ACE: A Model Poisoning Attack on Contribution Evaluation Method in Federated Learning

USENIX Security 2024

Zhangchen Xu[1], Fengqing Jiang[1], Luyao Niu[1], Jinyuan Jia[2], Bo Li[3] and Radha Poovendran[1]

[1] University of Washington
[2] The Pennsylvania State University
[3] University of Chicago

33RD USENIX SECURITY SYMPOSIUM

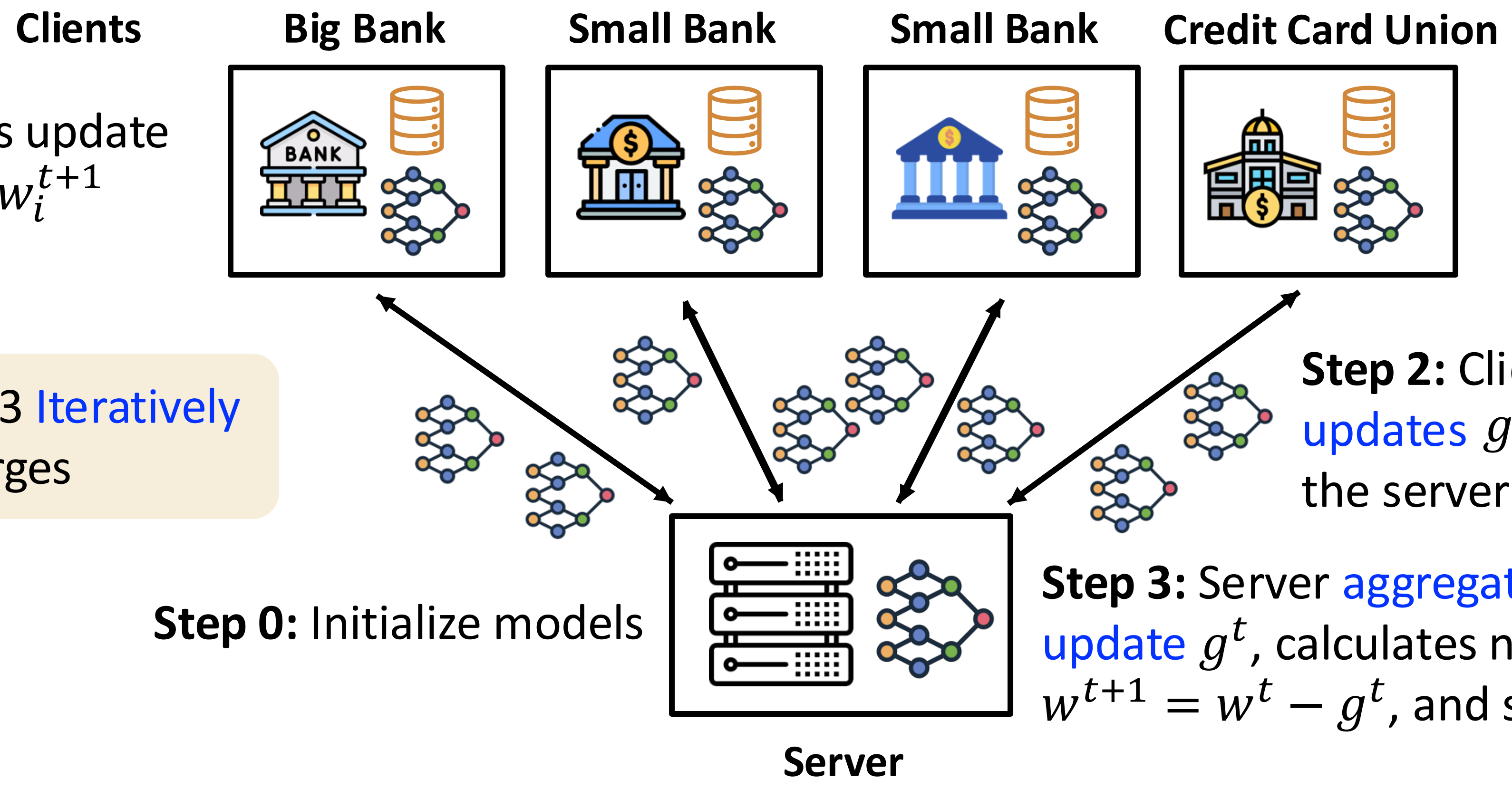
Outline

- Introduction: Federated Learning and Contribution Evaluation in FL
- Threat Model
- Design of a Model Poisoning Attack to Contribution Evaluation, **ACE**
- Evaluation of **ACE**
- Conclusion and Future Work

