On the Byzantine-Resilience of Distillation-Based Federated Learning

Christophe Roux, Max Zimmer, Sebastian Pokutta

ZIB ZUSE INSTITUTE BERLIN

Berlin Mathematics Research Center MATH+

Algorithm

Federated Learning

for communication round $t = 0$ to $T - 1$ **do**
  SERVER: Broadcast parameters $\bar{w}_i$ to the clients
  CLIENTS: Train on private datasets
  FedAVG
  CLIENTS: Send updated parameters to server
  SERVER: Aggregate parameters to obtain $\bar{w}_i^{t+1}$
  FedDistill
  CLIENTS: Send public dataset predictions to server
  SERVER: Train on public dataset with aggregated client predictions to obtain $\bar{w}_i^{t+1}$
end for
Output: $\bar{w}_T$

- **Federated Averaging (FedAVG):** Clients share model parameters.
- **Federated Distillation (FedDistill):** Clients share *predictions* on a public, unlabeled dataset. Server distills knowledge using these predictions.

FedAVG vs. FedDistill Attack Vectors

FedAVG: (A single attacker can arbitrarily shift $\bar{w}$!)

$$\bar{w} \leftarrow \frac{1}{N} \sum_{i \in \mathcal{H} \cup \mathcal{B}} w_i = \underbrace{\frac{1}{N} \sum_{i \in \mathcal{B}} w_i}_{\text{Attack vector}} + \frac{1}{N} \sum_{i \in \mathcal{H}} w_i$$

FedDistill: Indirect influence via distillation targets.

Honest distillation: $\min_w \sum_{x \in D_p} \mathcal{L}(h(x, w), \bar{Y}_{\mathcal{H}}(x))$ $(\mathcal{P}_{\text{honest}})$

Actual distillation: $\min_w \sum_{x \in D_p} \mathcal{L}(h(x, w), \underbrace{\bar{Y}(x)}_{\text{Attack vector}})$ $(\mathcal{P}_{\text{distill}})$

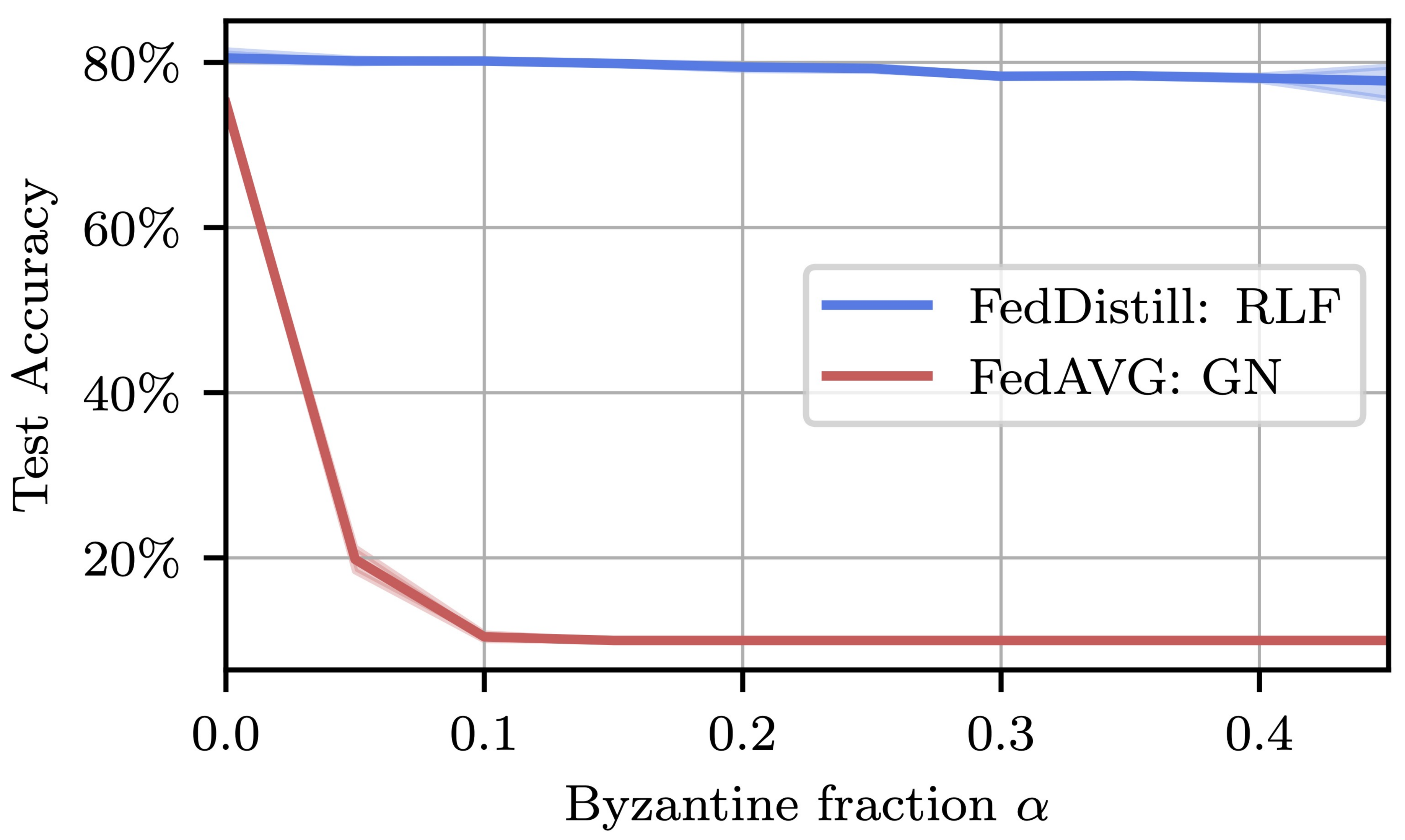
where $\bar{Y}(x) = \frac{1}{N} \sum_{i \in \mathcal{H} \cup \mathcal{B}} Y_i(x)$ and $\bar{Y}_{\mathcal{H}}(x) = \frac{1}{N} \sum_{i \in \mathcal{H}} Y_i(x)$
$\Rightarrow$ Indirect influence, predictions $Y_i(x)$ lie in (bounded) probability simplex.

