Aggregation Rules of Defenses in Federated Learning

Xiaobing Chen
Jul 7, 2021

Outlines

1. Introduction

Attack schemes;

Capabilities of adversary;

Defenses based on aggregation rules;

2. Defense Methods based on Altering Aggregation Rules

Method 1: ARFL[1];

Method 2: FL-IOWA-DQ[2];

3. Extra Experiments

Compare ARFL, FL-IOWA-DQ in the same setting;

Test their performance in extreme non-i.i.d distribution;

[1]. Li, Shenghui, et al. "Auto-weighted Robust Federated Learning with Corrupted Data Sources." *arXiv preprint arXiv:2101.05880* (2021).
[2]. Rodríguez-Barroso, Nuria, et al. "Dynamic federated learning model for identifying adversarial clients." *arXiv preprint arXiv:2007.15030* (2020).

1 Introduction

Federated learning (FL) is a machine learning setting where many clients collaboratively train a model under the orchestration of a central server, while keeping the training data decentralized.

“

Kairouz, Peter, et al. "Advances and open problems in federated learning." arXiv preprint arXiv:1912.04977 (2019).

Introduction

Attack Risks of FL

- Vulnerable spots: the distributed nature, architectural design, and data constraints ;
- Adversarial Attacks: attacks on **model performance** or on data inference

Attack Schemes

Training-time attacks(poisoning)	**Data poisoning*** • the adversary alters the client datasets used to train the model
	Model update poisoning • the adversary alters model updates sent to the server.
Inference-time attacks(evasion)	model evasion attacks • the adversary alters the data used at inference-time

Data poisoning* will be mainly discussed in the further context.

Introduction

Capabilities of adversaries

Model inspection • Whether the adversary can observe the model parameters.	• Black box: the adversary has no ability to inspect the parameters of the mode. • Stale white box: the adversary can only observe the model while participating in the global aggregation. • White box: the adversary can observe the model all the time.
Participant collusion • Whether multiple adversaries can coordinate an attack.	• Non-colluding: there is no capability for participants to coordinate an attack. • Cross-update collusion: past client participants can coordinate with future participants on attacks to future updates to the global model. • Within-update collusion: current client participants can coordinate on an attack to the current model update.
Adaptability • Whether an adversary can alter the attack parameters as the attack progresses.	• Static: the adversary must fix the attack parameters at the start of the attack and cannot change them. • Dynamic: the adversary can adapt the attack as training progresses.

Stale white box, non-colluding and static are considered in further context.

Introduction

Defense schemes

- Many existing defense methods in distributed datacenter learning and centralized learning are hard to be deployed in federated learning, such as, data sanitization and network pruning.
- Designing robust aggregators by replacing the averaging aggregation in the server is one direction that has been widely explored.
- Dynamic Quantifier federated aggregation operator (FL-IOWA-DQ) and Auto-weighted Robust Federated Learning (ARFL) will be introduced.

2

Defense Methods based on Altering Aggregation Rules

Aggregation of FedAvg

TABLE I: Summary of main notations

N	total number of the clients
$\bar{L}_i$	the local training loss of ith client
D_i	ith local training dataset
m_i	the size of ith local training dataset
M	the size of total training dataset
$\boldsymbol{\omega}_t$	the global weight in time t
α_i	weight assigned to ith local update
$\delta_i(t)$	local update of ith client in time t

FedAvg:

$$\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t + \sum_{i=1}^{N} \alpha_i \delta_i(t)$$
$$\delta_i(t) = \boldsymbol{\omega}_t(D_i) - \boldsymbol{\omega}_t$$

$$\alpha_i = \frac{m_i}{M}$$
$$s.t. \quad \boldsymbol{\alpha} \in \mathbb{R}_+^n, \mathbf{1}^\top \boldsymbol{\alpha} = 1,$$

Larger local datasets, higher weights.

2.1 Auto-weighted Robust Federated Learning (ARFL)

Aggregation of ARFL

$$\min_{\boldsymbol{w},\boldsymbol{\alpha}} \sum_{i=1}^{N} \alpha_i \hat{\mathcal{L}}_i(\boldsymbol{w})$$

Objective for FedAvg

$$\min_{\boldsymbol{w},\boldsymbol{\alpha}} \sum_{i=1}^{N} \alpha_i \hat{\mathcal{L}}_i(\boldsymbol{w}) + \frac{\lambda}{2}\sum_{i=1}^{N} \frac{\alpha_i^2}{m_i}$$