Introduction: Federated Learning

Federated Learning (FL) [1,2]**:** Collaboratively train a machine learning (ML) model without sharing local training data

Step 1: Clients update local models w_i^{t+1}



Step 2: Clients send local model updates $g_i^t = w^t - w_i^{t+1}$ back to the server

Run Step 1-3 Iteratively until converges

Step 0: Initialize models

Step 3: Server aggregates a global model update g^t, calculates new global model $w^{t+1} = w^t - g^t$, and sends back to clients

Introduction: Contribution Evaluation in FL

Factors that affect FL success:
Data quality (e.g., size, distribution), and participation willingness of clients

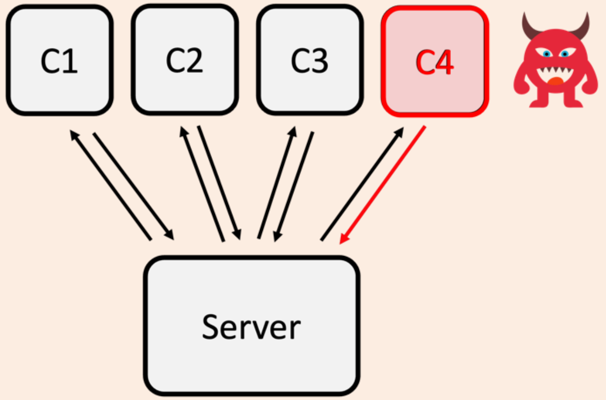
- Current methods [3-14] assume honest participants
- Contribution cannot be measured by data quality (server doesn't have raw data)
- This unique feature may be leveraged by malicious clients by sending carefully manipulated local model updates

Research Question: Can a malicious client processing low-quality data elevate its contribution evaluated by the server? And How?

Threat Model

Attacker's capabilities and knowledge:

- Has access to the local training dataset
- Has access to the global model
- Controls the training processes
- Manipulates its local model updates before sending them to the server (Model Poisoning)



Attacker's objective:

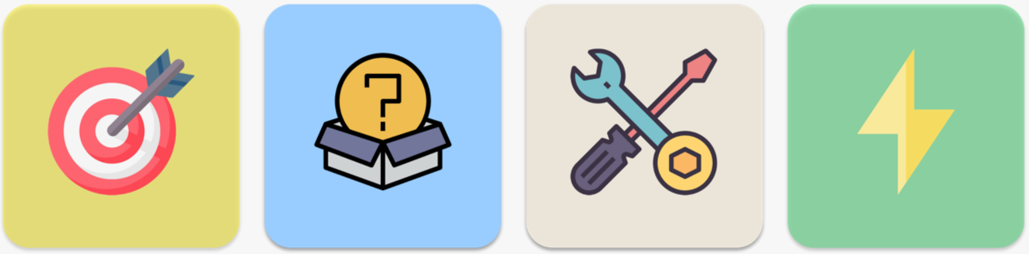
Elevate the attacker's contribution

Contribution Evaluation Method

$$\max_{g_i} E(g_i)$$

Local Model Update of Client i

Design Goals:

