DROPFL: CLIENT DROPOUT ATTACKS AGAINST FEDERATED LEARNING UNDER COMMUNICATION CONSTRAINTS

Wenjun Qian, Qingni Shen✉, Haoran Xu, Xi Huang, Zhonghai Wu

School of Software and Microelectronics, Peking University, China
National Engineering Research Center for Software Engineering, Peking University, China
PKU-OCTA Laboratory for Blockchain and Privacy Computing, Peking University, China

Introduction

Federated learning (FL) has emerged as a promising paradigm for decentralized machine learning while preserving data privacy. However, under communication constraints, the standard FL protocol faces the risk of client dropout. Although some research has focused on the risk from the perspectives of communication optimization and privacy protection, it is still challenging to deal with the client dropout issue in dynamic networks, where clients may join or drop the training process at any time.

In this paper, we systematically investigate and measure the impact of client dropout on federated learning by considering the offline duration, frequency, and pattern. Our work allows researchers to gain valuable insights into federated learning about potential vulnerabilities. First, we assume an attacker can control a limited subset of clients and manipulate these clients to persistent dropout (PD) or random dropout (RD) in some iterative round during the training process. Then, we simulate a Shapley value-based dropout (SVD) attack to preferentially drop the local model of these controlled clients with highly valuable data per iterative round.

Contributions:

1. We systematically study and measure the impact of client dropout in FL under communication constraints due to unstable network connections or low bandwidth.
2. We specify two attack scenarios and propose three client dropout attacks against FL. Our attacks manipulate a subset of clients to fail in uploading their model updates by employing various strategies.
3. We perform extensive experiments to demonstrate the effectiveness of three client dropout attacks on existing FL algorithms.

Threat Model

Assume that the adversary can compromise a small limited subset of clients and then consider two attack scenarios under communication constraints:

- *Non-knowledge setting*: An adversary $\mathcal{A}_{non}$ can only control the upload communication channel of compromised clients without knowing the aggregation rule in FL.
- *Partial knowledge setting*: An adversary $\mathcal{A}_{par}$ can control the upload communication channel and has access to the local model updates of these controlled clients in order to assess the data valuation, but can not tamper with these model updates.

DropFL

Workflow. The model training process per iterative round can be roughly categorized into six steps per iteration:

① global model broadcast
② local model training
③ client dropout
④ upload parameters
⑤ model aggregation
⑥ update global parameters

Design Goal. In DropFL, our attack goal is to drop out part of clients from the set of controlled clients with various strategies.

- An adversary $\mathcal{A}_{non}$ can induct the *persistent dropout attack* and *random dropout attack*.
- An adversary $\mathcal{A}_{par}$ can perform the *Shapley value-based dropout attack*.

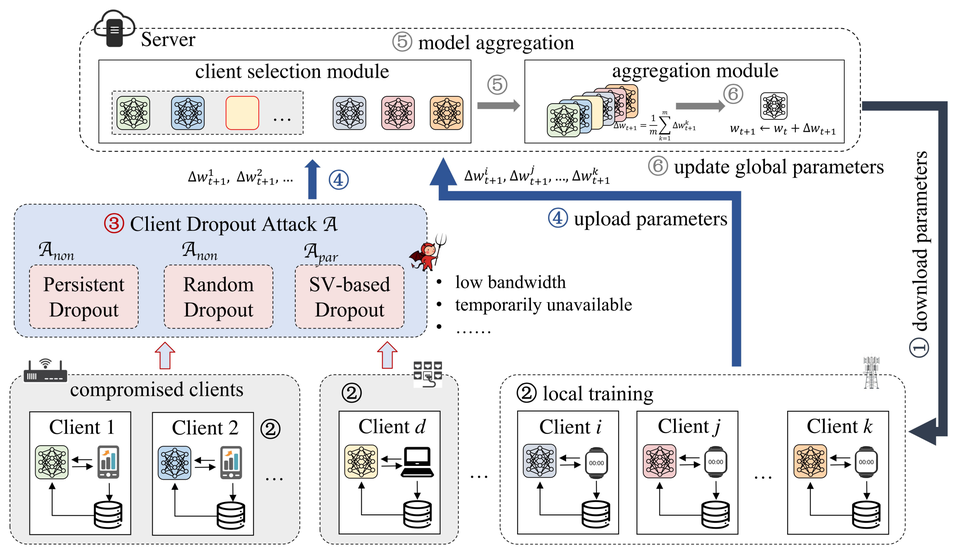

Fig.1. Workflow of the DropFL.