Objective for ARFL

- Add **L2 regularization** based on data size

Aggregation of ARFL

Theorem 2. *For any* $\boldsymbol{w}$*, when* $\lambda > 0$ *and* $\{\hat{\mathcal{L}}_i(\boldsymbol{w})\}_{i=1}^N$ *are sorted in increasing order:* $\hat{\mathcal{L}}_1(\boldsymbol{w}) \leq \hat{\mathcal{L}}_2(\boldsymbol{w}) \leq \ldots \leq \hat{\mathcal{L}}_N(\boldsymbol{w})$*, by setting:*

$$p = \underset{k}{\operatorname{argmax}}\{1 + \frac{M_k(\overline{\mathcal{L}}_k(\boldsymbol{w}) - \hat{\mathcal{L}}_k(\boldsymbol{w}))}{\lambda} > 0\}, \tag{7}$$

where $M_k = \sum_{i=1}^k m_i$*,*

$$\overline{\mathcal{L}}_k(h) = \frac{\sum_{i=1}^k m_i \hat{\mathcal{L}}_i(\boldsymbol{w})}{M_k} \tag{8}$$

is the average loss over the first k *clients that have the smallest empirical risks. Then the optimal* $\boldsymbol{\alpha}$ *to the problem* $\boxed{(6)}$ *is given by:*

$$\alpha_i(\boldsymbol{w}) = \frac{m_i}{M_p}[1 + \frac{M_p(\overline{\mathcal{L}}_p(\boldsymbol{w}) - \hat{\mathcal{L}}_i(\boldsymbol{w}))}{\lambda}]_+, \tag{9}$$

where $[\cdot]_+ = max(0, \cdot)$.

Lower training loss, higher weights.

Aggregation of ARFL

Algorithm 1 Optimization of ARFL

Server executes:
1: Initialize $\boldsymbol{w}_0, \hat{\mathcal{L}}, \boldsymbol{\alpha}$
2: **for** each round $t = 1, 2, \ldots$ **do**
3: Select a subset S_t from N clients at random
4: Broadcast the global model $\boldsymbol{w}_t$ to selected clients S_t
5: **for** each client $i \in S_t$ **in parallel do**
6: $\boldsymbol{w}_{t+1}^i, \hat{\mathcal{L}}_i \leftarrow \text{ClientUpdate}(i, \boldsymbol{w}_t)$
7: **end for**
8: Update $\boldsymbol{w}_{t+1}$ according to Eq. (11)
9: Update $\boldsymbol{\alpha}$ according to Theorem 2
10: **end for**
11:
ClientUpdate($i, \boldsymbol{w}$): *// Run on client i*
12: $\mathcal{L}_i \leftarrow$ (evaluate training loss using training set)
13: $\mathcal{B} \leftarrow$ (split local training set into batches of size B)
14: **for** each local epoch i from 1 to E **do**
15: **for** batch $b \in \mathcal{B}$ **do**
16: $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \nabla \ell(\boldsymbol{w}; b)$
17: **end for**
18: **end for**
19: return $\boldsymbol{w}$ and $\hat{\mathcal{L}}_i$

- Line 9: weights are maintained by the server; only updates the losses from those selected clients while keeping the others unchanged
- Line 12: clients submit the training losses, together with updates

Experiments: Datasets

	CIFAR-10[1]	FEMNIST[2]	Shakespeare
#Classes:	10;	62;	80;
#Clients:	100;	1039;	71;
#Samples:	60,000;	236,500;	417,469;
i.i.d:	Yes;	No;	No;
Model:	CNN;	CNN;	LSTM;
Task	Image classification	Image classification	Next-character prediction
#Participants(ratio)	20(20%);	32(3%);	16(22%);

i.i.d: “each local client has approximately the same amount of samples and in proportion to each of the classes. ”
Non i.i.d in source: each speaking role or writer is treated as a client.

[1]. Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, 2009.
[2]. Caldas, Sebastian, et al. "Leaf: A benchmark for federated settings." arXiv preprint arXiv:1812.01097 (2018).

Experiments: Models

- **FedAvg**: The standard Federated Averaging aggregation approach that just calculates the weighted average of the parameters from local clients.
- **RFA[1]**: A robust aggregation approach that minimizes the weighted Geometric Median(GM) of the parameters from local clients.
- **MKrum (Multi-Krum)[2]**: A Byzantine tolerant aggregation rule, which computes a distance-related score for each update, and then add the averaged updates with high scores to the model.
- **CFL[3]**: A Clustered Federated Learning (CFL) approach that separates the client population into different groups based on the pairwise cosine similarities between their parameter updates. Updates in benign group are aggregated.

[1]. Pillutla, Krishna, Sham M. Kakade, and Zaid Harchaoui. "Robust aggregation for federated learning." arXiv preprint arXiv:1912.13445 (2019).
[2]. Blanchard, Peva, et al. "Machine learning with adversaries: Byzantine tolerant gradient descent." Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017.
[3]. Sattler, Felix, et al. "On the byzantine robustness of clustered federated learning." ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020.

Experiments: Attack Schemes

- **Label Shuffling:** the labels of all samples are shuffled randomly in each corrupted client
- **Label Flipping:** the labels of all samples are switched to a random one in each corrupted client
- **Noisy Clients:** for CIFAR-10 and FEMNIST, first normalize the inputs into [0, 1], then add Gaussian noise following N(0, 0.7) to the values, at last normalize them again; for Shakespeare, randomly select half of the characters and shuffle them so that the input sentence might be disordered
- Two corruption level: 30% or 50% are malicious clients.