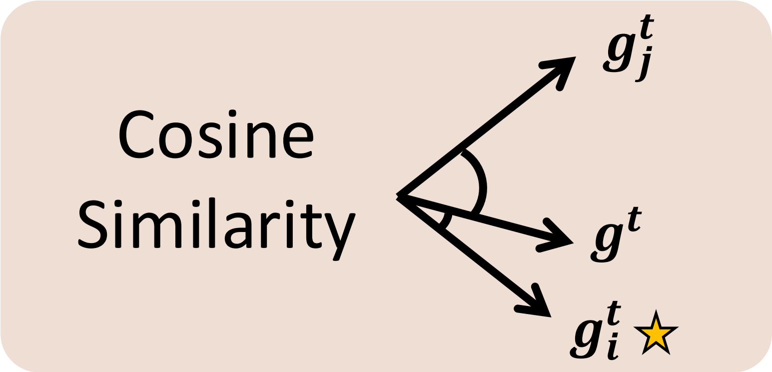


Effective | Universal | Performance Preserving | Efficient

Design of ACE

Current Contribution Evaluation Methods

1. *Individual Evaluation*
    - Cosine similarity between local and global model updates [3-7]
    - Loss / Accuracy in a server validation dataset [8-9]

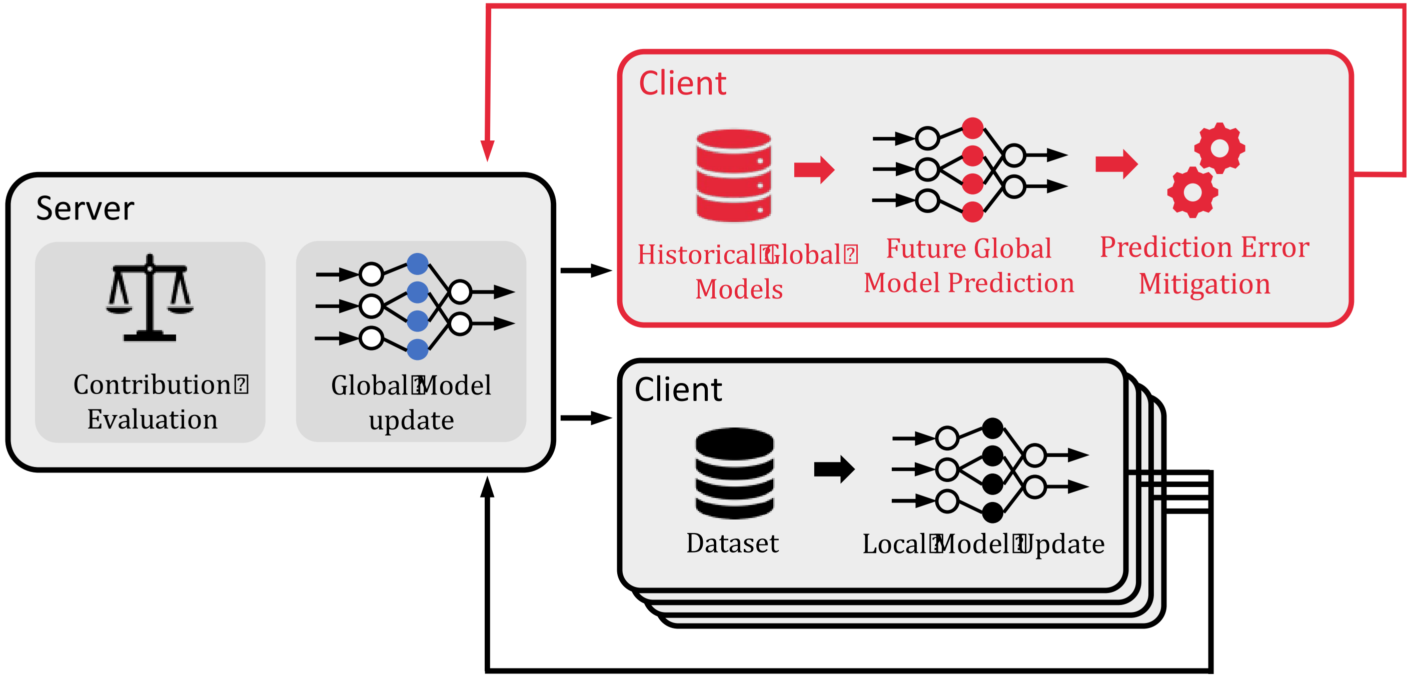


2. *Joint Evaluation*
    - Marginal loss (Leave-One-Out) [10-11]
    - Shapley Value (SV) [12-14]

