FLTracer: Accurate Poisoning Attack Provenance in Federated Learning

APPENDIX A
CAM EXTRACTION DETAILS

Fig. 13 depicts the process of CAM extraction. Here three clients are present, each having a model consisting of two convolutional layers. The first and second layers convolutional layers have sizes of $3\times3\times3\times3$ and $4\times4\times3\times3$, respectively. The convolution kernel's size is $L_1 \times L_2 = 3\times3$.

Step1. We collect the first convolution kernel of each client and compute the anomaly score (i.e., a_1, a_2, and a_3) using the function SCORECONVKERNEL of Alg. 2.

Step2. ($\circledast$ operation) We compute the anomaly scores for the convolution kernels of the same channels in layer one using the function SCORECONVKERNEL. After obtaining all anomaly scores for client 1, we combine them into an anomaly vector A_1. The anomaly vectors A_2 and A_3 for client 2 and client 3 can be obtained in a similar method.

Step3. After computing the anomaly matrices for all layers of client 1, we merge them into a convolution matrix $\texttt{CAM}_1$ using the function SCORECONVMATRIX of Alg. 2. Similarly, we can derive $\texttt{CAM}_2$ and $\texttt{CAM}_3$ for client 2 and client 3.

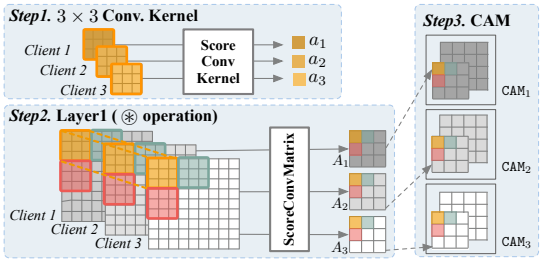


Figure 13: The CAM extraction for each client in a round.

Alg. 2 described functions used in CAM extraction. The function SCORECONVKERNEL computes anomaly scores for a set of convolution kernels ($w_{\{i\in[n]\}}$) in the same channel, where u_i is a convolution kernel for client i. PCA and Mahalanobis Distance[1] are employed to obtain the outliers. The MAIN function computes CAM for n clients. The loop in lines 7-9 computes the anomaly scores for n clients in layer κ. The loop in lines 10-11 generates A_i^κ for each client.

Algorithm 2 SCORECONVMATRIX Algorithm

Input: $\theta_{\{i\in[n]\}}$: updates of n clients; k: total number of conv. layers
Output: $\texttt{CAM}_{\{i\in[n]\}}$: convolution anomaly matrices of n clients

1: **function** SCORECONVKERNEL($w_{\{i\in[n]\}}$):
2: $w_{pca} \leftarrow \text{PCA}(u, \text{componenets} = 2)$
3: $w_{md} \leftarrow \text{MahalanobisDistance}(u_{pca})$
4: $a_{\{i\in[n]\}} \leftarrow \text{Normalize}(u_{md})$ **return** $a_{\{i\in[n]\}}$
5: **function** MAIN:
6: **for** conv. layer $\kappa \in [k]$ **do** ▷ the operation $\circledast$
7: **for** each channel ι **do**
8: $w^\iota_{\{i\in[n]\}} \leftarrow$ Conv. kernel set of channel ι for n clients
9: $a^\iota_{\{i\in[n]\}} \leftarrow$ SCORECONVKERNEL$(w^\iota_{\{i\in[n]\}})$
10: **for** $i \in [n]$ **do**
11: $A_i^\kappa \leftarrow$ Combine all a_i^ι of client i in layer κ
12: $\texttt{CAM}_i \leftarrow$ Generate using A_i^κ and Eq.(1), $i \in [n]$
13: **return** $\texttt{CAM}_{\{i\in[n]\}}$

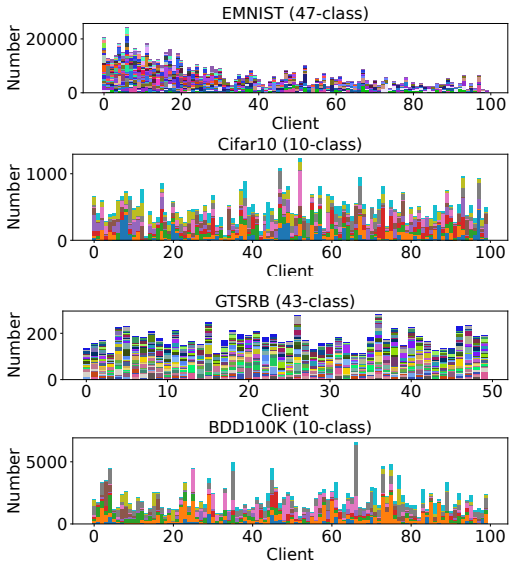


Figure 14: Data distribution of N clients derived from the Dirichlet distribution with $d = 0.5$.

