Privacy-preserving Federated Learning with Hierarchical Clustering to Improve Training on Non-IID Data

Songwei Luo[1] Shaojing Fu[1] Yuchuan Luo[1] Lin Liu[1]
Yanxiang Deng[1] Shixiong Wang [2]

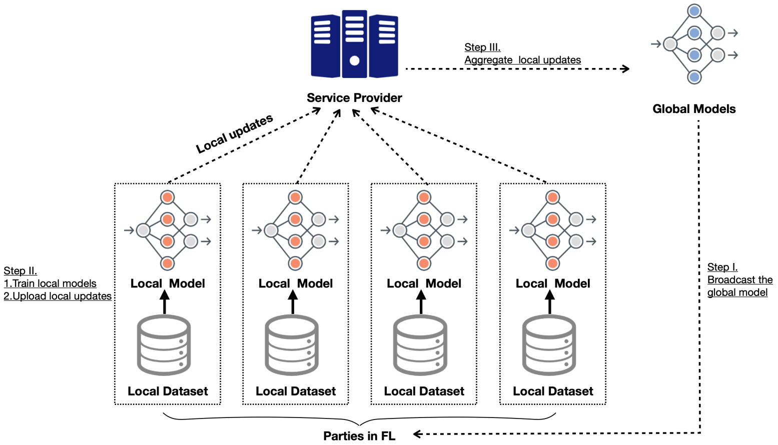
College of Computer, National University of Defense Technology
{luosongwei20,fushaojing,luoyuchuan09,liulin16,dengyanxiang20}
@nudt.edu.cn

Academy of Military Sciences
wsx09@foxmail.com

August 14, 2023

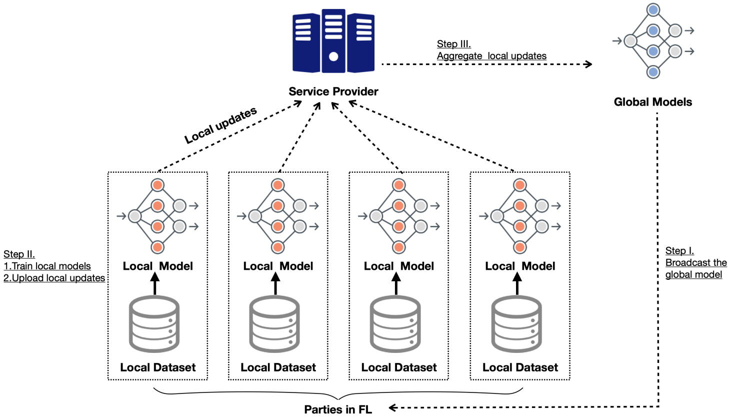
Table of Contents

Federated Leanring



1. SP broadcasts the global model.
2. Parties train local models and upload them.
3. SP aggregates local updates

Federated Learning



- Privacy threats in FL: Local updates will leak the information of raw datasets.
- Data heterogeneity in FL: Non-IID data between parties poses challenges.

Related Works

Propose secure aggregation schemes

- Homomorphic Encryption (HE)
- Secure Multi-party Computation (MPC)
- Differential Privacy (DP)

Improve training performance under Non-IID data.

- Fine-tune the local training process

Our Contributions

- Simultaneously preserve gradients privacy and is compatible with Non-IID scenarios.
- Elaborate protocols for secure distance computation on the secret shared gradients.
- Perform experiments on real-world datasets.

Federated Learning with Hierarchical clustering (FL+HC)