Robustness of FedDistill

Theorem: (Informal) *If $\tilde{w}$ is a stationary point of $(\mathcal{P}_{\text{distill}})$, then it is also an $\mathcal{O}(C^2\alpha^2)$-approximate stationary point of $(\mathcal{P}_{\text{honest}})$, where $C > 0$ is a constant independent of the client predictions. Further, in expectation, running SGD on $(\mathcal{P}_{\text{distill}})$ to achieve an ε-approximate stationary point yields an $\mathcal{O}(\varepsilon + C^2\alpha^2)$-approximate stationary point of $(\mathcal{P}_{\text{honest}})$.*

Intuition: $Y \mapsto \nabla_w \mathcal{L}(h(x, w), \cdot)$ is Lipschitz for typical loss functions.

Motivation: FedDistill is more byzantine-resilient than FedAVG



ResNet-18 on CINIC-10: Final test accuracy of FedAVG and FedDistill, varying the fraction of byzantine clients for two naive attacks. For FedAVG, the byzantine clients simply send Gaussian noise (GN) instead of parameter updates. For FedDistill, they send random one-hot predictions, we refer to this as the Random Label Flip (RLF) attack.

Results

	CINIC-10 (ResNet-18), BA=80.2±0.1				Clothing1M (ResNet-50), BA=69.0±0.3			
	Mean	GM	Cronus	ExpGuard	Mean	GM	Cronus	ExpGuard
RLF	76.9±0.4	79.3±0.3	76.6±0.0	78.8±0.1	84.6±0.1	85.4±0.6	84.7±0.3	85.4±0.0
LMA	54.6±1.2	75.0±0.6	71.1±1.7	77.4±1.1	73.4±8.6	83.3±0.2	80.6±2.3	85.4±0.1
CPA	45.9±0.4	71.2±5.2	65.9±3.3	79.2±0.2	68.4±0.8	78.4±0.9	74.5±0.6	85.5±0.8
HIPS+LMA	75.3±0.1	68.7±0.1	67.7±1.0	73.3±0.9	84.8±0.1	78.0±1.6	78.5±1.1	83.8±0.2
HIPS+CPA	74.2±1.1	65.8±0.5	66.4±0.1	72.9±0.7	85.0±0.1	79.4±0.8	77.3±0.1	83.2±0.9

	CIFAR-100 (WideResNet-28), BA=66.8±0.5				CIFAR-10 (ResNet-18), BA=87.7±1.2			
	Mean	GM	Cronus	ExpGuard	Mean	GM	Cronus	ExpGuard
RLF	65.2±0.7	65.2±0.3	44.3±1.7	63.9±0.6	69.4±1.2	68.7±0.8	68.6±0.4	68.7±1.1
LMA	41.8±4.4	51.3±0.1	44.6±0.2	57.2±1.2	40.3±3.3	58.3±0.7	61.4±0.5	68.3±0.7
CPA	43.3±1.2	56.7±0.9	55.3±0.3	62.1±1.4	33.7±2.7	58.4±0.3	43.9±12.9	68.5±1.0
HIPS+LMA	50.3±3.3	34.3±0.4	34.4±2.8	49.3±0.5	33.7±2.7	58.4±0.3	43.9±12.9	68.5±1.0
HIPS+CPA	47.2±4.2	32.6±4.5	28.1±0.5	46.4±0.0	63.4±1.12	55.2±1.1	54.8±2.1	57.7±0.5

FedDistill: 20 clients of which nine are byzantine (α=0.45). Final test accuracy averaged over multiple runs with standard deviation for different attacks and defences. BA refers to the baseline accuracy, i.e., the final accuracy of FedDistill if all clients are honest.