Experiments: Results

Table 2. Averaged test accuracy over five random seeds for FedAvg, RFA, MKRUM, CFL and ARFL in four different scenarios

CIFAR-10	Clean	Shuffling		Flipping		Noisy	
Corr. Per.	-	30%	50%	30%	50%	30%	50%
FedAvg	73.59 ± 0.44	61.17 ± 1.81	47.00 ± 7.51	65.01 ± 2.38	51.75 ± 7.75	**73.75 ± 0.49**	73.61 ± 0.53
RFA	71.36 ± 0.47	57.86 ± 3.22	40.26 ± 9.14	55.47 ± 5.17	40.91 ± 11.06	73.74 ± 0.52	**73.69 ± 0.63**
MKRUM	67.03 ± 0.93	59.27 ± 9.34	52.32 ± 14.90	60.21 ± 5.73	47.96 ± 10.25	73.41 ± 0.69	73.49 ± 0.49
CFL	71.68 ± 0.36	52.54 ± 1.71	50.29 ± 1.95	52.87 ± 1.07	51.67 ± 0.92	54.97 ± 1.14	55.26 ± 1.96
ARFL	73.42 ± 0.40	**71.68 ± 1.01**	**69.66 ± 0.73**	**71.78 ± 0.53**	**70.25 ± 0.56**	73.48 ± 0.56	73.29 ± 0.79

FEMNIST	Clean	Shuffling		Flipping		Noisy	
Corr. Per.	-	30%	50%	30%	50%	30%	50%
FedAvg	82.12 ± 0.20	61.91 ± 21.33	39.69 ± 20.80	70.19 ± 10.17	48.53 ± 23.49	79.94 ± 0.36	78.27 ± 0.47
RFA	82.11 ± 0.32	74.36 ± 7.52	52.02 ± 22.51	73.80 ± 7.49	50.75 ± 19.91	80.45 ± 0.30	79.21 ± 0.41
MKRUM	79.38 ± 0.41	57.51 ± 21.17	42.40 ± 24.84	78.57 ± 4.83	67.10 ± 7.35	**81.52 ± 0.53**	**79.80 ± 0.22**
CFL	82.18 ± 0.30	81.24 ± 0.47	36.03 ± 36.38	81.22 ± 0.36	65.54 ± 26.94	80.13 ± 0.70	79.21 ± 0.64
ARFL	**82.32 ± 0.19**	**81.60 ± 0.31**	**81.35 ± 0.43**	**81.87 ± 0.22**	**81.30 ± 0.24**	80.71 ± 0.28	79.40 ± 0.45

Shakespeare	Clean	Shuffling		Flipping		Noisy	
Corr. Per.	-	30%	50%	30%	50%	30%	50%
FedAvg	53.80 ± 0.33	51.98 ± 0.48	47.70 ± 4.96	52.08 ± 0.39	41.85 ± 16.18	51.85 ± 0.56	50.43 ± 1.19
RFA	**54.27 ± 0.41**	50.16 ± 1.28	32.49 ± 13.81	50.50 ± 1.02	23.84 ± 21.78	**52.17 ± 0.50**	50.69 ± 1.04
MKRUM	50.81 ± 0.85	40.38 ± 7.44	24.46 ± 6.88	44.95 ± 2.43	16.11 ± 15.46	48.19 ± 0.40	45.67 ± 0.46
CFL	54.01 ± 0.34	49.76 ± 4.47	43.68 ± 12.68	51.09 ± 1.36	37.30 ± 19.76	51.98 ± 1.03	50.38 ± 1.39
ARFL	53.52 ± 0.32	**52.85 ± 0.49**	**51.61 ± 0.68**	**52.82 ± 0.48**	**51.74 ± 0.69**	52.09 ± 1.27	**50.98 ± 0.75**

2.2 Dynamic Quantifier federated aggregation operator (FL-IOWA-DQ)

Aggregation of FL-IOWA-DQ

$$\alpha_i = Q_{a,b,c,y_b}\left(\frac{i}{N}\right) - Q_{a,b,c,y_b}\left(\frac{i-1}{N}\right)$$

$$Q_{a,b,c,y_b}(x) = \begin{cases} 0 & 0 \le x \le a \\ \dfrac{x-a}{b-a} \cdot y_b & a \le x \le b \\ \dfrac{x-b}{c-b} \cdot (1-y_b) + y_b & b \le x \le c \\ 1 & c \le x \le 1 \end{cases}$$

where α_i is corresponding to the update with **ith highest accuracy in validation dataset**.
b and c are dynamic; b = 0.2c;
c is the portion of clients whose distance to highest acc is least than ¾ the maximum distance between lowest and highest one.

Higher accuracy, higher weights.

