Identify Backdoored Model in Federated Learning via Individual Unlearning Supplementary Material

Jiahao Xu Zikai Zhang Rui Hu
University of Nevada, Reno
{jiahaox,zikaiz,ruihu}@unr.edu

A. More details of attack settings

We add a "plus" trigger to benign samples to generate the poisoned data samples. For DBA attack [6], we decompose the "plus" trigger into four local patterns, and each malicious client only uses one of these local patterns. For Scaling attack [1], we use a scale factor of 2.0 to scale up all malicious model updates. For PGD attack [5], malicious local models are projected onto a sphere with a radius equal to the L_2-norm of the global model in the current round for CIFAR-100, while for CIFAR-100 we make the radius of the sphere 10 times smaller than the norm. For Neurotoxin [7], malicious model updates are projected to the dimensions that have Bottom-75% importance in the aggregated model update from the previous round. For Lie attack [2], we set the maximal value $z = 1.5$.

B. More details of defense model

In our setting, the server does not have access to the clients' local datasets but is familiar with the training objective, allowing the server to collect a proxy dataset independently which is correlated to the local data distribution. Additionally, the server lacks specific information about the backdoor attacks, such as the type of trigger used. We further assume that the server has no prior knowledge of the number of malicious clients. To defend against backdoor attacks, the server will apply an AGR to handle local model updates received from clients and generate an aggregated model update at each training round.

C. More details of training settings

We use stochastic gradient descent (SGD) as the local solver, with the learning rates set as 0.1 with the decay ratio 0.99 and the number of local training epochs set as 2. Note that in our setting, malicious clients share the same settings as benign ones. The number of training rounds is set to $T = 100$ for CIFAR-10 [3] and $T = 150$ for CIFAR-100 [3].

Table 1. The MA, BA, and RA comparison across different proxy dataset sizes.

Distribution	Metric	Proxy dataset size $\|D_p\|$					
		500	250	200	125	100	50
IID	MA↑	90.86	90.83	90.68	91.28	91.03	90.87
	BA↓	0.87	0.50	0.58	0.84	0.72	1.19
	RA↑	88.91	88.74	88.67	88.36	88.68	88.14
Non-IID	MA↑	88.44	88.05	88.41	86.46	87.73	88.35
	BA↓	0.77	1.83	1.72	6.78	60.37	99.99
	RA↑	85.21	84.02	84.21	77.60	35.58	0.01

D. Experiments on various proxy data sizes

We further examine how the size of the proxy dataset affects MASA's performance. Specifically, we vary the number of images in the proxy dataset from the default 500 (1% of the training dataset) down to an extreme of 50 images. The MA, BA, and RA on both IID and non-IID CIFAR-10 datasets are presented in Table 1. For the IID case, MASA's performance remains relatively stable regardless of the proxy dataset size. Even when the proxy dataset is reduced to 50 images, MASA experiences only a slight drop in BA and RA. However, in non-IID scenarios, MASA shows greater sensitivity to proxy dataset size. MASA remains robust until the dataset size drops to 125 images, after which its ability to defend against backdoor attacks weakens significantly. Based on these results, MASA should be implemented with a reasonably sized proxy dataset in practice. Our experiments show that using a proxy dataset with a size of just 1% of the training dataset is sufficient, which not only reduces the time and effort required for data collection but also minimizes storage needs and computational overhead. This makes MASA more practical and scalable in real-world applications where resources are limited.

E. Experiments on generated proxy datasets

In our default setting, we sample 1% of training data to construct the proxy dataset. Here, we assess MASA's performance with a proxy dataset generated by cutting-edge

Table 2. Performance of MASA* and MASA on IID and non-IID CIFAR-10 datasets.

Distribution	Method	Badnet			Scaling		
		MA↑	BA↓	RA↑	MA↑	BA↓	RA↑
IID	MASA*	90.88	0.71	88.96	90.88	0.71	88.96
	MASA	90.86	0.87	88.91	90.86	0.87	88.91
Non-IID	MASA*	88.52	1.47	84.31	88.48	0.80	85.57
	MASA	88.44	0.77	85.21	88.60	0.96	85.34

pre-trained generative models. Specifically, we utilize the checkpoint from the SOTA StyleGAN-XL [4][1] to generate 50 images per class of CIFAR-10 dataset, forming the proxy dataset. We refer to MASA using this generated dataset as MASA*. The MA, BA, and RA under both Badnet and Scaling attacks on IID and non-IID CIFAR-10 datasets are summarized in Table 2. Overall, MASA* demonstrates performance consistent with MASA across both IID and non-IID scenarios. These results suggest that MASA remains effective when applied to a generated proxy dataset, significantly improving its practical utility in situations where collecting a proxy dataset is challenging or infeasible.

F. Discussion and future works

In this section, we discuss the primary limitation of MASA: the individual unlearning performed on the server adds an extra computational load. This limitation can impact MASA's effectiveness, especially in larger-scale FL deployments. One potential solution to mitigate this limitation is to utilize a more powerful server capable of parallel unlearning. This approach would reduce the computational cost of individual unlearning by a factor of $1/n$ compared to the current MASA implementation.

Another limitation of MASA is its reliance on a clean proxy dataset that overlaps with the main task data, which may conflict with the privacy-preserving goals of FL in sensitive scenarios. To address this, one possible solution is to shift the unlearning process to local execution on clients. This approach would require protection to ensure that malicious clients follow the unlearning protocol. Alternatively, a verification mechanism could be introduced to detect if the models returned by clients genuinely reflect the unlearning process, thereby maintaining robustness against malicious behavior.

References

[1] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International conference on artificial intelligence and statistics*, pages 2938–2948. PMLR, 2020. 1

[2] Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[3] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. *University of Toronto*, 2009. 1

[4] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022. 2

[5] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. *Advances in Neural Information Processing Systems*, 33:16070–16084, 2020. 1

[6] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *International conference on learning representations*, 2019. 1

[7] Zhengming Zhang, Ashwinee Panda, Linyue Song, Yaoqing Yang, Michael Mahoney, Prateek Mittal, Ramchandran Kannan, and Joseph Gonzalez. Neurotoxin: Durable backdoors in federated learning. In *International Conference on Machine Learning*, pages 26429–26446. PMLR, 2022. 1

[1] https://github.com/autonomousvision/stylegan-xl