APPENDIX B
EXPERIMENTAL SETUP DETAILS

Details on datasets and model architectures: MNIST [2] is a 10-class class-balanced digit image classification dataset. EMNIST [3] is a 47-class class-imbalanced digit image classification dataset. CIFAR10 [4] is a 10-class class-balanced color image classification dataset. German Traffic Sign Recognition Benchmark (GTSRB) [5] is a 43-class class-imbalanced traffic sign dataset with varying light conditions and rich backgrounds. For more details on datasets, please refer to Table XII (columns 3-5).

Table XII (column 7) lists the model architectures utilized for each dataset. SimpleNet comprises two convolution layers and two fully-connected layers. AlexNet [6], ResNet18 [7], VGG16 [8], and ResNet34 [7] are different architectures of convolution networks. DNN contains two fully connected layers. Vision Transformer (ViT) [9] is a Transformer based model for image recognition.

Details on learning parameters: Following the standard setup

Table XII: Datasets, model structures, and parameters

	Dataset	Training input	Class	Input size	Data distribution	Model structure	N	n^*	P_m^*	P_p^*	Benign lr/epochs	Malicious lr/epochs
Un targeted	MNIST	60000	10	28×28	IID	SimpleNet	100	10	20%	0%	0.01/2	0.01/2
	EMNIST	73168	47	28×28	non-IID	SimpleNet	100	10	20%	0%	0.01/2	0.01/2
	CIFAR10	50000	10	32×32	IID& non-IID	AlexNet	100	10	20	0%	0.1/2	0.1/2
	CIFAR10	50000	10	32×32	IID& non-IID	ResNet18	100	10	10%-40%	0%	0.1/2	0.1/2
Targeted	CIFAR10	50000	10	32×32	IID& non-IID	ResNet18	100	10	10%-40%	3%	0.01/2	0.005/4
	CIFAR10	50000	10	32×32	IID & non-IID	VGG100	100	10	10%	3%	0.01/2	0.005/4
	GTSRB	50000	43	32×32	IID & non-IID	ResNet34	50	10	10%	3%	0.01/2	0.005/4
	HAR	7353	6	1×561	non-IID	DNN	21	10	20%	3%	0.01/2	0.005/4
	BDD100K	474706	10	64×64	non-IID	ViT (Transformer)	100	10	10%	3%	0.0001/4	0.00005/6

$^*n = 20$ for MB and Fang attacks, $P_m = 20\%$ for DBA, $P_p = 5\%$ for dirty label attack.
In MRA, the adversary attacks for one round and scales the updates by $10\times$.

in [10], [11], we train local models with local learning rate lr, local epochs 2, and batch size 64. At each round, the n selected clients train a local model to aggregate. For untargeted attacks, we start with a sustained attack from scratch of the training. We train MNIST and EMNIST with SimpleNet using $lr = 0.01$ for rounds 0-100. We train CIFAR10 with AlexNet and ResNet18 using $lr = 0.1$ for rounds 0-200. For backdoor attacks, we begin the attack when the global model is convergent, which is round 200 for CIFAR10, 100 for GTSRB (IID), and 200 for GTSRB (non-IID). We train CIFAR10 with ResNet18 and VGG16 using $lr = 0.1$ and $lr = 0.05$ for rounds 0-400. We train GTSRB with ResNet34 using $lr = 0.1$ for rounds 0-200 in the IID settings and $lr = 0.1$ for rounds 0-300 in the non-IID settings.