Aggregation of FL-IOWA-DQ

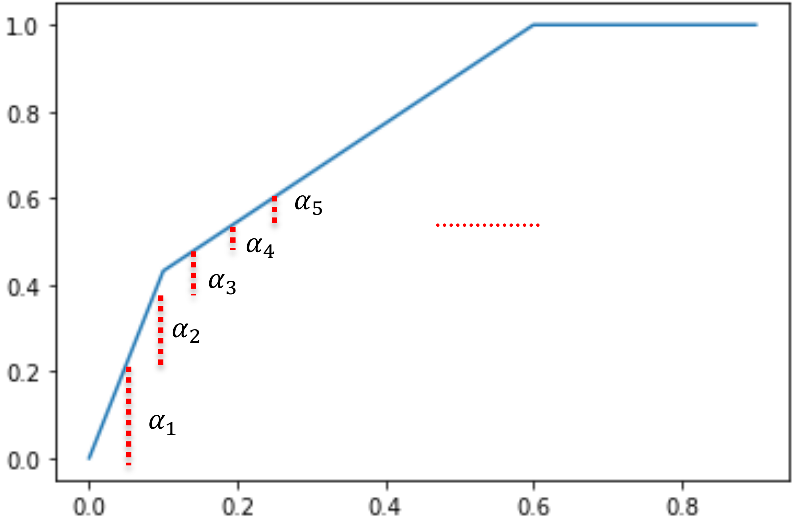

Fig. 1. An example of IOWA-DQ. a = 0, b = 0.2c, yb = 0.4.

a=0: all clients with high acc are included in the aggregation;
yb: total weights assigned to updates with top acc;
c: the portion of clients with non-zero weights

Experiments: Models

<table>
<tr><th></th><th>a</th><th>b</th><th>c</th><th>yb</th></tr>
<tr><td>FL-IOWA-DQ-0.4</td><td rowspan="5">0</td><td rowspan="2">0.2*c</td><td rowspan="2">dynamic</td><td>0.4</td></tr>
<tr><td>FL-IOWA-DQ-0.75</td><td>0.75</td></tr>
<tr><td>IOWA-SQ-0.4</td><td rowspan="2">0.2</td><td rowspan="2">0.8</td><td>0.4</td></tr>
<tr><td>IOWA-SQ-0.75</td><td>0.75</td></tr>
<tr><td>FL-AL-80</td><td colspan="2">0.8</td><td>1</td></tr>
</table>

Table 3. Models with different configurations

- **Besides above, FedAvg and weighted FedAvg(W-FedAvg) are also used**

Experiments: Models

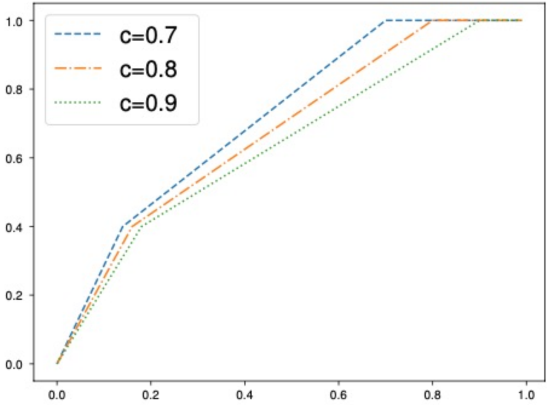

Fig. 2: FL-IOWA-DQ-0.4
with different c.

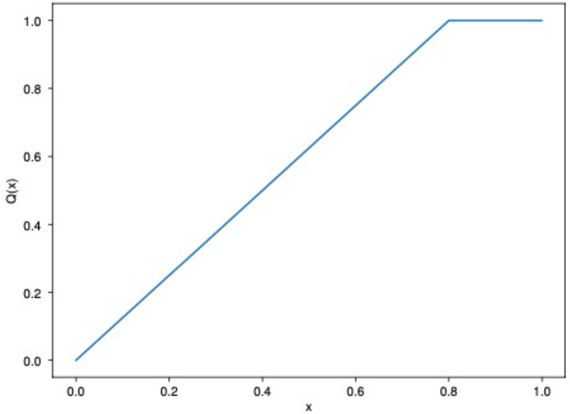

Fig. 3: FL-AL-80

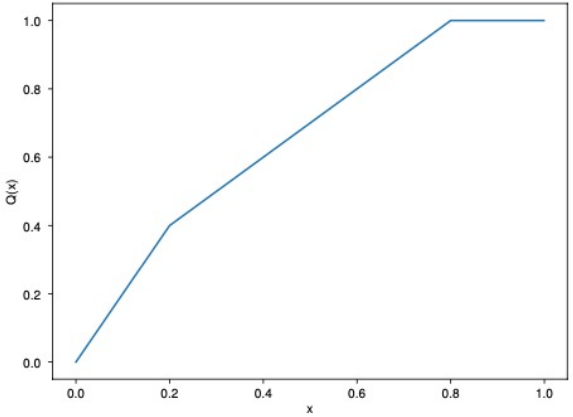

Fig. 4: FL-IOWA-SQ-0.4

Experiments: Datasets

	EMNIST	Fashion MNIST
#Classes:	10;	
#Clients:	20/50;	
#Training:	200,000;	50,000;
#Validation:	40,000;	10,000
#Test:	40,000;	10,000
i.i.d:	No;	
Model:	CNN with two convolutional layers;	

- **Non i.i.d in size: "randomly assign instances of a reduced number of labels to each client "**
- **Validation dataset: follows the same distribution of the training subsets**

Experiments: Attack Scheme

Definition 3.1 (Adversarial client): Let $C_i \in \{C_1, ..., C_n\}$ be an arbitrary client of a FL environment whose original training dataset is $D_i = \langle x_i^l; y_i^l \rangle$, where x_i^l is the sample data and y_i^l the label. We say that C_i is an **adversarial client** if it uses the altered dataset D_i' as training dataset with

$$D_i' = \langle x_i^l; y_i^{\sigma(l)} \rangle,$$

where σ is a random permutation.