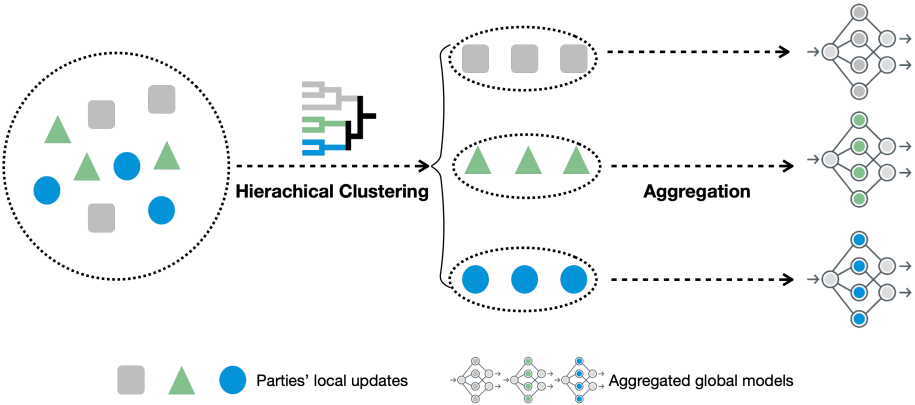
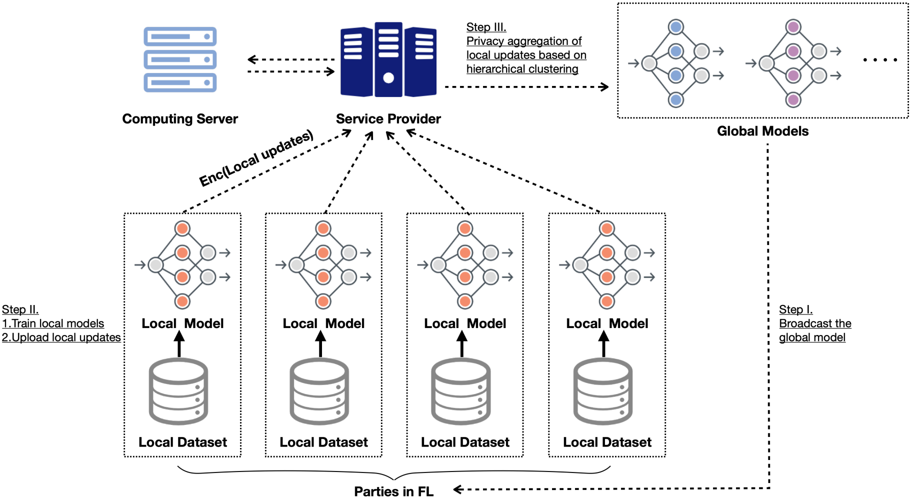


Table of Contents

PPFL+HC System Model



- Service Provider (SP): coordinates the whole FL training process.
- Computing Server (CS): helps SP to performs 2PC.
- Parties in FL: possesses its local data $\mathcal{D}$

Problem Setup

Threat Model

- Honest-but-curious servers.
- No complicit.

Design goals

- Privacy protection
- Accuracy on Non-IID data
- Efficient 2PC protocols

Table of Contents

Our PPFL+HC framework

Phase 1

The Initialization Phase.

- Each participant P_i establishes a private seed key k^{seed} with CS.

Phase 2

The Gradients' Generation and Encoding Phase.

- $Encode(v) = \lfloor 2^s \times v \rfloor \bmod p$,
- $\boldsymbol{g_i} = Encode(\overline{\boldsymbol{g_i}})$
- $\langle \boldsymbol{g_i} \rangle_1 = \boldsymbol{r_i}$, $\langle \boldsymbol{g_i} \rangle_0 = \boldsymbol{g_i} - \boldsymbol{r_i}$

PPFL+HC framework

Phase 3.1 Secure Euclidean Distance of Gradients

Algorithm 1 Secure Euclidean Distance
SED($\langle \boldsymbol{g_i} \rangle, \langle \boldsymbol{g_j} \rangle$) $\rightarrow$ *EDis*

Input: SP holds $\langle \boldsymbol{g_i} \rangle_{\boldsymbol{0}}$ and $\langle \boldsymbol{g_j} \rangle_{\boldsymbol{0}}$, CS holds $\langle \boldsymbol{g_i} \rangle_{\boldsymbol{1}}$ and $\langle \boldsymbol{g_j} \rangle_{\boldsymbol{1}}$. $\mathcal{F}_{\text{SMul}}$ are adopted from Ezpc.
Output: Euclidean distance *EDis* between $\boldsymbol{g_i}$ and $\boldsymbol{g_j}$