Design of ACE

Key Insight of ACE: Iterative nature of FL leaks information about other clients

→ Mimic global model updates using historical information of past global models



Design of ACE

Step 1: Future Global Model Prediction

Using **Cauchy Mean value theorem**[15] and **L-BFGS Algorithm**[16] to estimate global model update $\hat{g}^t$:

$$\begin{aligned}\hat{g}^t &= g^{t-1} + H(t)(w^t - w^{t-1}) \\ &\approx g^{t-1} + \mathrm{LBFGS}(w^t - w^{t-1},\ \boldsymbol{\Delta W^t}, \boldsymbol{\Delta G^t}),\end{aligned}$$

$\boldsymbol{\Delta W^t}, \boldsymbol{\Delta G^t}$: Buffered historical information

$$\Delta w^t = w^t - w^{t-1} \xrightarrow{\text{Buffer}} \Delta W^t = [\Delta w^{t-m}, \Delta w^{t-m+1}, \ldots, \Delta w^{t-1}]$$

$$\Delta g^t = g^t - g^{t-1} \qquad \Delta G^t = [\Delta g^{t-m}, \Delta g^{t-m+1}, \ldots, \Delta g^{t-1}]$$

Design of ACE

Step 2: Prediction Error Mitigation

Threshold-based Filtering - Global model updates should have a similar scale

$$\hat{g}^t \approx g^{t-1} + \mathrm{LBFGS}(\Delta \boldsymbol{W}^t, \Delta \boldsymbol{G}^t, w^t - w^{t-1}),$$

If the l-2 norm of the L-BFGS is less than a threshold:

$$\|\mathrm{LBFGS}(\Delta \mathrm{W}^t, \Delta G^t, v)\| \leq \tau$$

The prediction error is tolerable.

(Step 3) Strategies to enhance ACE based on different measurements

Evaluation of ACE: Setup

Datasets: MNIST, CIFAR10, and Tiny-ImageNet

Models: CNN and VGG11

Data Partition:

- Uniform Distribution (UNI)
- Power Law Distribution (POW)
- Class Imbalance (CLA)

Attacker: Client with the lowest contribution

Contribution Evaluation Methods:

$\underbrace{\text{Federated-SV (FedSV) [16], Leave-One-Out (LOO) [12]}}_{\textit{Joint Evaluation}}$, $\underbrace{\text{CFFL [11], GDR [8], and RFFL [7]}}_{\textit{Individual Evaluation}}$

Baseline Attacks:

- Delta Weight Attack [17] $g_i^t = w^{t-1} - w^t + \delta$
- Scaling Attack [18]
- Data Augmentation

Evaluation of ACE

Evaluation Metrics

Effective

Metrics:

