Guided by Noise: Vulnerable Poisoning Attack to Differentially Private Federated Learning

1st Siqi Dai
ECE Department
University of Florida
Gainesville, USA
dais@ufl.edu

2nd Yaodan Hu
ECE Department
Idaho State University
Pocatello, USA
cindyhu@isu.edu

3rd Honggang Yu
ECE Department
University of Florida
Gainesville, USA
honggang.yu@ufl.edu

4th Hanqiu Wang
ECE Department
University of Florida
Gainesville, USA
wanghanqiu@ufl.edu

5th Shuo Wang
ECE Department
University of Florida
Gainesville, USA
shuo.wang@ece.ufl.edu

ICC 2025 - IEEE International Conference on Communications | 979-8-3315-0521-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICC52391.2025.11161449

***Abstract*—Distributed Federated Learning (FL) enables client models to communicate with the central server and collaboratively train a central model while keeping individual datasets localized. Despite its advantages, recent research highlights vulnerabilities in the communication between clients and the central server, which attackers can exploit to compromise client privacy. To protect clients' privacy against such attacks, differential privacy (DP) is widely adopted and implemented during the stochastic gradient descent (SGD) stage. While DP-SGD has shown efficacy in previous studies, our analysis reveals that its implementation inadvertently disperses the distribution of data features. This necessitates the relaxation of alert thresholds during weight verification, introducing unforeseen security risks. In this work, we demonstrate how such relaxed criteria allow malicious clients to execute stealthy targeted attacks, evading detection while corrupting the central model. Specifically, our attack degrades the classification accuracy of selected classes without significantly impacting overall model performance. By leveraging the distortions introduced by DP noise, we precisely undermine the classification accuracy of targeted classes. Experimental evaluations validate the effectiveness of our approach, achieving high success rates. Targeted class accuracy can be reduced to as low as 35%, while the accuracy of non-targeted classes declines by less than 1%. Our attack exhibits exceptional stealth, successfully bypassing widely used defenses, even under independent and identically distributed (i.i.d.) data scenarios.**

***Index Terms*—Privacy-preserving, Federated learning, Differential privacy, Model poisoning**

I. Introduction

Federated learning (FL) is a decentralized machine learning paradigm that enables models to be trained across multiple devices or servers, each holding local data samples, without requiring the exchange of the data itself. Within communication systems, FL offers key advantages, such as preserving privacy, reducing communication bandwidth, and improving overall performance during data transmission. These benefits have spurred extensive interest in FL, leading to its adoption in diverse applications, including healthcare information management [1], IoT system management [2], and beyond. Despite its advantages, studies have revealed that FL models are vulnerable to various attacks, which can lead to privacy breaches [3], degradation of global model accuracy [4], and targeted input misclassification [5]. Among these, inference attacks are particularly concerning as they severely compromise the confidentiality of FL models. Such attacks can extract private information, including precise training data, network architecture, and model parameters [6]. These threats pose significant risks to FL models, especially those deployed on commercial Machine Learning as a Service (MLaaS) platforms.

Anonymous defenses, such as Differential Privacy (DP), have been developed to counter inference attacks. DP introduces noise to data, offering strong privacy guarantees while enabling accurate data analysis. Differential Privacy Stochastic Gradient Descent (DP-SGD) mechanisms, in particular, apply DP noise to gradient updates, effectively safeguarding clients' private data from being inferred [6]. While DP-SGD algorithms have demonstrated controlled and acceptable impacts on overall model utility [7], their class-wise effects remain underexplored. This gap in understanding enables attackers to exploit differential privacy by analyzing class-wise robustness, allowing for more efficient targeted attacks.

Moreover, outlier detection algorithms are often forced to relax detection thresholds to maintain a low false positive rate, as the inclusion of DP noise increases variability. This relaxation further exposes the system to vulnerabilities, granting attackers greater opportunities to execute model poisoning attacks undetected. As demonstrated in the evaluation section, the attack proposed in this paper achieves exceptional stealth, bypassing even the most advanced robust aggregation rules. Based on these observations, we introduce a stealthy model poisoning attack that exploits the detection relaxation induced by differential privacy mechanisms in FL models.