New Attacks

Loss Maximization Attack (LMA): Byzantine clients choose predictions $Y_{\mathcal{B}}(x)$ to maximize the server's distillation loss $\mathcal{L}(h(x, w), \bar{Y}(x))$ given the honest mean $\bar{Y}_{\mathcal{H}}(x)$. This means predicting the class with the minimum probability under $\bar{Y}_{\mathcal{H}}(x)$.

Class Prior Attack (CPA): Exploits semantic similarity. Uses a class similarity matrix C. Predicts the class least similar (via C) to the most likely class under $\bar{Y}_{\mathcal{H}}(x)$.

Attack Obfuscation: HIPS

- **Problem:** Aggressive attacks (LMA/CPA) generate easily detectable predictions (e.g., one-hot vectors).
- **HIPS Idea:** Make attacks stealthier by constraining Byzantine predictions $Y_{\mathcal{B}}$ to lie within the convex hull of honest predictions $\{Y_i\}_{i \in \mathcal{H}}$.
- **Tradeoff:** Increased stealth vs. potentially reduced attack impact.

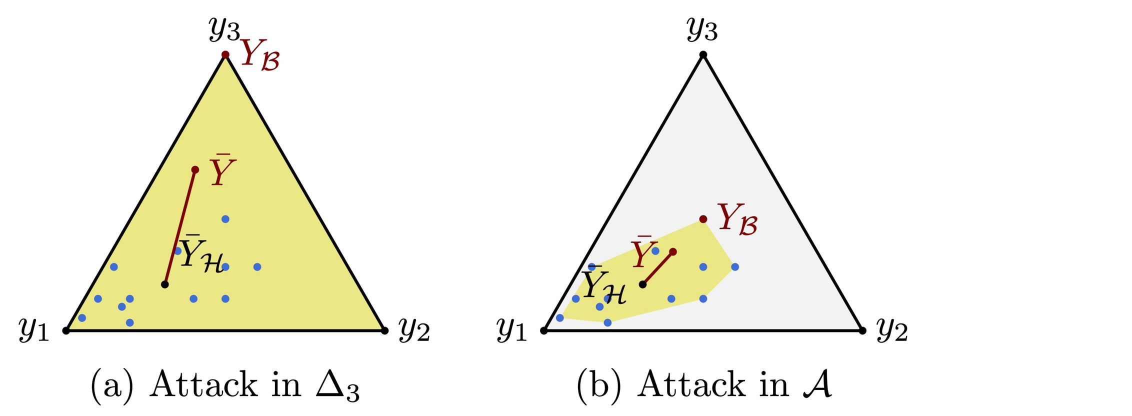


Illustration in Δ_3. HIPS restricts Byzantine prediction (yellow area) based on honest predictions (blue dots).

New Defence: ExpGuard

ExpGuard

Input: Pred. $Y_i^{t+1}(\mathcal{D}_p)$, weights p_i, $\forall i \in N$, aggregation method AGG.
$\sigma_i \leftarrow \text{AGG}(Y_i^{t+1}(\mathcal{D}_p)), \forall i \in [n]$ ▷ Compute outlier scores
$p_i^{t+1} \leftarrow p_i \exp(-\sigma_i), \forall i \in [n]$ ▷ Update weights
$\bar{Y}_i^{t+1}(x) \leftarrow \frac{1}{\sum_{j=1}^{n} p_j^{t+1}} \sum_{i=1}^{N} p_i^{t+1} Y_i^{t+1}(x)$ ▷ Comp. weighted sum $\forall x \in \mathcal{D}_p$
Output: $\bar{Y}_i^{t+1}(x)$, p_i^{t+1}, $\forall i \in [N]$

ExpGuard:

- Enhances robust aggregators by incorporating historical information.
- Tracks each client's deviation from the robust aggregate over time.
- Assigns weights p_i to clients, reducing weight for larger deviations.
- Uses weighted average for aggregation.
- Significantly improves resilience across various base aggregators, often approaching performance of the non-attacked setting.