- Normalized Contribution Score

Sum of contributions of for all rounds

$$CS_i = \frac{\sum_{t=1}^{T} e_i^t}{\sum_j \sum_{t=1}^{T} e_j^t}$$

- Rank Gain

$$\Delta R_i = \widehat{R}_i - R_i$$

Universal

Diverse contribution evaluation methods

Performance Preserving

Metric:
Test Accuracy

Efficient

Metric: $\frac{t(train)}{t(ACE)}$

Evaluation of ACE

Takeaway 1: ACE is **Effective** and **Universal**

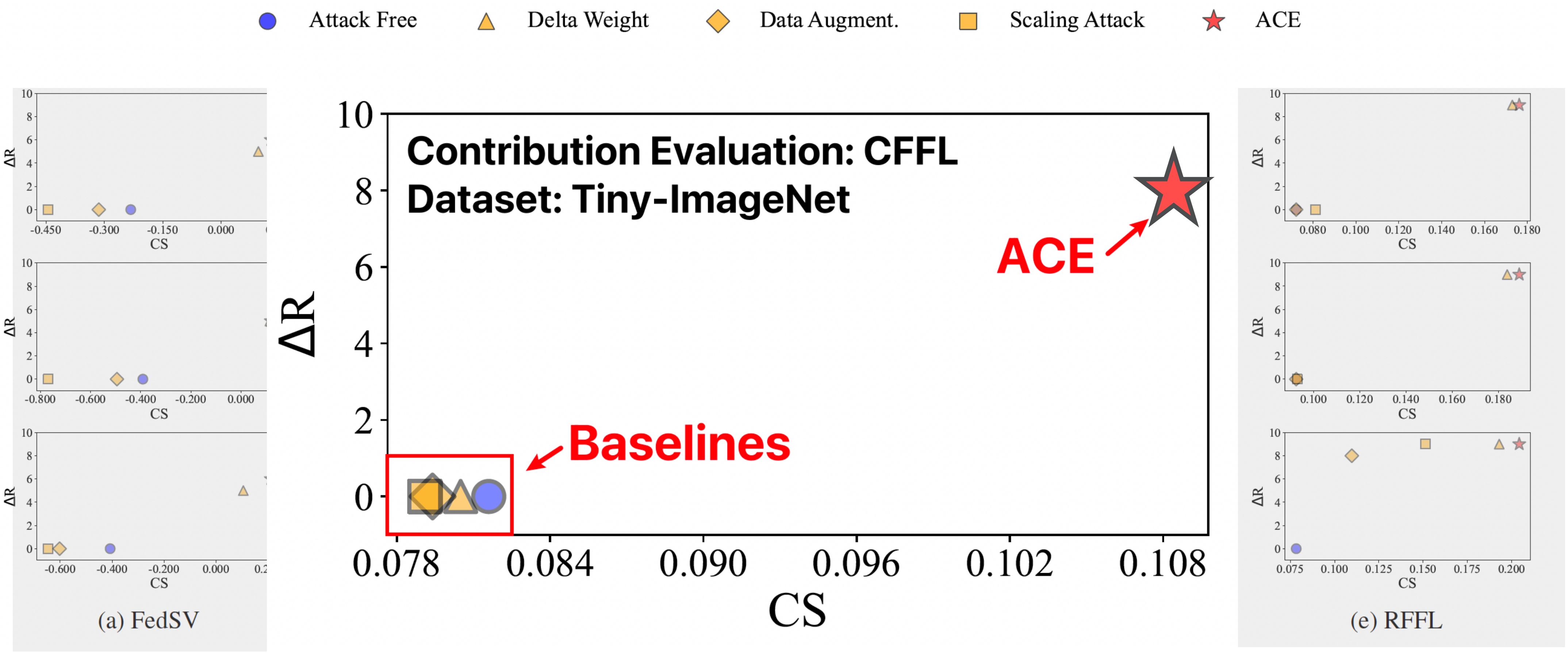
Metrics:
- Rank Gain
- Contribution Score