Details on attacks: For Add noise attack, we use Gaussian noise $\delta \sim \mathcal{N}(\mu, \sigma^2)$, where $\mu = 0$ and $\sigma = 0.3$. For Dirty label attacks, we increase the number of sources and target labels to enhance the attack's impact. Based on the different strategies applied to "source-target label" pairs, we evaluate four types of dirty label attacks, including the Fix-Fix attack, the Fix-Rnd attack, the Rnd-Fix attack, and the Rnd-Rnd attack. We set the proportion of poisoned samples for each malicious client to 50%. In the Fix-Fix attack, we mislabel the samples in categories 1 to 5 as category 3. In the Fix-Rnd attack, we mislabel the samples in categories 1 to 5 as random categories. In the Rnd-Fix attack, we mislabel half of the samples to category 3. In the Rnd-Rnd attack, we mislabel half of the samples into random categories. For MB and Fang attacks, we set $n = 20$ because these attacks require more than four malicious clients per round to generate malicious updates. For the backdoor attacks, we set $P_m = 10\%$ for most backdoor attacks and increase the $P_m = 20\%$ for DBA because of the distributed trigger and injection strategy. Following the setup in [11], [12], we attack at the beginning of the training for MRA and at round 200 for other backdoor attacks. In all attacks, adversaries modify their local lr/epochs to attack efficiently.

Appendix C
Assessment of Previous Attacks and Detection

Untargeted attacks: Fig. 15 shows the clean accuracy drop of untargeted and adaptive attacks in non-IID settings. Most untargeted attacks achieve a high clean accuracy drop, except for three dirty label attacks (Fix-Rnd to Rnd-Rnd). Specifically, the clean accuracy drop of MB Mkrum and Fix-Rnd on CIFAR10 (AlexNet) is 39.29% and 0.85%. We also

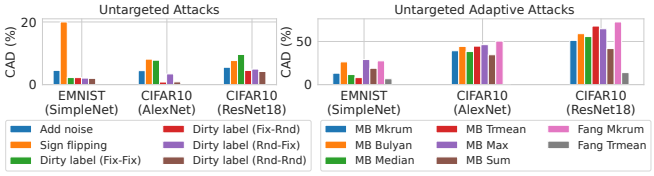


Figure 15: Clean accuracy drop (CAD) of untargeted attacks in non-IID settings. The CAD of the Sign-flipping attack on EMNIST is 81.28%.

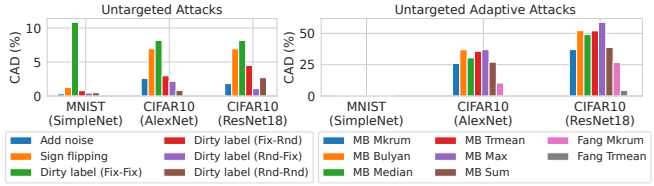


Figure 16: Clean accuracy drop (CAD) of untargeted attacks in IID settings. The CAD of untargeted adaptive attacks on MNIST is less than 0.2%.

observe that the performance of untargeted adaptive attacks is generally stronger than that of untargeted attacks due to the ability of adaptive attacks to generate optimized perturbations. The average clean accuracy drop for adaptive and untargeted attacks is 25.8% and 3.77%, respectively. (see Fig.16 for IID).

Fig. 17 shows the accuracy, model stability, worst/best category accuracy, and category accuracy stability of untargeted and adaptive attacks. We observe that untargeted attacks mainly affect the stability of the model. Specifically, the model stability for the Fix-Fix attack is 4.15, much higher than the clean vanilla model (0.7). In addition, untargeted adaptive attacks significantly affect the accuracy and category accuracy. The accuracy and category accuracy of the MB attack and the Fang attack are smaller than the baseline by 41.82% and 41.15%, respectively. (see Fig.23 and 25 for other datasets).

Remark 1. The 14 untargeted attacks (except for Fix-Rnd to Rnd-Rnd) lead to a significant decrease in the accuracy or stability of the global model.

Targeted attacks: Fig. 18 shows the attack success rate of 16 backdoor attacks in non-IID settings. Most attacks can inject backdoors with almost 100% attack success rate, except for Patch-MRA and Noise-MRA on GTSRB. For different attack strategies, MRA is more difficult to inject backdoors because

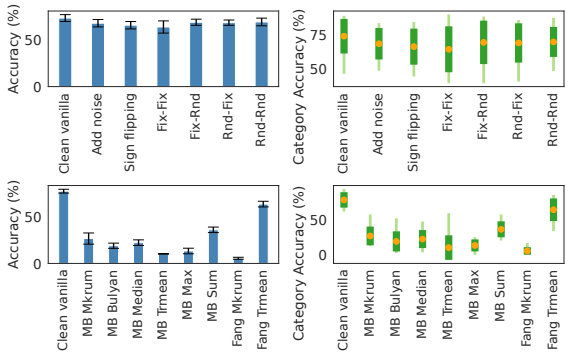


Figure 17: Accuracy and category accuracy of untargeted attacks compared to the clean vanilla on CIFAR10 (ResNet18). Top-untargeted attacks mainly affect stability. Bottom-untargeted adaptive attacks mainly affect accuracy.