Attach Scheme: **Label Shuffling**

Experiments: Results

1. AD Scenario: 10% of clients are malicious, 2 out of 20 or 5 out of 50.

Table 4. Accuracy of Models in AD Scenario.

	EMNIST		Fashion-MNIST	
	20 clients	**50 clients**	**20 clients**	**50 clients**
FL-FedAvg	0.9826	0.9791	0.8661	0.8439
FL-W-FedAvg	0.9776	0.9774	0.8699	0.8321
FL-AL-80	0.9832	0.9803	0.8708	0.8469
FL-IOWA-SQ-0.4	0.9863	0.9824	0.8747	0.8541
FL-IOWA-SQ-0.75	0.9883	0.9869	0.8656	0.8671
FL-IOWA-DQ-0.4	0.9870	0.9886	**0.8782**	0.8694
FL-IOWA-DQ-0.75	**0.9900**	**0.9898**	0.8680	**0.8729**

Experiments: Results

1. AD Scenario: 10% of clients are malicious, 2 out of 20 or 5 out of 50.

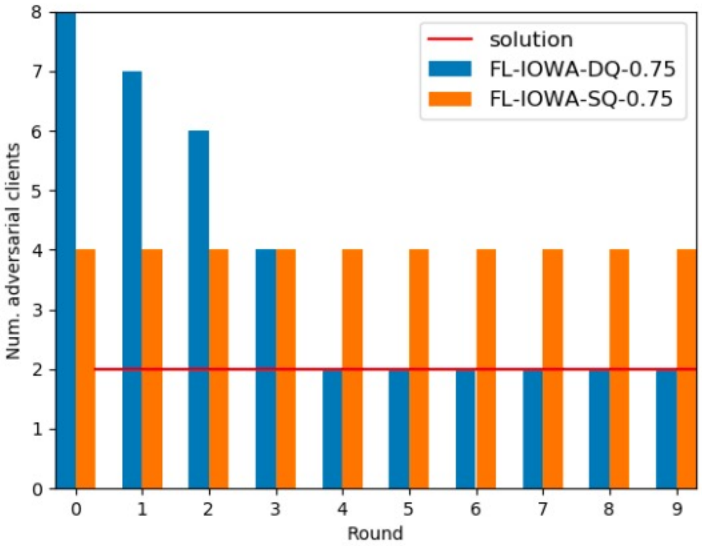

Fig. 5. Adversarial clients detected in AD Scenario with 20 clients

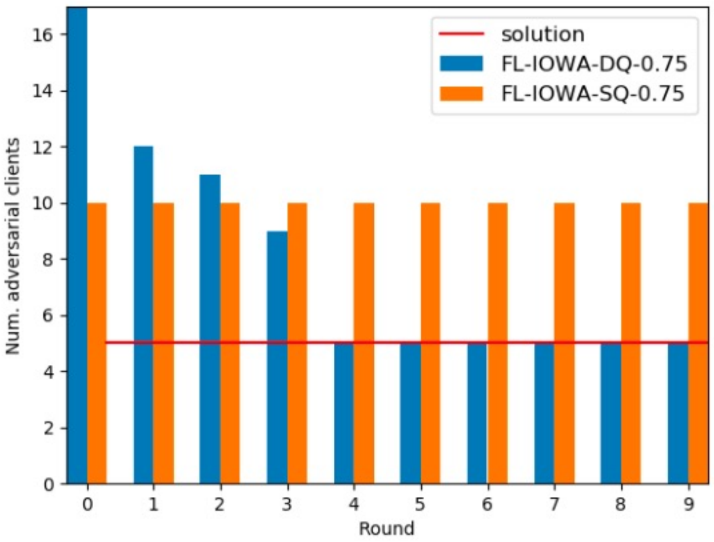

Fig. 6. Adversarial clients detected in AD Scenario with 50 clients

Experiments: Results

2. NON-AD Scenario: without adversarial clients

Table 5. Accuracy of Models in NON-AD Scenario.