In addition to relaxing detection thresholds, DP noise amplifies the robustness discrepancies across different classes. As demonstrated in [8], class-wise robustness varies significantly under identical attacks and remains consistent across various attack methods. This suggests that class-wise discrepancy is an intrinsic property that is challenging to mitigate through existing defense mechanisms.

Building on this observation, we investigate the class-wise discrepancies introduced by DP noise and leverage them to push selected vulnerable class samples beyond the spatial decision boundary. This enables the execution of a targeted attack based on a strategic (vulnerable, target) class pair selection. Unlike prior targeted attack methods, which determine target objects based on image content (e.g., stop signs or speed signs) [5], our proposed attack evaluates class

stability under DP and identifies the least stable class—termed the vulnerable class—as the attack focus. Concurrently, we select a misclassification direction for the attack, labeling the misclassified class as the target class.

Experimental results highlight the efficacy of this approach. The (vulnerable, target) class pair demonstrates significantly lower attack costs and reduced detection risks compared to suboptimal class pair selections. Conversely, suboptimal choices necessitate higher effort to skew the global model and result in greater distribution distortions during weight verification.

Contributions Table I presents a comparison between our work and existing studies. Unlike prior approaches, we utilize DP noise to enhance the stealthiness of attacks by simultaneously identifying a vulnerable class and a target class to guide misclassification direction. Our method leverages DP noise to examine class-wise robustness discrepancies, enabling the formation of strategic attack class pairs. This approach facilitates highly stealthy and efficient targeted attacks. Moreover, we demonstrate that our proposed attack successfully evades mainstream detection algorithms, even in independent and identically distributed (i.i.d.) data scenarios, achieving a balance of exceptional performance and high stealthiness.

TABLE I: Comparasion to existing works

Method	DP	vulnerable study	attack orientation	stealthiness
Tian et al [8]	✗	✓	✗	✗
DeSMP [9]	✓	✗	✗	✓
Yang [10]	✓	✗	✗	✓
this work	✓	✓	✓	✓

The paper is organized as follows: Section II introduces the federated learning framework and details the construction of the differential privacy federated learning model. Section III presents the proposed threat model and attack methodoloy. Experimental results, evaluating both attack performance and stealthiness, are detailed in Section IV. Lastly, Section V provides a discussion of related work, while Section VI concludes the paper.

II. Model Framework

A. DP-Federated Learning

A general setting of FL is depicted in Fig. 1. Without loss of generality, we consider the state-of-the-art DP-Federated Learning model, the Differential Privacy-Stochastic Gradient Descent (DP-SGD) [11] Federated Learning model. We assume that there are n clients in total. The i-th client train a local model independently on a local dataset S_i containing N_i data sample ($i \in \{1, 2, \ldots, n\}$). The loss function of the local model is denoted as $f(S_i, w_i)$, where w_i denotes the weights of client i's local model. A central server is responsible for aggregating updates from clients and updating the central model. The objective of the central server is to solve the optimization problem $\arg\min_w \frac{1}{n}\sum_{i=1}^{n} f(S_i, w)$, where w is the central model's weights. At round t, the central server dispatches a global weight W^t to each client. Client i trains their local model with weights initizlied as W_i^t and updated as $w_i^{t+1} = W_i^t - \eta \cdot \nabla f(S_i, W_i^t)$, where η is the learning rate. To preserve data privacy, all clients implement local differential privacy (LDP) [12]. The clients can pick any local privacy budget $\epsilon_i < \epsilon_{max}$ for Gaussian noise generation and add the noises to the weights w_i^{t+1}. The noisy weights w_i^{t+1} are uploaded to the central server. Upon receiving local weight updates, the central server validates the updates by evaluating the difference in the weight distribution between the current and the former round. If the difference is lower than a threshold, the local model update will be accepted for the central model updates. Without loss of generality, we assume that the aggregator follows the FedAverage [13] aggregation rule $A(\cdot)$ to update the central model: $W^{t+1} = W^t + A(w_1^t, w_2^t, ..., w_n^t)$.

B. Threat Model

We assume the central server is benign and honest to all included clients. The communication between the central server and clients is secure. We assume that the attacker can compromise and hijack m out of n clients with $m \ll n$. The attacker manipulates weight updates on behalf of the compromised client(s). We assume a black-box setting where the attacker cannot access additional information beyond the compromised clients.