Dataset

MNIST

CIFAR-10

Tiny-Imagenet



CLA (heterogeneous) Data Distribution

Evaluation of ACE

Takeaway 2: ACE preserves **Utility**

Metric:
Test Accuracy

Contribute Evaluation	Attack	MNIST UNI	MNIST POW	MNIST CLA	CIFAR-10 UNI	CIFAR-10 POW	CIFAR-10 CLA	Tiny-ImageNet UNI	Tiny-ImageNet POW	Tiny-ImageNet CLA
	Attack Free	95.86%	95.69%	89.89%	71.16%	70.82%	56.32%	46.37%	47.84%	44.98%
					70.89%	71.02%	57.16%	46.10%	47.80%	45.27%
					71.63%	70.27%	56.05%	46.77%	48.37%	45.26%
					71.58%	71.01%	55.29%	46.59%	48.07%	45.01%
					71.30%	71.45%	57.60%	46.35%	48.23%	45.94%
					71.16%	70.82%	56.32%	46.37%	47.84%	44.98%
					70.89%	71.02%	57.16%	46.10%	47.80%	45.27%
					71.63%	70.27%	56.05%	46.77%	48.37%	45.26%
					71.58%	71.01%	55.29%	46.59%	48.07%	45.01%
					71.30%	71.45%	57.60%	46.35%	48.23%	45.94%
					71.84%	60.65%	49.99%	51.77%	48.23%	39.96%
					70.66%	59.37%	50.62%	51.30%	44.18%	40.54%
					73.08%	60.93%	50.62%	51.92%	47.83%	40.04%
					71.55%	60.41%	49.91%	52.22%	44.23%	39.87%
	ACE	96.61%	95.35%	83.18%	70.44%	62.03%	52.45%	51.53%	49.20%	42.02%
GDR	Attack Free	96.26%	96.23%	85.41%	70.97%	71.33%	56.66%	51.80%	51.96%	44.78%
	Delta Weight	96.84%	96.43%	89.02%	70.32%	70.76%	59.18%	52.19%	52.57%	46.01%
	Data Augment.	96.43%	96.18%	87.42%	72.01%	71.12%	57.38%	51.79%	52.04%	44.84%
	Scaling Attack	96.26%	96.23%	85.42%	71.01%	71.36%	56.63%	51.84%	51.89%	44.78%
	ACE	96.78%	96.53%	89.12%	70.27%	70.60%	59.23%	52.64%	52.77%	46.61%
RFFL	Attack Free	96.78%	96.85%	92.67%	71.78%	71.03%	57.66%	52.35%	52.43%	46.72%
	Delta Weight	96.66%	96.85%	91.83%	70.69%	71.07%	56.95%	51.89%	52.49%	46.84%
	Data Augment.	96.25%	96.08%	92.67%	71.84%	71.04%	57.60%	51.83%	52.50%	46.31%
	Scaling Attack	95.96%	95.97%	91.73%	71.73%	71.07%	56.60%	50.84%	52.50%	46.17%
	ACE	96.64%	96.87%	92.30%	70.72%	70.90%	57.36%	51.75%	52.31%	46.54%

No Attack	71.16%	70.82%	56.32%
	70.89%	71.02%	57.16%
	71.63%	70.27%	56.05%
	71.58%	71.01%	55.29%
ACE	71.30%	71.45%	57.60%

Evaluation of ACE

Takeaway 3: ACE is **Efficient**

Metric: The ratio between the computation costs of using a local training dataset to learn a local model update and ACE.

Dataset	FedSV	LOO	CFFL	GDR	RFFL
MNIST	30.88×	30.88×	7.48×	16.15×	18.26×
CIFAR-10	270.81×	270.81×	21.25×	86.48×	101.44×
Tiny-ImageNet	35.35×	35.35×	13.26×	29.22×	24.79×

Evaluation: Countermeasures to ACE

ACE is stealthy against state-of-the-art defenses [19-23]

Conclusion and Future Work

- Current contribution evaluation methods in FL can be attacked by malicious clients
- We propose **ACE**, a model poisoning attack to contribution evaluation in FL, which successfully elevates malicious clients' contributions
- **ACE** is effective, preserves utility, efficient, and universal
- Current countermeasures fail to defend against **ACE**

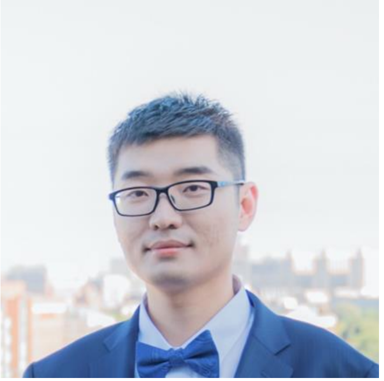
- New mitigation strategies need to be developed

Acknowledgements

This is a collaborative work!

Co-authors:

Fengqing Jiang
(UW)



Prof. Luyao Niu
(UW)

Prof. Jinyuan Jia
(PSU)

Prof. Bo Li
(UChicago)

Prof. Radha Poovendran
(UW)

Acknowledgements

This work is supported by:

References

[1] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics (pp. 1273-1282). PMLR.

[2] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated Learning: Strategies for Improving Communication Efficiency. arXiv preprint arXiv:1610.05492, 2016.