	EMNIST		Fashion-MNIST	
	20 clients	**50 clients**	**20 clients**	**50 clients**
FL-FedAvg	0.9864	0.9801	0.8704	0.8452
FL-W-FedAvg	0.9857	0.9769	0.8721	0.8396
FL-AL-80	0.9861	0.9807	0.8772	0.8492
FL-IOWA-SQ-0.4	0.9882	0.9836	0.8793	0.8547
FL-IOWA-SQ-0.75	0.9890	0.9868	0.8726	0.8673
FL-IOWA-DQ-0.4	0.9891	0.9848	**0.8953**	0.8684
FL-IOWA-DQ-0.75	**0.9893**	**0.9873**	0.8923	**0.8728**

Experiments: Results

2. NON-AD Scenario: without adversarial clients

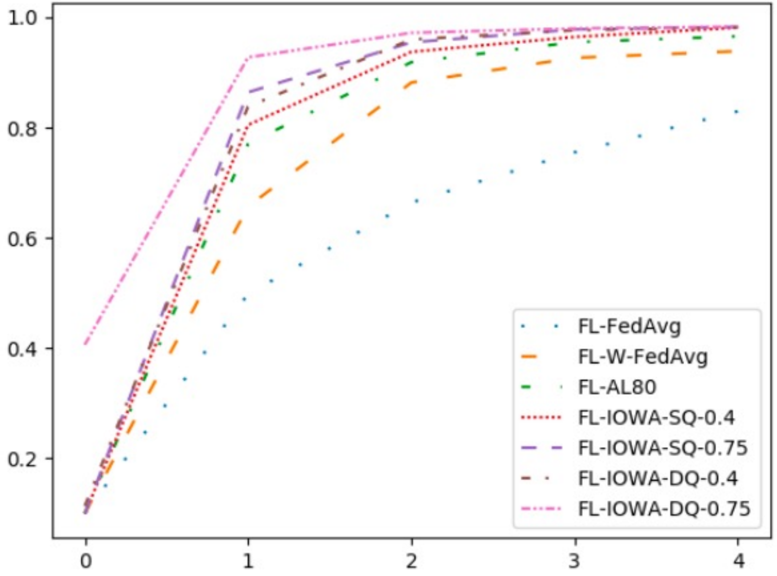

Fig. 7. Accuracy per round of FL models using 20 clients without adversarial clients (NON-AD Scenario) during the first 5 rounds

Experiments: Results

3. High-AD Scenario: 30% of clients are malicious, 6 out of 20 or 15 out of 50.

Table 6. Accuracy of Models in High-AD Scenario.

	EMNIST		Fashion-MNIST	
	20 clients	**50 clients**	**20 clients**	**50 clients**
FL-FedAvg	0.9788	0.9753	0.8451	0.8435
FL-W-FedAvg	0.9769	0.9758	0.8456	0.8228
FL-AL-80	0.9713	0.9781	0.8439	0.8212
FL-IOWA-SQ-0.4	0.9826	0.9820	0.8468	0.8539
FL-IOWA-SQ-0.75	0.9844	0.9861	0.8518	0.8604
FL-IOWA-DQ-0.4	**0.9876**	0.9860	0.8648	0.8610
FL-IOWA-DQ-0.75	0.9873	**0.9874**	**0.8722**	**0.8684**

3 Extra Experiments

Setting

Datasets	Fashion MNIST(fMNIST) • 10 classes • 60k/10k	CIFAR-10 • 10 classes • 50k/10k
Date Distribution	I.I.D • # local epochs = 5	Extreme NON-I.I.D[1] • # local epochs = 1 • #classes per client = 2
Attack schemes	Label Shuffling • 10% corrupted data	Label Mislabeling* • One class
NN	CNN with two convolutional layers	
#max aggregation rounds	40	
#clients	10	
#mal clients	1	

*Label Mislabeling: in mal client, randomly select an original class and replace its labels with target class

[1]. Zhao, Yue, et al. "Federated learning with non-iid data." arXiv preprint arXiv:1806.00582 (2018).

Setting: Extreme Non-i.i.d

```
Data split:
 - Client 0: [ 857 1500  857 1000 1000 1200 3000    0 1000 2000]
 - Client 1: [ 857 1500  857 1000 1000 1200    0    0 1000 2000]
 - Client 2: [ 857    0  857    0    0    0    0    0 1000    0]
 - Client 3: [  0    0 857    0    0    0    0    0    0    0]
 - Client 4: [ 857    0    0 1000    0    0    0    0    0    0]
 - Client 5: [ 857    0    0 1000 1000    0    0    0    0    0]
 - Client 6: [ 857    0  857    0 1000 1200    0    0 1000    0]
 - Client 7: [   0 1500  857 1000 1000 1200    0    0 1000    0]
 - Client 8: [ 858 1500  858 1000 1000 1200    0    0 1000 2000]
 - Client 9: [   0    0    0    0    0    0 3000 6000    0    0]
 - Data size for clients: [12414, 9414, 2714, 857, 1857, 2857, 4914, 6557, 9416, 9000]
```

Fig. 8. Data split of fMNIST in non-i.i.d in size manner.

VS

```
Data split:
 - Client 0: [   1    0    0    0    0    0    0    0 2614 2614]
 - Client 1: [   0    0    0    0    0    0 3165 3165    0    0]
 - Client 2: [2201    1    0    0    0    0    0    0    0 2201]
 - Client 3: [   0    0    0    0 4449 4449    1    0    0    0]
 - Client 4: [   0    0    0 2706 1551 1155    0    0    0    0]
 - Client 5: [   0    0    0    0    0  396 2834 2835 1366    0]
 - Client 6: [ 604    0    0    0    0    0    0    0 1788 1185]
 - Client 7: [   0 2247 2247    1    0    0    0    0    0    0]
 - Client 8: [3194 3752 1904    0    0    0    0    0  232    0]
 - Client 9: [   0    0 1849 3293    0    0    0    0    0    0]
 - Data size for clients: [5229, 6330, 4403, 8899, 5412, 7431, 3577, 4495, 9082, 5142]
```

Fig. 9. Data split of fMNIST in extreme non-i.i.d manner. Each client has no more than four classes.