1: SP sets $\langle \boldsymbol{z} \rangle_{\boldsymbol{0}} = \langle \boldsymbol{g_i} \rangle_{\boldsymbol{0}} - \langle \boldsymbol{g_j} \rangle_{\boldsymbol{0}}$.
2: CS sets $\langle \boldsymbol{z} \rangle_{\boldsymbol{1}} = \langle \boldsymbol{g_i} \rangle_{\boldsymbol{1}} - \langle \boldsymbol{g_j} \rangle_{\boldsymbol{1}}$.
3: **for** $i \in 1$ **to** m **do** $\triangleright$ m is the dimension of $\boldsymbol{g_i}$
4: SP and CS invoke an instance of $\mathcal{F}_{\text{SMul}}$, in which SP's input is $\langle \boldsymbol{z} \rangle_{\boldsymbol{0}}[i]$ and CS's input is $\langle \boldsymbol{z} \rangle_{\boldsymbol{1}}[i]$. After that SP and CS learn result of multiplication $\langle \boldsymbol{d} \rangle_{\boldsymbol{0}}[i]$ and $\langle \boldsymbol{d} \rangle_{\boldsymbol{1}}[i]$, respectively.
5: **end for**
6: SP sets $\langle EDis^2 \rangle_0 = \sum_{i=1}^{m} \langle \boldsymbol{d} \rangle_{\boldsymbol{0}}[i]$.
7: CS sets $\langle EDis^2 \rangle_1 = \sum_{i=1}^{m} \langle \boldsymbol{d} \rangle_{\boldsymbol{1}}[i]$.
8: CS sends $\langle EDis^2 \rangle_1$ to SP, SP reconstructs $EDis^2 = \langle EDis^2 \rangle_0 + \langle EDis^2 \rangle_1$ and gets *EDis*.
9: **return** Eucliean distance *EDis* at SP.

PPFL+HC framework

Phase 3.2 Secure Manhattan Distance of Gradients

Algorithm 2 Secure Manhattan Distance
SMD($\langle \boldsymbol{g_i} \rangle, \langle \boldsymbol{g_j} \rangle) \rightarrow MDis$

Input: SP holds $\langle \boldsymbol{g_i} \rangle_\mathbf{0}$ and $\langle \boldsymbol{g_j} \rangle_\mathbf{0}$, CS holds $\langle \boldsymbol{g_i} \rangle_\mathbf{1}$ and $\langle \boldsymbol{g_j} \rangle_\mathbf{1}$. $\mathcal{F}_{\text{DRelu}}$ and $\mathcal{F}_{\text{MUX}}$ are adopted from Ezpc.
Output: Manhattan distance $MDis$ between $\boldsymbol{g_i}$ and $\boldsymbol{g_j}$

1: SP sets $\langle \boldsymbol{z} \rangle_\mathbf{0} = \langle \boldsymbol{g_i} \rangle_\mathbf{0} - \langle \boldsymbol{g_j} \rangle_\mathbf{0}$
2: CS sets $\langle \boldsymbol{z} \rangle_\mathbf{1} = \langle \boldsymbol{g_i} \rangle_\mathbf{1} - \langle \boldsymbol{g_j} \rangle_\mathbf{1}$
3: SP and CS invoke $\mathcal{F}_{\text{DRelu}}$ with input $\langle \boldsymbol{z} \rangle$ to learn output $\langle \boldsymbol{y} \rangle^B$
4: SP and CS set $\langle \widetilde{\boldsymbol{y}} \rangle_\mathbf{0}{}^B = \langle \boldsymbol{y} \rangle_\mathbf{0}{}^B$ and $\langle \widetilde{\boldsymbol{y}} \rangle_\mathbf{1}{}^B = \langle \boldsymbol{y} \rangle_\mathbf{1}{}^B \oplus 1$, respectively.
5: SP and CS invoke $\mathcal{F}_{\text{MUX}}$ with input $\langle \boldsymbol{z} \rangle$ and $\langle \boldsymbol{y} \rangle^B$ to learn the positive values $\langle \boldsymbol{d_p} \rangle$
6: SP and CS invoke $\mathcal{F}_{\text{MUX}}$ with input $\langle \boldsymbol{z} \rangle$ and $\langle \widetilde{\boldsymbol{y}} \rangle^B$ to learn the negative values $\langle \boldsymbol{d_n} \rangle$
7: SP sets $\langle MDis \rangle_0 = \sum_{i=1}^{m} \langle \boldsymbol{d_p} \rangle_\mathbf{0}[i] - \sum_{i=1}^{m} \langle \boldsymbol{d_n} \rangle_\mathbf{0}[i]$ ▷ m is the dimension of $\boldsymbol{g_i}$
8: CS sets $\langle MDis \rangle_1 = \sum_{i=1}^{m} \langle \boldsymbol{d_p} \rangle_\mathbf{1}[i] - \sum_{i=1}^{m} \langle \boldsymbol{d_n} \rangle_\mathbf{1}[i]$
9: CS sends $\langle MDis \rangle_1$ to SP and SP reconstructs $MDis = \langle MDis \rangle_0 + \langle MDis \rangle_1$.
10: **return** Manhattan distance $MDis$ at SP.