MRA only injects a single round. For different datasets, those with fewer categories and more training inputs are easier to inject backdoors. Specifically, the average attack success rate of 96.21% for CIFAR10 is higher than that of 76.11% for GTSRB. In addition, the effect of our new adaptive strategy, FRA, achieves similar results as without it. Specifically, the average attack success rate w/ and w/o FRA is 88.3% and 89.51%, respectively. We also observe that the backdoor injection is unstable and slow at the beginning of the training, as shown in Fig. 20. Due to the low accuracy of the global model, it is difficult to converge the global model and inject backdoors into it simultaneously. (see Fig. 19 for IID).

Remark 2. When the global model converges, backdoors can be injected faster and more successfully, similar to the conclusion in [11].

Fig. 21 shows the backdoor injection rounds and backdoor removal rounds of backdoor attacks in non-IID settings. All attacks with and without Patch-FRA (except MRA) achieve an attack success rate of over 90%, with an average of only 36 and 39 backdoors injection rounds required. In almost all attacks (except MRA), the backdoor removal rounds exceed more than 200 rounds. In Patch-MRA, the adversary injects a backdoor for one round, and the global model takes more than 84 rounds to remove the backdoor. Such small backdoor injection rounds and large backdoor removal rounds imply that backdoor attacks are robust, posing a significant security risk to FL. (see Fig. 24 and Fig. 26 for other datasets).

Remark 3. The 16 backdoor attacks can rapidly inject backdoors and persist for much more training rounds.

Detection against attacks: In non-IID settings, the effectiveness of existing methods against untargeted and backdoor attacks is unstable, as shown in Table VII (column MKrum and DnC) and Table VIII (column Mkrum, FLAME, and DnC). DnC is only effective against a few specific attacks, such as Add noise, MB Median, MB Trmean, and backdoor attacks under ViT, but is less effective against other attacks. The effectiveness of Mkrum is generally low under non-IID because Mkrum can only guarantee convergence of FL, which

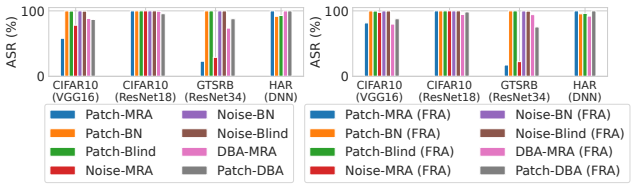


Figure 18: Attack success rate (ASR) of backdoor attacks in non-IID settings.

Figure 19: Attack success rate (ASR) of backdoor attacks in IID settings.

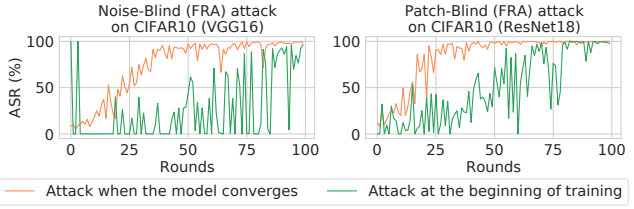


Figure 20: Comparison of attack success rate between the attack when the global model converges and the attack at the beginning of training.

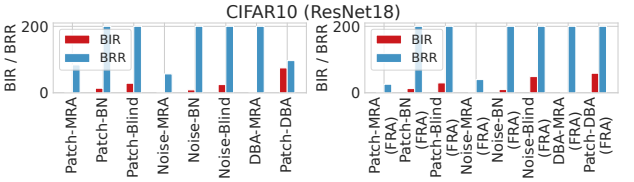


Figure 21: Backdoor injection rounds (BIR) and backdoor removal rounds (BRR) of backdoor attack in non-IID settings. A BIR of 200 indicates that more than 200 training rounds are required to remove the backdoor.

is not sufficient to detect malicious clients [11]. We also observe that existing methods are less effective on complex models and datasets (such as CIFAR10 and ResNet/VGG) because anomalies are more difficult to detect in complex settings.