Extreme Non-i.i.d:

- **Has less variance of the size of the local samples;**
- **Has sparser distribution of classes;**

Results: fMNIST-iid

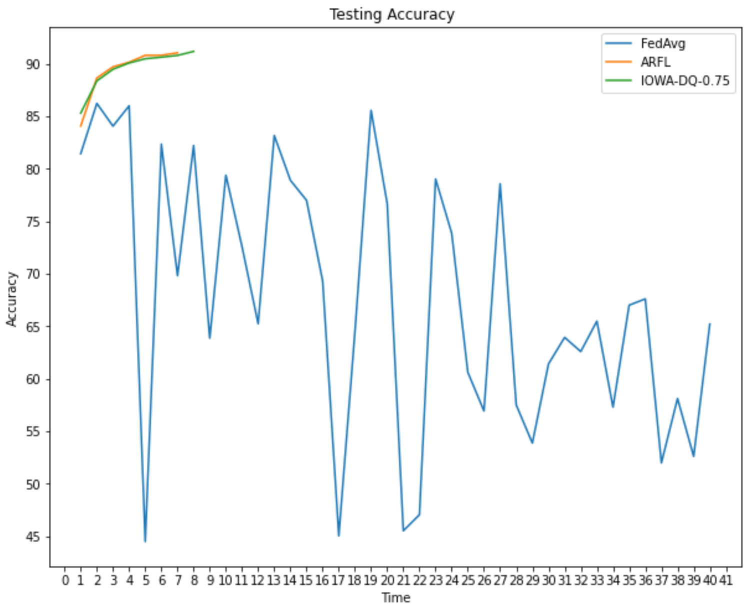

Fig. 10. Testing accuracy of models in shuffling scenario.

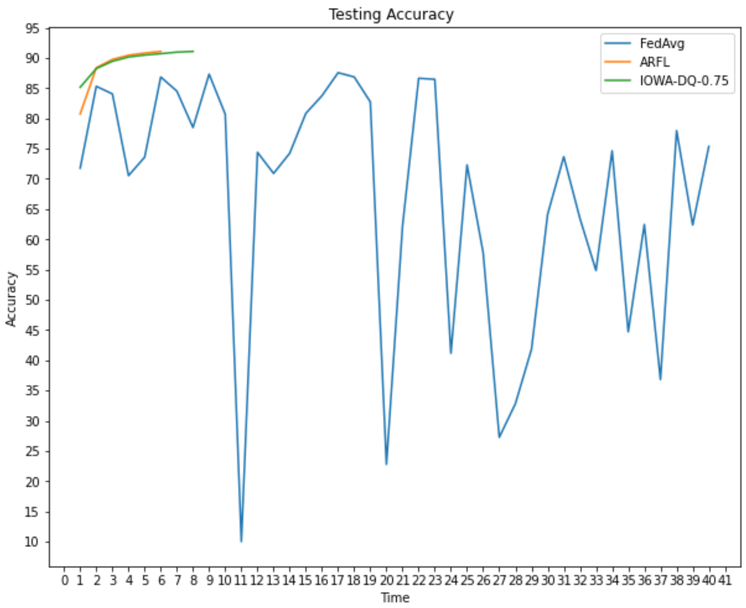

Fig. 11. Testing accuracy of models in mislabeling scenario.

Results: CIFAR-10-iid

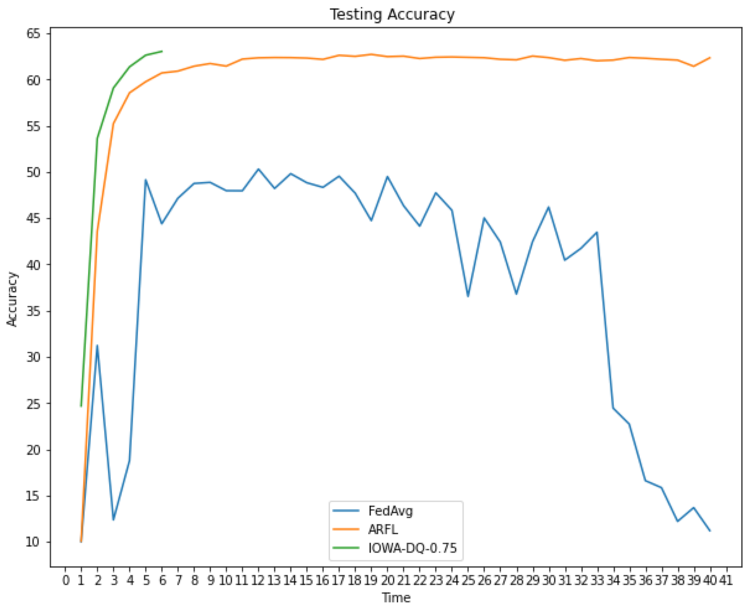

Fig. 12. Testing accuracy of models in shuffling scenario.