C. Weight Update Validation

To defend against potential Byzantine attacks and maintain higher model performance, the central server can set up an update validation process to decide whether to accept or deny a local update. Below we discuss mainstream countermeasures for defending Byzantine faults.

k-out-of-n Selection Byzantine robust methodologies such as Krum [14] and Bulyan [15] defend against attackers by accepting weights from partial clients during the aggregation process. These methodologies establish robust aggregation rules against Byzantine clients. In the Krum algorithm, the central model weight is updated using the local weight from the client whose updated local weight has the smallest L_p norm distance to the local weights of all other clients. Similar to Krum, Bulyan advances the Byzantine robust aggregation methodology by selecting $k (k \leq n-2m)$ out of n local model updates and m malicious clients. Compared to the Bulyan method, the Krum algorithm offers a more robust aggregation solution suitable for general settings. In this paper, we will focus on the Krum algorithm and show how our proposed attack can be successfully selected in the Krum algorithm.

Distance Measurement [16] proposes an attack that escapes anomaly detection by restricting optimization objectives within limited weight distribution difference constraints. Similarly, previous research [17] has also proposed distance-based detection methods for identifying potential malicious clients and weight update outliers. [18] predicts client updates based on historical model updates and flags the malicious clients if the upcoming weight updates are inconsistent with predicted updates. Likewise, [19] explores a suspicious indicator that

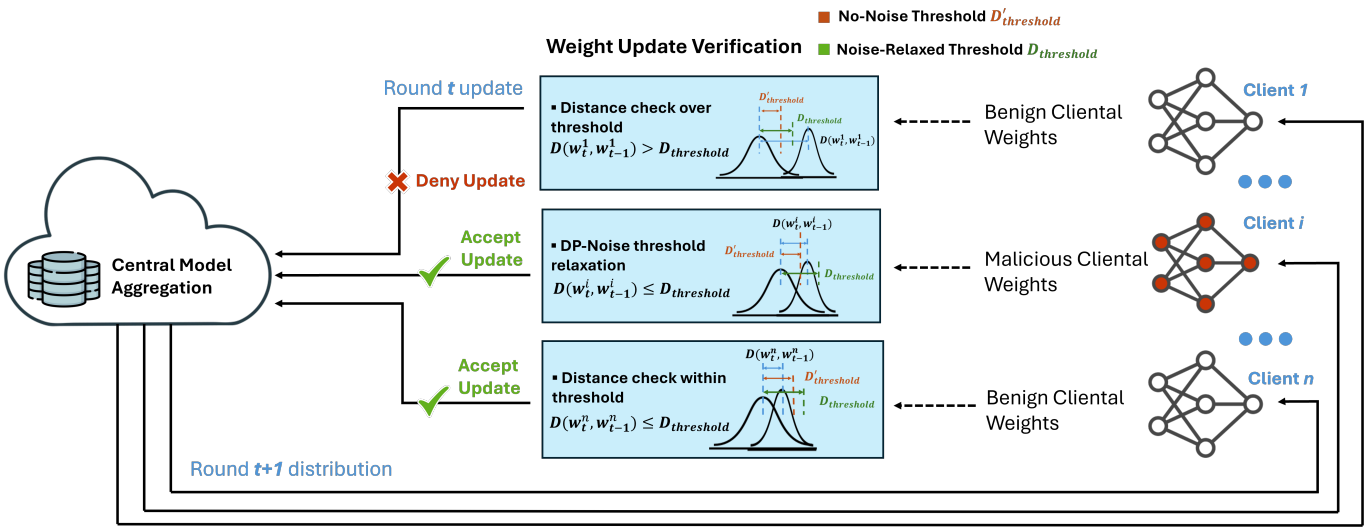


Fig. 1: Poisoning attack and weight verification of the federated learning framework

measures the suspicious level for the features of the n^{th} weight parameter of node m. Their work shows the indicative features of varied datasets and node parameters. In addition, spectrum-based detections [17] demonstrate that benign and malicious updates can be easily differentiated in low-dimensional latent space.