Remark 4. Existing detection can only ensure high accuracy in detecting a small fraction of specific attacks, sacrificing high false positive rates (FPR) in non-IID settings.

APPENDIX D
DP-BASED DEFENSE AGAINST BACKDOOR ATTACKS

We evaluate the effectiveness of backdoor attacks launched under different DP strategies (see Fig. 22 (top)). We notice

that adversaries can inject backdoors successfully through optimized attack strategies, such as increasing the adversary's local training epochs. Despite DP's inability to defend such optimized attacks, our method is still effective under DP. Fig. 22 (middle and bottom) show the effectiveness of our method against this attack. The results show that FLTracer can prevent backdoor injections (attack success rate$=10\%$) and the global model's accuracy can approximate the no-attack setting.

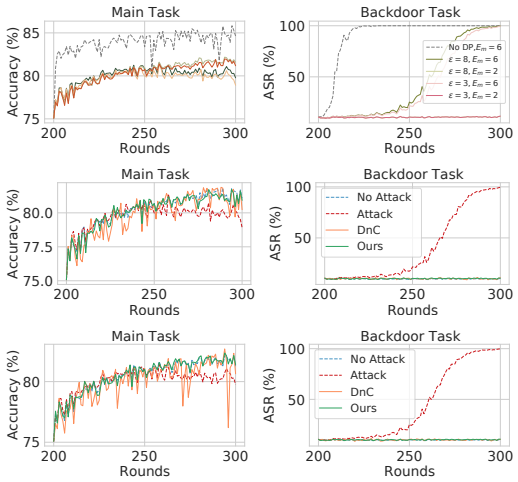


Figure 22: Accuracy and attack success rate (ASR) on CIFAR10 (ResNet18) under DP-based method. Top-under DP (E_m: malicious epochs). Middle-w/ detection under DP with sensitivity $\epsilon = 3$. Bottom-w/ detection under DP with $\epsilon = 8$.

Appendix E
Performance of FLTracer under no attack

Table XIII lists the results of FLTracer under no attack in non-IID settings. It shows that FLTracer has little impact on the global model and increases its stability. In particular, FLTracer achieves a low 8% FPR (under joint), which is close to the sum of the FPRs of the four features. This means that each feature detects malicious clients from different attack locations. The clean accuracy drop of FLTracer decreases by a total of 0.28%, and the model stability decreases by 0.4. This is because without attacks, the updates of the clients with extremely heterogeneous data are significantly different from others, and aggregating these small fractions of updates decreases the global model performance. This observation matches the conclusion in [13]. (see Table XIV for IID).

Table XIII: Detection results under no attack (non-IID).

	Baseline	signv	sortv	classv	featv	Joint
False positive rate	-	1.80%	3.30%	3.00%	1.70%	8.00%
Accuracy	84.38	84.97	84.99	84.49	84.37	84.65
Clean accuracy drop	0	-0.59	-0.61	-0.11	+0.01	-0.28
Model stability	1.06	0.95	0.56	0.68	0.92	0.66

Table XIV: Detection results under no attack in IID settings.

	Baseline	signv	sortv	classv	featv	Joint
False positive rate	-	0.20%	3.80%	0.60%	1.00%	4.50%
Accuracy	89.76	89.86	89.84	90.04	89.86	89.85
Clean accuracy drop	0	-0.10	-0.08	-0.27	+0.10	-0.09
Model stability	0.11	0.08	0.07	0.08	0.08	0.14

Appendix F
Additional Experimental Results

Table XV: Comparing TPR(%) and FPR(%) of DnC and FLTracer (Ours) against backdoor attacks in IID settings.