Client Dropout Attacks

1. Persistent Dropout Attack (PD)

The goal of PD attack is to manipulate some clients not to upload local model updates at all, which will make the learned model unavailable, and affect FL tasks in terms of model accuracy and training time. If some controlled clients ($S^{(att)}$) happen to be selected by the server (S_t), indicated as the compromised-then-selected clients (Q_t), each client belonging to the set Q_t trains a local model based on its private data. However, the local model updates from these clients are lost in the aggregation process, where $S_t^{(drop)} \leftarrow S_t \cap S^{(att)}$.

2. Random Dropout Attack (RD)

Compared with the PD attack, the purpose of RD attack is to make the behavior of adversary $\mathcal{A}_{non}$ less detectable by the server. The objective can be achieved by having these compromised-then-selected clients temporarily offline with a certain dropout probability $\tau \in (0, 1)$. At the t-th iterative round, an adversary in the RD attack randomly selects $\lceil |Q_t| \cdot \tau \rceil$ clients from the set Q_t to drop, indicated as the actual dropped clients (set $S_t^{(drop)}$), where $S_t^{(drop)} \subseteq S_t \cap S^{(att)}$.

3. Shapley Value-based Dropout Attack (SVD)

For the SVD attack, an adversary $\mathcal{A}_{par}$ further improves the attack effect by measuring the data valuation of the compromised clients based on the *Shapley Value* mechanism. To this end, $\mathcal{A}_{par}$ preferentially drops the local model of these controlled clients with highly valuable data per iterative round. Specifically, the number of compromised-then-selected clients $|Q_t|$ can be described as formula (1). $\mathcal{A}_{par}$ maintains the SV values of all compromised clients $\{SV_k\}_{k \in Q_t}$ by the *SVEstUpdate* function as formula (2)-(4).

$$|Q_t| \in \begin{cases} [0, max(C \cdot K, 1)], & \text{where } 0 < C < \alpha; \\ [0, max(\alpha \cdot K, 1)], & \text{where } \alpha \le C \le 1 - \alpha; \\ [(C + \alpha - 1) \cdot K, max(\alpha \cdot K, 1)], & \text{where } 1 - \alpha < C \le 1. \end{cases} \quad (1)$$

$$SV_i \leftarrow SV_i + (U(w_t + S[:I]) - U(w_t)), \quad (2)$$

$$SVEstUpdate(w_{t+1}^k, w_t) \leftarrow \{SV_i\}_{i \in Q_t}, \quad (3)$$

$$\{SV_k\} \leftarrow \{SV_k\} + SVEstUpdate(\{w_{t+1}^k\}, w_t), \quad (4)$$

Evaluation

Insight I: Client dropout attacks cause the model accuracy down in DropFL.

The actual number of dropped clients per iterative round is affected by three parameters, including $\alpha, \tau,$ and C. The dropout rate γ of clients is the product of three parameters ($\gamma = \alpha\tau C$). For the compromise fraction α, Table 1 shows that the test accuracy for FedAvg in our attacks decreases as the number of compromised clients increases. For the dropout probability τ, experiment results prove that the performance of the global model will decrease when the dropout probability increases. Besides, Table 1 indicates that the impact of our dropout attack is different under various selection fractions C.

Insight II: Client dropout attacks are effective for impacting the related algorithms.

We conduct the client dropout attacks to some related algorithms, like FedProx and q-FedAvg, which were proposed under the communication constraints and heterogeneity condition. As shown in Table 2, the test accuracy for q-FedAvg under SVD attack decreased by 3.17% compared with that under RD attack. It means that q-FedAvg and FedProx need to further improve the robustness of client dropout attacks.