III. Vulnerable Targeted Attack on Federated Learning

A. Class-wise Discrepancy Identification

The targeted attack expects a prediction performance degradation for the specified class while maintaining good performance on other classes. Previous class-wise studies [8] have shown that the robustness varies between classes under the same training task. Multiple factors, including data balance, class variance, spatial distribution, etc, affect such by-class discrepancy. Based on such speculation, we exploit the discriminant score to measure the class spatial separation features and identify the *vulnerable class* within the whole dataset. The Average Pairwise Distance (APD) score measures the mean of Euclidean distances between all possible pairs of samples within a single class and provides a measure of intra-class compactness. Let x_i and x_j be any two data samples within the measured class. The APD score of the class can be calculated through:

$$\text{APD} = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \|x_i - x_j\| \tag{1}$$

A higher APD score indicates a lower degree of clustering, reflecting a higher class-wise vulnerability risk. Moreover, to identify the *target class* for attack orientation, the Fisher's Linear Discriminant Ratio (FLDR) measurement exposes the optimal direction in which the vulnerable class's spatial place has the maximum separability. A higher FLDR score implies larger separability between the two classes. Attacking along the maximum FLDR direction can help the attacker align the attack gradient with the most vulnerable decision boundary. Let μ_k and μ_j denote the mean vectors for class k and j separately, and Σ_w be the within-class scatter matrix, $k \neq j$, the FLDR score between class k and j can be given as:

$$\text{FLDR} = \frac{\left(\boldsymbol{\mu}_j - \boldsymbol{\mu}_k\right)^\top \mathbf{w}}{\mathbf{w}^\top \Sigma_w \mathbf{w}} \tag{2}$$

TABLE II: FLDR score for class pairs and benchmarks

Benchmark	$[C_{vul}, C_{tg}]$	**FLDR score**	C_{vul}, C_{tg}	FLDR score
FashionMNIST	$[5, 7]$	0.525	$[5, 1]$	7.621
Cifar 10	$[3, 7]$	0.021	$[0, 6]$	0.379
Purchase	$[9, 36]$	0.077	$[1, 11]$	0.528

Given the fact that the training accuracy will experience a significant increase within the first few rounds, it is vital to settle the attack targets at the early stage of training. In our experiment, the vulnerable class and the target will be decided within the first 5%th rounds. A comparison of the FLDR score of different datasets and class pairs is shown in Table II. The FLDR score differs significantly for different class pairs, which reflects the misclassified challenge for different classes. Let C_{vul} denote the vulnerable class identified to be attacked, C_{tg} as the target misclassified class. As listed in the table II, the FLDR score for the vulnerable and target class pair is significantly smaller than non-vulnerable class pairs, indicating a huge by-class discrepancy within the dataset.

B. Optimal Targeted Poisoning Attack

f denotes the feature extraction process of the client model that maps the input figure to high dimensional feature space,

and f' denotes the poisoned model learning process. In training, the data loader collects a set of vulnerable class inputs $\mathcal{D}_{vul} = \{x_{v1}, x_{v2}, \cdots, x_{vn}\}$, corresponding target class set is $\mathcal{D}_tg = \{x_{t1}, x_{t2}, \cdots, x_{tf}\}$. The input pairs of the vulnerable class are (x_{vi}, τ_{vi}), (x_{ti}, τ_{ti}), respectively. x represents for the input and τ for the corresponding labels. To maximize the optimization performance, we adopt a differentiable and symmetric metric, Jensen-Shannon Divergence (JSD), rather than normal metrics such as KL-Divergence for better gradient optimization.

$$\mathcal{L}_1 = \frac{1}{N} \sum_{x_i \in D_f} D_{jsd}(f(x_i), f'(x_i)) \tag{3}$$

From the class feature aspect, we push the sample features in the spatial space through a squared L2 norm constraint. The squared L2 norm constraint D_{mse} maintains similarity between the feature representations for data samples from the vulnerable class and the target class.

$$\mathcal{L}_2 = \frac{1}{N} \sum_{x_i \in D_f} D_{mse}(f(x_i), f'(x_i)) \tag{4}$$

To maintain a good utility in other classes and to keep updating weight consistency with other benign clients, we attach $\mathcal{L}_0$ that measures the original loss (e.g., cross-entropy loss) of all other classes. After defining these semi-loss items, we formulate the problem as an optimization problem. Thus, the adversarial objective can be summarized as:

$$\min \mathcal{L}_{x_i \in D_f} = \mathcal{L}_{total} = \mathcal{L}(D_i, w_i^t) + \lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_2 \tag{5}$$

λ_1 and λ_2 are hyperparameters to balance the terms. As the number of corrupted clients ranges from 10% to 20%, the corrupted client needs to skew the model by defending against all other clients. Thus hyperparameters are set as $\lambda_1 = 10$, $\lambda_2 = 10$ for MR = 10%, and $\lambda_1 = 5$, $\lambda_2 = 5$ for MR = 20%.