Dataset (Model)	Attack	DnC [14]		FLTracer (Ours)	
		TPR	FPR	TPR	FPR
CIFAR10 (ResNet18)	Patch-BadNets	100.0	0.00	100.0	0.00
	Noise-BadNets	100.0	0.00	100.0	0.00
	Patch-DBA	98.50	4.13	100.0	0.00
	Patch-Blind	92.00	0.89	100.0	0.00
	Patch-FRA	100.0	0.00	100.0	0.11
CIFAR10 (VGG16)	Patch-BadNets	100.0	0.00	100.0	0.00
	Noise-BadNets	100.0	0.00	100.0	0.11
	Patch-DBA	100.0	0.00	100.0	0.00
	Patch-Blind	100.0	0.00	100.0	0.00
	Patch-FRA	100.0	0.00	100.0	0.00
GTSRB (ResNet34)	Patch-BadNets	100.0	0.00	100.0	0.00
	Noise-BadNets	100.0	0.00	100.0	0.11
	Patch-DBA	93.00	10.25	99.00	0.00
	Patch-Blind	100.0	0.11	100.0	0.00
	Patch-FRA	95.00	23.67	100.0	0.00
Average		98.57	2.60	**99.93**	**0.02**

Table XVI: Comparing TPR(%) and FPR(%) of DnC, and FLTracer (Ours) against untargeted attacks in IID settings.

Dataset (Model)	Attack	DnC [14]		FLTracer (Ours)	
		TPR	FPR	TPR	FPR
MNIST (SimpleNet)	Add Noise	100.0	0.00	100.0	0.00
	Sign-flipping	74.00	10.13	100.0	0.00
	Dirty label (Fix-Fix)	98.50	0.38	98.00	3.63
	MB Mkrum	100.0	0.00	100.0	0.00
	MB Bulyan	100.0	0.00	100.0	0.00
	MB Median	100.0	0.00	100.0	0.00
	MB Trmean	100.0	0.00	100.0	0.00
	MB Max	100.0	0.00	100.0	0.00
	MB Sum	100.0	0.00	100.0	0.00
	Fang Mkrum	100.0	0.00	100.0	0.00
	Fang Trmean	100.0	0.00	100.0	0.00
CIFAR10 (AlexNet)	Add Noise	100.0	0.00	100.0	0.00
	Sign-flipping	99.50	22.50	100.0	0.00
	Dirty label (Fix-Fix)	100.0	1.13	100.0	2.94
	MB Mkrum	99.50	1.13	92.00	1.63
	MB Bulyan	98.50	1.88	92.25	2.69
	MB Median	100.0	0.00	100.0	0.00
	MB Trmean	100.0	0.00	96.00	0.50
	MB Max	99.75	1.50	92.00	2.25
	MB Sum	97.50	4.25	94.50	4.50
	Fang Mkrum	34.25	27.69	100.0	0.00
	Fang Trmean	98.25	2.25	100.0	0.00
CIFAR10 (ResNet18)	Add Noise	99.00	0.25	99.50	0.38
	Sign-flipping	68.00	23.75	100.0	0.13
	Dirty label (Fix-Fix)	99.49	9.38	100.0	1.56
	MB Mkrum	62.50	41.31	100.0	0.19
	MB Bulyan	51.75	45.94	99.25	0.19
	MB Median	100.0	0.00	100.0	0.00
	MB Trmean	100.0	0.00	100.0	0.00
	MB Max	50.25	42.25	100.0	0.06
	MB Sum	33.75	52.69	100.0	0.00
	Fang Mkrum	49.17	48.40	98.61	0.00
	Fang Trmean	19.25	38.63	100.0	0.00
Average		85.85	11.38	**98.85**	**0.63**

(a) MNIST (SimpleNet)

(b) CIFAR10 (AlexNet))

(c) CIFAR10 (ResNet18)

Figure 23: The accuracy and category accuracy of untargeted attacks compared with clean vanilla in IID settings.