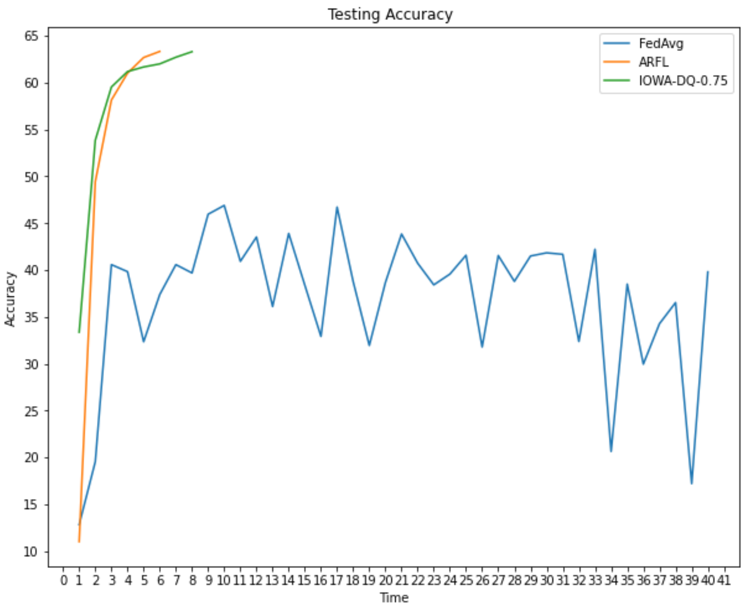

Fig. 13. Testing accuracy of models in mislabeling scenario.

Results: fMNIST-non-iid

Fig. 14. Testing accuracy of models in shuffling scenario.

Fig. 15. Testing accuracy of models in mislabeling scenario.

Results: CIFAR-non-iid

Fig. 16. Testing accuracy of models in shuffling scenario.

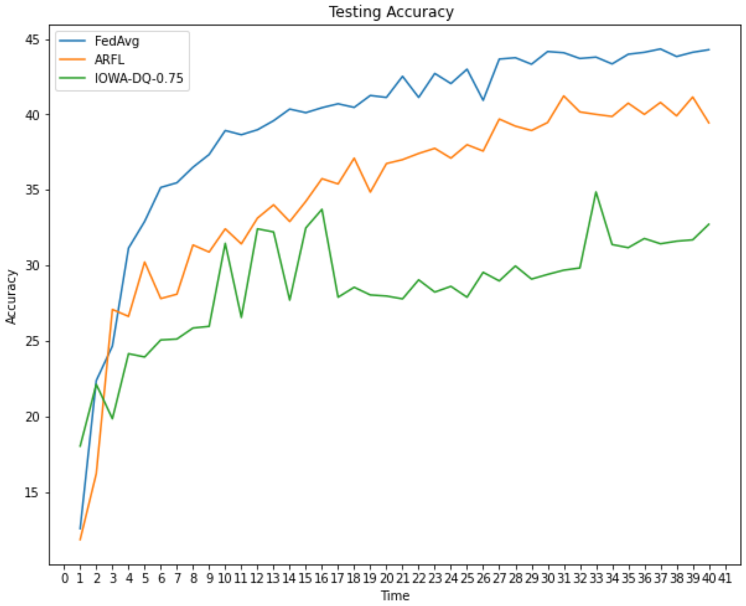

Fig. 17. Testing accuracy of models in mislabeling scenario.

Conclusions

- In iid setting, IOWA-DQ and ARFL are both effective and protect the model from serious performance degeneration. They have very close performance.
- In extreme non-iid setting, defenses by simply adjusting weights are ineffective, even weaker than FedAvg.
- The data distribution, corruption degree highly affect the robustness of defenses.

References

- Li, Shenghui, et al. "Auto-weighted Robust Federated Learning with Corrupted Data Sources." *arXiv preprint arXiv:2101.05880* (2021).
- Rodríguez-Barroso, Nuria, et al. "Dynamic federated learning model for identifying adversarial clients." *arXiv preprint arXiv:2007.15030* (2020).
- Kairouz, Peter, et al. "Advances and open problems in federated learning." arXiv preprint arXiv:1912.04977 (2019).
- Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, 2009.
- Caldas, Sebastian, et al. "Leaf: A benchmark for federated settings." arXiv preprint arXiv:1812.01097 (2018).
- Pillutla, Krishna, Sham M. Kakade, and Zaid Harchaoui. "Robust aggregation for federated learning." arXiv preprint arXiv:1912.13445 (2019).
- Blanchard, Peva, et al. "Machine learning with adversaries: Byzantine tolerant gradient descent." Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017.
- Sattler, Felix, et al. "On the byzantine robustness of clustered federated learning." ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020.
- Zhao, Yue, et al. "Federated learning with non-iid data." arXiv preprint arXiv:1806.00582 (2018).

Thanks