IV. Evaluation

A. Experimental Setup

We evaluate the performance of the proposed attack on three de facto benchmark datasets, FashionMNIST[20], CIFAR-10[21] and Purchase[22]. All data analyses are carried out on a deep learning server equipped with an Intel Xeon(R) E5-2623 v4 2.60GHz CPU, 128GB RAM, running Red Hat Enterprise Linux 7, and accelerated by an NVIDIA Tesla V100 GPU. Our model is implemented with the PyTorch framework and includes three layers of convolutional neural networks. The training process involves the deliberate tuning of hyperparameters, including the learning rate, number of epochs, and batch size, to optimize performance and ensure a robust evaluation of our attack methodologies.

B. Vulnerable Optimal Attack Evaluation

In section III-A, we have identified vulnerable, non-vulnerable, and target class pairs for three benchmarks. Here we evaluate the experimental results for our proposed optimal attack. Here, Malicious Rate (MR) measures the proportion of malicious clients over the total number of clients that are selected for each training round. For a fair evaluation, the attacker's capability is deliberately limited. We assume that the attacker can compromise at most 20% of clients, which amounts to 2 out of 10 in a small client group and 20 out of 100 for large scenarios. The lower bound of MR is 10%. To ensure a rapid convergence, a sufficient number of clients must be selected for each round. A lower bound of 10% ensures that at least one malicious client is included when considering a group of 10 small client models, which is typical in real-world systems. We also define the following metrics to measure the attack performance:

By-Class Accuracy Drop (AD_{class}) By-class accuracy Drop represents the decrease in accuracy before and after the attack for the vulnerable class. We denote the before-attack accuracy for class i at round t as acc_i^t. The after-attack accuracy is $\hat{acc}_i^t$, the by-class accuracy drop is: $AD_i^t = \frac{acc_i^t - \hat{acc}_i^t}{acc_i^t}$.

Global Accuracy Drop (AD_g) We denote the validated before-attack accuracy of the central model at round t as acc_g^t, and after-attack accuracy as $\hat{acc}_g^t acc_g^t$, the global model accuracy drop is: $AD_g^t = \frac{acc_g^t - \hat{acc}_g^t}{acc_g^t}$

Table III summarizes the performance of each benchmark dataset. The global model converges when reaching 80 and 100 training rounds and arrives at the training stop point.

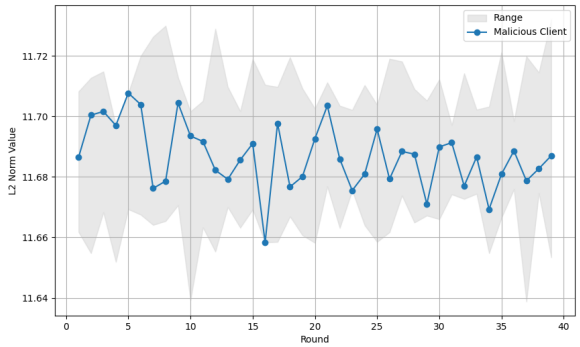
Bnenchmark	MR(%)	Round	AD_{class}	AD_g
FashionMNIST [20]	10	80	15.2%	0.97%
		100	14.1%	1.09%
	20	80	26.7%	0.73%
		100	28.2%	0%
Purchase [22]	10	80	58.8%	9.15%
		100	40.4%	9.15%
	20	80	65.5%	12.1%
		100	61.5%	11.6%
Cifar10 [21]	10	80	19.2%	7.3%
		100	19.2%	7.5%
	20	80	19.7%	10.1%
		100	19.2%	12.6%