[3] Meirui Jiang, Holger R Roth, Wenqi Li, Dong Yang, Can Zhao, Vishwesh Nath, Daguang Xu, Qi Dou, and Ziyue Xu. Fair federated medical image segmentation via client contribution estimation. In CVPR, pages 16302–16311, 2023.

[4] Zhuan Shi, Lan Zhang, Zhenyu Yao, Lingjuan Lyu, Cen Chen, Li Wang, Junhao Wang, and Xiang-Yang Li. Fedfaim: A model performance-based fair incentive mechanism for federated learning. IEEE Trans. Big Data, 2022.

[5] Xinyi Xu and Lingjuan Lyu. A reputation mechanism is all you need: Collaborative fairness and adversarial robustness in federated learning. arXiv preprint arXiv:2011.10464, 2020.

[6] Xinyi Xu, Lingjuan Lyu, Xingjun Ma, Chenglin Miao, Chuan Sheng Foo, and Bryan Kian Hsiang Low. Gradient driven rewards to guarantee fairness in collaborative machine learning. NeurIPS, 34:16104–16117, 2021.

[7] Jingwen Zhang, Yuezhou Wu, and Rong Pan. Incentive mechanism for horizontal federated learning based on reputation and reverse auction. In WWW, pages 947–956, 2021.

[8] Yiqiang Chen, Xiaodong Yang, Xin Qin, Han Yu, Piu Chan, and Zhiqi Shen. Dealing with label quality disparity in federated learning. Federated Learning: Privacy and Incentive, pages 108–121, 2020.

References

[9] Lingjuan Lyu, Xinyi Xu, Qian Wang, and Han Yu. Collaborative fairness in federated learning. Federated Learning: Privacy and Incentive, pages 189–204, 2020.

[10] Guan Wang, Charlie Xiaoqian Dang, and Ziye Zhou. Measure contribution of participants in federated learning. In IEEE BigData, pages 2597–2604. IEEE, 2019.

[11] Zhebin Zhang, Dajie Dong, Yuhang Ma, Yilong Ying, Dawei Jiang, Ke Chen, Lidan Shou, and Gang Chen. Refiner: A reliable incentive-driven federated learning system powered by blockchain. VLDB Endowment,14(12):2659–2662, 2021.

[12] Amirata Ghorbani and James Zou. Data Shapley: Equitable valuation of data for machine learning. In ICML, pages 2242–2251. PMLR, 2019.

[13] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. Towards efficient data valuation based on the shapley value. In AISTATS, pages 1167–1176. PMLR, 2019.

[14] Tianhao Wang, Johannes Rausch, Ce Zhang, Ruoxi Jia, and Dawn Song. A principled approach to data valuation for federated learning. Federated Learning: Privacy and Incentive, pages 153–167, 2020.

[15] Serge Lang. Real and functional analysis, volume 142. Springer Science & Business Media, 2012.

[16] Richard H Byrd, Jorge Nocedal, and Robert B Schnabel. Representations of quasi-newton matrices and their use in limited memory methods. Mathematical Programming, 63(1-3):129–156, 1994

References

[17] Jierui Lin, Min Du, and Jian Liu. Free-riders in federated learning: Attacks and defenses. arXiv preprint arXiv:1911.12560, 2019.
[18] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In AISTATS, pages 2938–2948. PMLR, 2020.
[19] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. NeurIPS, 30, 2017.
[20] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In ICML, pages 5650–5659. PMLR, 2018.
[21] Qi Xia, Zeyi Tao, Zijiang Hao, and Qun Li. FABA: an algorithm for fast aggregation against byzantine attacks in distributed neural networks. In IJCAI, 2019.
[22] Di Cao, Shan Chang, Zhijian Lin, Guohua Liu, and Donghong Sun. Understanding distributed poisoning attack in federated learning. In ICPADS, pages 233–239. IEEE, 2019.
[23] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. Mitigating Sybils in federated learning poisoning. arXiv preprint arXiv:1808.04866, 2018.

Thank You

zxu9@uw.edu