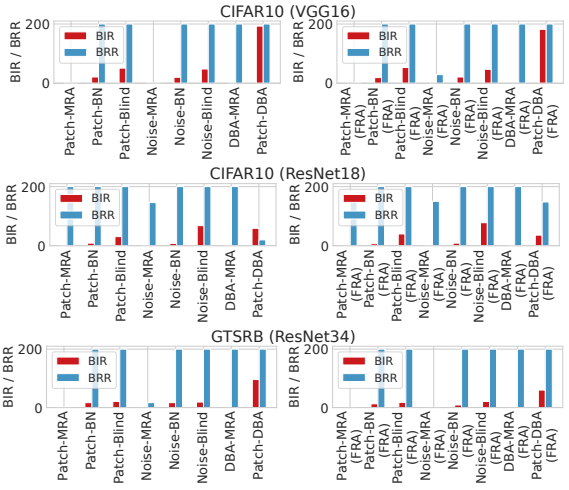
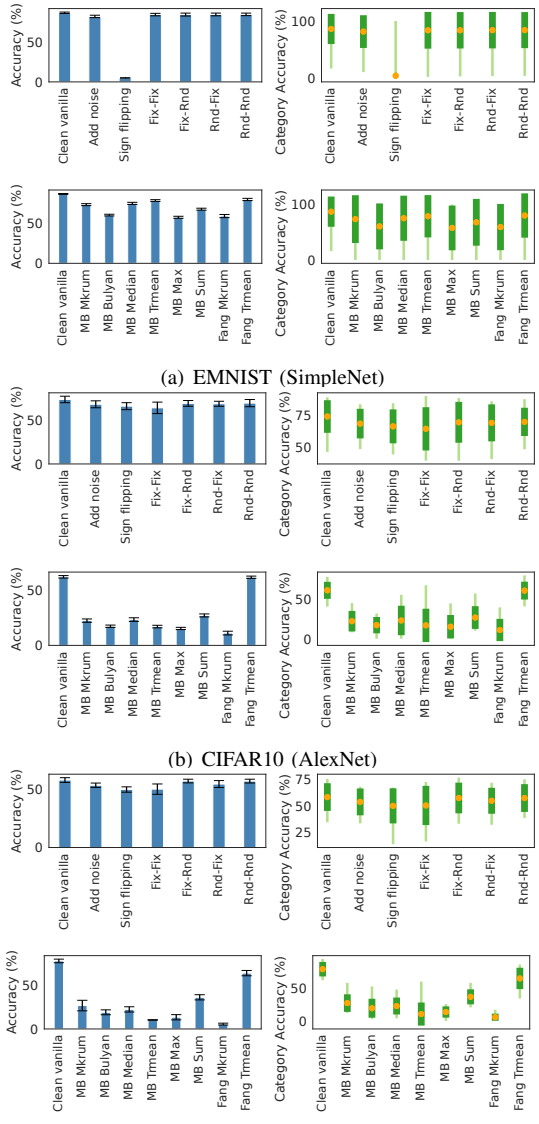


Figure 24: The backdoor injection rounds (BIR) and backdoor removal rounds (BRR) of backdoor attacks in IID settings.



(a) EMNIST (SimpleNet)

(b) CIFAR10 (AlexNet)

(c) CIFAR10 (ResNet18)

Figure 25: The accuracy and category accuracy of untargeted attacks compared with clean vanilla in non-IID settings.

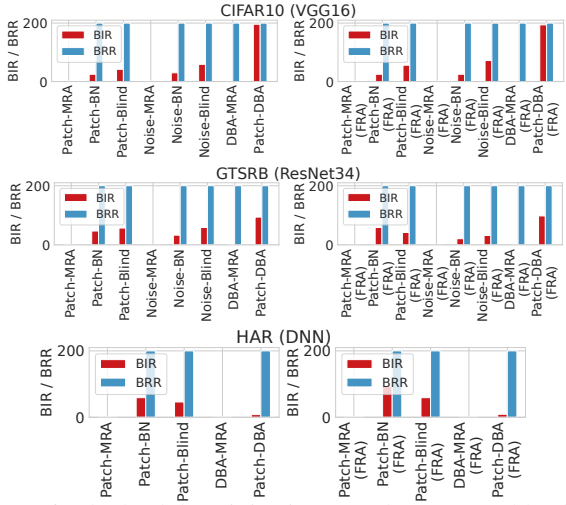


Figure 26: The backdoor injection rounds (BIR) and backdoor removal rounds (BRR) of backdoor attacks in non-IID settings.

REFERENCES

[1] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The mahalanobis distance," *Chemometrics and intelligent laboratory systems*, vol. 50, no. 1, pp. 1–18, 2000.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[3] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "Emnist: Extending mnist to handwritten letters," in *2017 international joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 2921–2926.

[4] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[5] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark," in *International Joint Conference on Neural Networks*, no. 1288, 2013.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[10] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[11] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2938–2948.

[12] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "Dba: Distributed backdoor attacks against federated learning," in *International Conference on Learning Representations*, 2019.

[13] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos, "Attack of the tails: Yes, you really can backdoor federated learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 070–16 084, 2020.

[14] V. Shejwalkar and A. Houmansadr, "Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning," in *NDSS*, 2021.