TABLE III: Attack performance for selected vulnerable class pairs

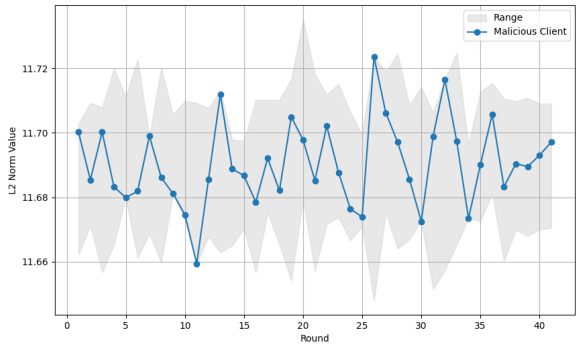
C. Stealthiness Assessment

We first define several metrics to evaluate the attack method's ability to evade malicious detection. Statistical malicious detection algorithms [17] measure the updated weights with both horizontal and vertical consistency. Horizontal consistency states the alignment of one client's updated weight with other client weights for each training round. Vertical consistency checks the alignment of the current weight update w_t for round t with history updates $\{w_i, \cdots, w_{t-1}\}$. In the evaluation of stealthiness, we use the L_2 norm to measure the distribution change of the updated weights for the clients. The

detection of outlier data can be done through the clustering of the detection target [19], where malicious updates can be clustered away from the benign group. Following such idea, for client i, the central server calculates the $D_{L_2}(\delta_i^t, \delta_i^{t-1})$. The weight updating range for all clients is given by: $\mathcal{R} = [\min_{i\in n} \delta_i^t, \max_{i\in n} \delta_i^t]$, as shown in the grey range in Fig 2. For every client, we measure the L2 norm of their current round updated weight and previous-round weight. The maximum and minimum L2 range is given by the grey scope, and the blue curve gives the L2 value of the malicious client updates. Fig 2a shows the malicious L2 are better covered in benign ranges, while the non-vulnerable updates in Fig 2b are located closer to the edge and have a higher risk of being detected.



(a) Vulnerable class pairs attack



(b) Non-vulnerable class pairs attack

Fig. 2: Comparison of L_2 Norm of Malicious Client with Range

Measured statistically on the global model, Fig. 3 shows a comparison of global weight distribution change measured with KL-Divergence. Here we measured the distribution difference of the aggregated global weight between round t and $t+1$. The orange bars represent no attack cases, the blue bars represent attack cases on non-vulnerable class pairs, and the green bars represent attacks on vulnerable class pairs. A smaller value indicates that the global weight differs less and the global model is running in a consistent direction along with the training process, while a larger value indicates

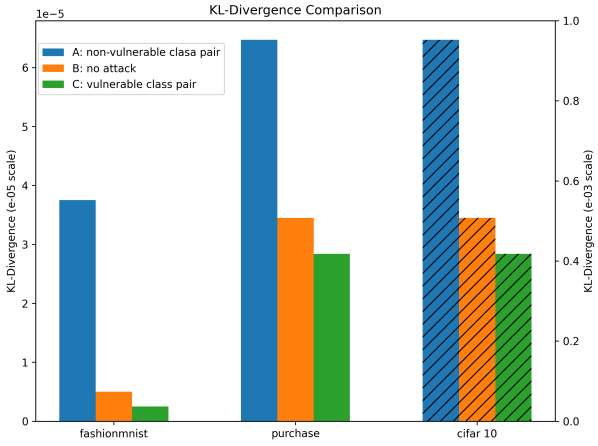


Fig. 3: Global Weight KL-Divergence

higher risk in outliers, and the global model fails to grow in a consistent orientation. Obviously, the global weight KL-Divergences of the vulnerable class pairs are closer to no attack cases, while the value of non-vulnerable class pairs is much higher, reflecting a higher risk of being detected by verification methods. To compare three benchmarks in one figure, the Fashionmnist and purchase bars are in $e-5$ scale and the dashed cifar10 bars are in $e-3$ scale. This difference can be explained by the inherent character of the datasets where simpler datasets such as FashionMNIST and Purchase share lower training difficulty while the complex Cifar10 dataset is inherently more challenging to train.

Similar stealthiness can also be assessed through the k-out-of-n defense measurement. Here we adopt the widely studied Byzantine resilience method Krum [14] as the detection criteria. Krum defends Byzantine clients by accepting only the most similar update weight among the clients into the aggregation process. The defender can set $max(\eta) = \lfloor \frac{n-2}{2} \rfloor$ potential malicious clients for different defending levels. As shown in Fig. 4, the malicious clients maintain a high selection rate even under a high defense level. We observed that the selection rate shows high discrepancy even between benign users, thus to be fair figure 4 shows the selection ratio of both malicious clients and benign clients w/o attack.

