR, 2021. [Online]. Available: https://arxiv.org/abs/2103.07150.
[22] Renuga Kanagavelu, Qingsong Wei, Zengxiang Li, Haibin Zhang, Juniarto Samsudin, Yechao Yang, Rick Siow Mong Goh, Shangguang Wang, CE-Fed: Communication Efficient Multi-Party Computation Enabled Federated Learning, Elsevier Array Journal,2022.
[23] Renuga Kanagavelu, Zengxiang Li, Juniarto Samsudin, Yechao Yang, Feng Yang, Rick Siow Mong Goh, Mervyn Cheah, Praewpiraya Wiwatphonthana, Khajonpong Akkarajitsakul, Shangguang Wang, Two Phase Multi-Party Computation Enabled Privacy-Preserving Federated Learning, CCGRID 2020: 410-419.
[24] Gong Biyao , Xing Tianzhang, Liu Zhidan Wang, Junfeng Liu, Xiuya. (2022). Adaptive Clustered Federated Learning for Heterogeneous Data in Edge Computing. Mobile Networks and Applications. 27. 1-11.
[25] Dwork and Roth, The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science 9 (3–4), pp. 211–407,2014.
[26] Tensorflow, https://www.tensorflow.org/
[27] NetworkX , https://networkx.org/
[28] MNIST Dataset, https://csc.lsu.edu/saikat/n-mnist/.
[29] CIFAR-10 Dataset, https://www.cs.toronto.edu/kriz/cifar.html.
[30] Fashion-MNIST Dataset, https://github.com/zalandoresearch/fashion-mnist.
[31] Yuan Wang, Huazhu Fu, Renuga Kanagavelu, Qingsong Wei, Yong Liu, and Rick Siow Mong Goh, "An Aggregation-Free Federated Learning for Tackling Data Heterogeneity", in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2024.