Insight III: Client dropout attacks gain significant effect with the greater degree of non-IID.

Fig. 2 shows that as the distribution probability increases, our dropout attacks cause a more serious impact. For instance, our SVD dropout attack decreases test accuracy from 47.8% to 44.5% compared with the non-attack, where the distribution probability is 0.8. When $p = 1$, the CNN model on the extremely non-IID CIFAR-10 and VGG model on the non-IID CIFAR-100 do not converge.

Insight IV: Client dropout attacks slow down the model training process in terms of communication rounds.

Table 3 shows the impact of dropout attacks on MNIST by adjusting the number of clients K. We count the number of communication rounds at a target test accuracy of 90%. The experimental results show both RD and SVD attacks slow down the convergence of the training model, which leads to more communication overhead compared with the non-attack.

Table 1. Comparison on test accuracy for FedAvg under various compromise fractions (α), dropout probabilities (τ) and selection fractions (C) respectively in non-IID setting. -: The scenario does not exist.

Algorithms	Dataset	Strategies	The Compromise Fraction α					The Dropout Probability τ				The Selection Fraction C		
			10%	20%	30%	40%	50%	0.25	0.5	0.75	1.0	0.1	0.3	0.5
FedAvg	MNIST	Non-attack	96.59%	96.57%	96.61%	96.58%	96.59%	-	-	-	-	93.68%	95.08%	96.61%
		PD	93.40%	91.40%	90.49%	88.30%	84.50%	-	-	-	90.49%	86.10%	89.01%	90.49%
		RD	95.37%	95.20%	95.05%	94.89%	93.58%	95.47%	95.05%	94.02%	-	92.06%	94.05%	95.05%
		SVD	94.40%	93.68%	92.32%	90.70%	90.03%	94.00%	92.32%	91.28%	-	90.40%	91.32%	92.32%
	CIFAR-10	Non-attack	55.17%	55.18%	55.17%	55.18%	55.16%	-	-	-	-	53.19%	54.12%	55.17%
		PD	52.04%	49.82%	47.53%	43.30%	41.53%	-	-	-	47.53%	42.20%	45.90%	47.53%
		RD	54.34%	53.77%	52.80%	52.68%	52.29%	53.89%	52.80%	52.37%	-	50.30%	51.90%	52.80%
		SVD	54.00%	51.68%	48.81%	47.23%	44.32%	52.10%	48.81%	47.92%	-	44.00%	47.61%	48.81%
	CIFAR-100	Non-attack	44.91%	44.89%	44.90%	44.87%	44.92%	-	-	-	-	38.20%	42.60%	44.90%
		PD	37.50%	36.75%	35.51%	33.78%	30.41%	-	-	-	35.51%	31.40%	34.30%	35.51%
		RD	43.20%	42.29%	40.54%	39.12%	38.10%	42.70%	40.54%	38.30%	-	35.20%	37.80%	40.54%
		SVD	40.90%	38.68%	37.10%	34.93%	32.82%	38.90%	37.10%	36.20%	-	33.90%	35.80%	37.10%

Table 2. Comparison on test accuracy for FedAvg, FedProx and q-FedAvg on MNIST.

Algorithms	Non-attack	PD (diff.†)	RD (diff.‡)	SVD (diff.*)
FedAvg	96.58%	90.49% (-6.09%)	94.11% (-2.47%)	91.26% (-2.85%)
FedProx	96.25%	88.98% (-7.27%)	93.54% (-2.71%)	89.43% (-4.11%)
q-FedAvg	96.75%	91.47% (-5.28%)	94.83% (-1.92%)	91.66% (-3.17%)

†: test accuracy difference between PD and Non-attack. ‡: test accuracy difference between RD and Non-attack. *: test accuracy difference between SVD and RD.

Table 3. Comparison on test accuracy and communication rounds under various numbers of clients for MNIST.