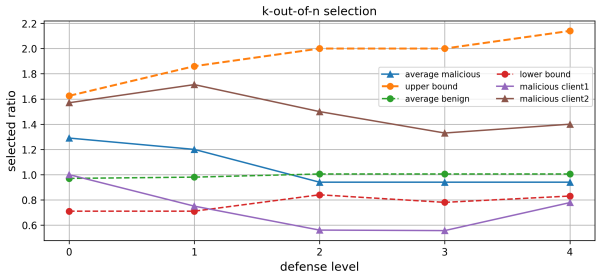


Fig. 4: k-out-of-n selection

The solid lines represent the selection ratio for malicious clients, while the dashed lines denote the selection rates for benign users. Notably, the overall selection ratio of malicious clients remains closely aligned with that of benign clients, even

as the defense level increases. These results indicate that the proposed attack maintains a high level of stealthiness under selection-based defense mechanisms.

Given the novelty of our work, direct comparisons with existing methods are limited. However, we provide references to related works for context. For future research, we suggest several potential defenses against the proposed attack. First, defenders could explore latent space analysis [23], where manipulated updates may exhibit a distinct separation from normal updates at higher levels of feature extraction. Additionally, multi-step processing techniques, such as encoding and decoding [24], could be employed to assign differentiable scores to benign and adversarial updates, thereby enhancing defense effectiveness.

V. Related Works

Poisoning attacks on federated learning models The vulnerabilities of FL have been well studied. [25] attacks FL models by distorting original models into attacker-chosen low-accuracy models. Besides such attacks, model poisoning attacks are more systematically studied. Untargeted poisoning attacks focus on attacking the central model's overall performance. For instance, [26] launches an untargeted attack on the training phase of the FL model by controlling several clients and sending manipulated weight updates. By optimizing attack objectives against selected defense rules, their proposed attack can run smoothly without being detected. In more practical settings, however, untargeted attacks are prone to be detected as the model accuracy will drop noticeably. On the contrary, targeted attacks, which degrade the classification accuracy of specific classes, are easier to keep stealthy.

Differential privacy vulnerabilities Recent studies investigate the attacks targeting differential privacy schemes. The authors in [27] explore the poisoning attack on non-learning-based local differential privacy (LDP) protocols. In their attack, a small number of users are compromised to manipulate inputs, thus skewing the aggregation results of the LDP protocol. Similar ideas are also introduced in [28], [29] on LDP protocols for frequency destination and heavy hitter (most frequent items) identification, such as kRR[30], OUE[31], PEM[32], etc. These works launch poisoning attacks by analyzing the aggregation rules of the LDP protocols and formulating objective functions based on the aggregation methods. However, compared to those explicitly defined in differential privacy protocols, the aggregation rules of machine learning models are more complex and hard to manipulate directly. Our work explores the unexpected security risks brought by differential privacy in the learning-based aggregation scenario.

VI. Conclusion

In this paper, we investigate overlooked security vulnerabilities in DP-SGD-based FL models. Empirical evidence reveals that DP introduces uneven harm across different classes. Building on these findings, we introduce a fragile poisoning attack targeting differentially private FL. By identifying the fragile class within the dataset and optimizing the attack objective, attackers can execute a low-cost, efficient, and highly covert attack. Our evaluation demonstrates that the proposed attack effectively evades detection, even under stringent malicious identification algorithms, and retains its potency when subjected to Byzantine-resilient aggregation rules. Additionally, this work sets the stage for the development and testing of novel defenses to address these emerging threats.

Acknowledgment

This research was supported by the National Science Foundation (NSF) under award numbers 2028897 and 2019283.

References

[1] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," *Journal of healthcare informatics research*, vol. 5, pp. 1–19, 2021.

[2] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, "A survey on federated learning for resource-constrained iot devices," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 1–24, 2021.

[3] Z. Wang, Y. Huang, M. Song, L. Wu, F. Xue, and K. Ren, "Poisoning-assisted property inference attack against federated learning," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[4] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *International Conference on Machine Learning*, pp. 634–643, PMLR, 2019.

[5] J. Jia, Y. Liu, and N. Z. Gong, "Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning," in *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 2043–2059, IEEE, 2022.