PPFL+HC framework

Phase 3.3 Secure Hierarchical Clustering of Gradients

Algorithm 3 Secure Hierarchical Clustering of Gradients
$\text{SHC}(\{\langle \boldsymbol{g_1} \rangle, \langle \boldsymbol{g_2} \rangle, ..., \langle \boldsymbol{g_n} \rangle\}) \rightarrow \{c_1, c_2, ..., c_l\}$

Input: SP and CS hold $\{\langle \boldsymbol{g_1} \rangle, \langle \boldsymbol{g_2} \rangle, ..., \langle \boldsymbol{g_n} \rangle\}$ ▷ n is the number of parties
Output: l clusters $\{c_1, c_2, ..., c_l\}$

SP and CS perform random dimensionality reduction with $\{\langle \boldsymbol{g_1} \rangle, \langle \boldsymbol{g_2} \rangle, ..., \langle \boldsymbol{g_n} \rangle\}$, and then obtain: $\{\langle \dot{\boldsymbol{g}}_1 \rangle, \langle \dot{\boldsymbol{g}}_2 \rangle, ..., \langle \dot{\boldsymbol{g}}_n \rangle\}$
for $i \leftarrow 1$ **to** n **do**
    for $j \leftarrow 1$ **to** n **do**
        SP and CS invoke $Dis_{ij} \leftarrow \text{SMD}(\langle \dot{\boldsymbol{g}}_i \rangle, \langle \dot{\boldsymbol{g}}_j \rangle)$ (or $\text{SED}(\langle \dot{\boldsymbol{g}}_i \rangle, \langle \dot{\boldsymbol{g}}_j \rangle)$), then SP holds Dis_{ij}
    end for
end for
$\{c_1, c_2, ..., c_l\} \leftarrow \textsc{Clustering}(\begin{bmatrix} Dis_{1,1} & Dis_{1,2} & \cdots & Dis_{1,n} \\ Dis_{2,1} & Dis_{2,2} & \cdots & Dis_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ Dis_{n,1} & Dis_{n,2} & \cdots & Dis_{n,n} \end{bmatrix})$ ▷ Hierarchical clustering results based on a precomputed distance matrix

Phase 4 Gradients' Aggregation and Broadcast

Algorithm 4 Secure Global Gradients Broadcast
SGB($\langle \boldsymbol{G_x} \rangle$) $\rightarrow \boldsymbol{G_x}$

Input: SP and CS hold party P_i's global gradients $\langle \boldsymbol{G_x} \rangle$.
Output: Party P_i gets the corresponding global gradients $\boldsymbol{G_x}$

1: P_i and CS generates $\boldsymbol{r_i'}$ = PRG(k_i^{seed}) with the same dimension as $\boldsymbol{G_x}$ ▷ Identical k_i^{seed} guarantee the consistency of $\boldsymbol{r_i'}$ in P_i and CS
2: CS masks $\langle \boldsymbol{G_x} \rangle_1$ as follows: $\langle \widehat{\boldsymbol{G_x}} \rangle_1 = \langle \boldsymbol{G_x} \rangle_1 + \boldsymbol{r_i'}$
3: CS sends $\langle \widehat{\boldsymbol{G_x}} \rangle_1$ to SP, then SP reconstructs masked global gradients $\widehat{\boldsymbol{G_x}}$ as follows: $\widehat{\boldsymbol{G_x}} = \langle \boldsymbol{G_x} \rangle_0 + \langle \widehat{\boldsymbol{G_x}} \rangle_1$
4: SP sends $\widehat{\boldsymbol{G_x}}$ to P_i, then P_i unmask the global gradients as follow: $\boldsymbol{G_x} = \widehat{\boldsymbol{G_x}} - \boldsymbol{r_i'}$