Metrics	K	non-attack	PD	RD	SVD
ACC	20	0.756	0.568	0.720	**0.654**
	50	0.835	0.754	0.806	**0.785**
	100	0.966	0.905	0.951	**0.922**
CR	20	615	-	922 (1.50×)	**1207** (1.96×)
	50	289	-	326 (1.13×)	**493** (1.71×)
	100	47	194 (4.1×)	75 (1.60×)	**116** (2.47×)

ACC: test accuracy. CR: communication rounds. -: not reach the target ACC.

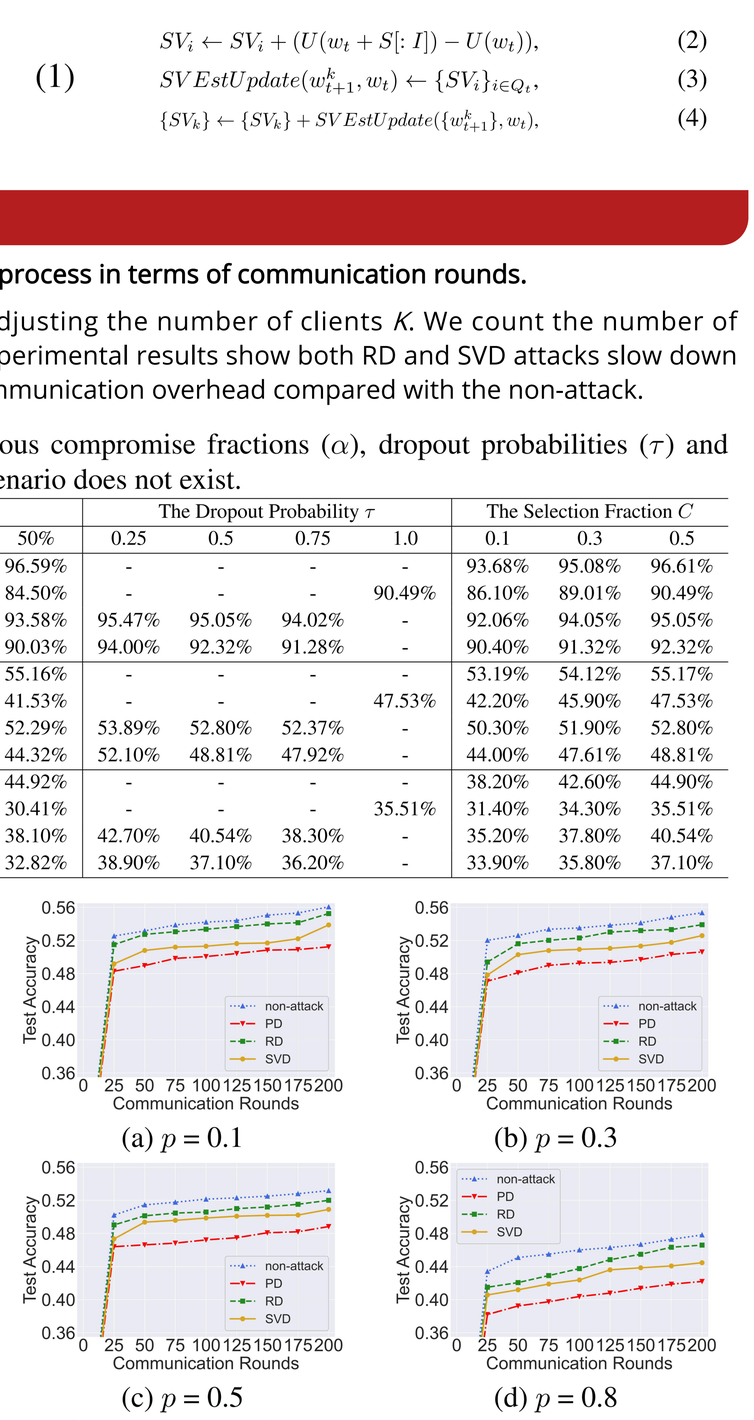

Fig. 2. Test accuracy for dropout attacks under various distribution probabilities on CIFAR-10.