[6] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE symposium on security and privacy (SP)*, pp. 739–753, IEEE, 2019.

[7] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," *arXiv preprint arXiv:2003.02133*, 2020.

[8] Q. Tian, K. Kuang, K. Jiang, F. Wu, and Y. Wang, "Analysis and applications of class-wise robustness in adversarial training," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 1561–1570, 2021.

[9] M. T. Hossain, S. Islam, S. Badsha, and H. Shen, "Desmp: Differential privacy-exploited stealthy model poisoning attacks in federated learning," in *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*, pp. 167–174, IEEE, 2021.

[10] M. Yang, H. Cheng, F. Chen, X. Liu, M. Wang, and X. Li, "Model poisoning attack in differential privacy-based federated learning," *Information Sciences*, vol. 630, pp. 158–172, 2023.

[11] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

[12] B. Bebensee, "Local differential privacy: a tutorial," *arXiv preprint arXiv:1907.11908*, 2019.

[13] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.

[14] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in neural information processing systems*, vol. 30, 2017.

[15] R. Guerraoui, S. Rouault, *et al.*, "The hidden vulnerability of distributed learning in byzantium," in *International Conference on Machine Learning*, pp. 3521–3530, PMLR, 2018.

[16] J. Giraldo, A. Cardenas, M. Kantarcioglu, and J. Katz, "Adversarial classification under differential privacy," in *Network and Distributed Systems Security (NDSS) Symposium 2020*, 2020.

[17] S. Li, Y. Cheng, W. Wang, Y. Liu, and T. Chen, "Learning to detect malicious clients for robust federated learning," *arXiv preprint arXiv:2002.00211*, 2020.

[18] Z. Zhang, X. Cao, J. Jia, and N. Z. Gong, "Fldetector: Defending federated learning against model poisoning attacks via detecting malicious clients," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2545–2555, 2022.

[19] S. Shen, S. Tople, and P. Saxena, "Auror: Defending against poisoning attacks in collaborative deep learning systems," in *Proceedings of the 32nd annual conference on computer security applications*, pp. 508–519, 2016.
[20] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.
[21] A. Krizhevsky, V. Nair, and G. Hinton, "Learning multiple layers of features from tiny images," tech. rep., University of Toronto, 2009.
[22] C. O. Sakar, S. O. Polat, M. Katircioglu, and Y. Kastro, "Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and lstm recurrent neural networks," *Neural Computing and Applications*, vol. 31, no. 10, pp. 6893–6908, 2019.
[23] N. Wang, Y. Xiao, Y. Chen, Y. Hu, W. Lou, and Y. T. Hou, "Flare: defending federated learning against model poisoning attacks via latent space representations," in *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, pp. 946–958, 2022.
[24] T. D. Luong, V. M. Tien, N. H. Quyen, D. T. T. Hien, P. T. Duy, and V.-H. Pham, "Fed-lsae: Thwarting poisoning attacks against federated cyber threat detection system via autoencoder-based latent space inspection," *arXiv preprint arXiv:2309.11053*, 2023.
[25] X. Cao and N. Z. Gong, "Mpaf: Model poisoning attacks to federated learning based on fake clients," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3396–3404, 2022.
[26] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to {Byzantine-Robust} federated learning," in *29th USENIX security symposium (USENIX Security 20)*, pp. 1605–1622, 2020.
[27] A. Cheu, A. Smith, and J. Ullman, "Manipulation attacks in local differential privacy," in *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 883–900, IEEE, 2021.
[28] X. Li, N. Li, W. Sun, N. Z. Gong, and H. Li, "Fine-grained poisoning attack to local differential privacy protocols for mean and variance estimation," in *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 1739–1756, 2023.
[29] X. Cao, J. Jia, and N. Z. Gong, "Data poisoning attacks to local differential privacy protocols," in *30th USENIX Security Symposium (USENIX Security 21)*, pp. 947–964, 2021.
[30] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," *Advances in neural information processing systems*, vol. 27, 2014.
[31] T. Wang, J. Blocki, N. Li, and S. Jha, "Locally differentially private protocols for frequency estimation," in *26th USENIX Security Symposium (USENIX Security 17)*, pp. 729–745, 2017.
[32] T. Wang, N. Li, and S. Jha, "Locally differentially private heavy hitter identification," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 2, pp. 982–993, 2019.