Table of Contents

Experiment Setup

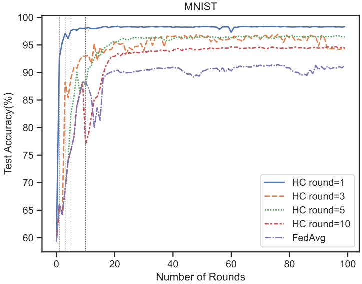
Datasets

- MNIST dataset
- CIFAR-10 dataset

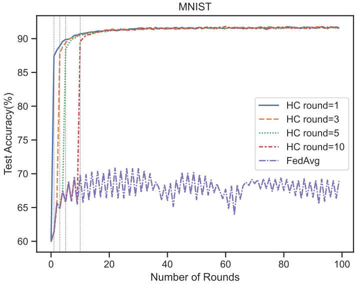
Non-IID Settings

- Pathological Non-IID
- Label-swapped Non-IID

Impact of the different Non-IID settings (MNIST)



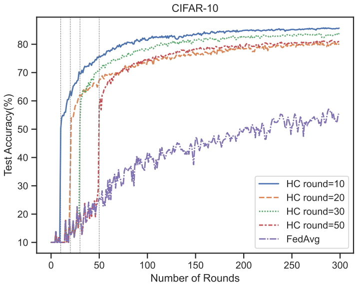
(a) MNIST(Pathological Non-IID)



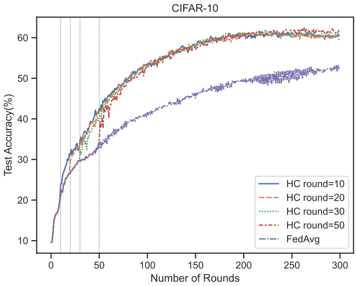
(b) MNIST(Label-swapped Non-IID)

Figure 1: Impact of different Non-IID settings and different HC rounds on test accuracy in MNIST dataset

Impact of the different Non-IID settings (CIFAR-10)



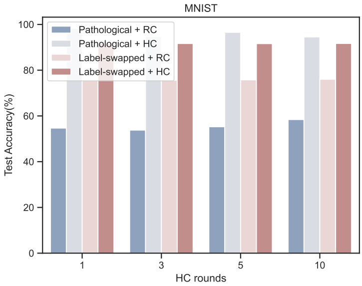
(a) CIFAR10(Pathological Non-IID)



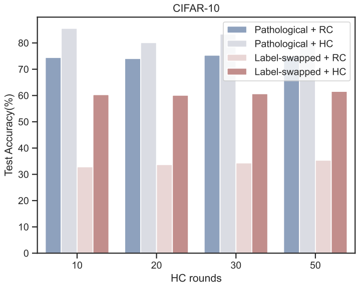
(b) CIFAR10(Label-swapped Non-IID)

Figure 2: Impact of different Non-IID settings and different HC rounds on test accuracy in CIFAR-10 dataset.

Comparing with Random Clustering



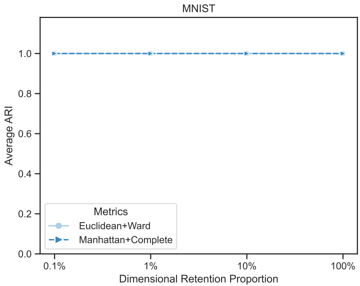
(a) MNIST Final Test Accuracy



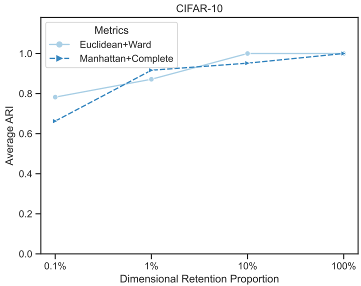
(b) CIFAR10 Final Test Accuracy

Figure 3: Comparing final test accuracy with Random Clustering (RC) in different Non-IID settings (Pathological and Label-swapped)

Impact of different metrics



(a) MNIST Average ARI



(b) CIFAR10 Average ARI

Figure 4: Different dimensional retention proportions' average ARI

Table of Contents

Conclusion

In this paper, we introduce PPFL+HC, a novel FL framework that achieves

Pros

- Full privacy protection of gradients and high accuracy over Non-IID data.
- Efficient cryptographic protocols to implement secure hierarchical clustering over 2PC.
- Evaluation on two real-world datasets over two classic Non-IID settings

Cons

- Inherits constraints of FL+HC.
- Two non-colluding servers.

Thanks for listening!