Stuck in the SAND: When Your Neighbor Becomes Your Enemy

Tre' R. Jeter[1], Minh N. Vu[1], Raed Alharbi[2], and My T. Thai[1]

[1] University of Florida, Gainesville FL, USA
[2] Saudi Electronic University, Riyadh, Saudi Arabia
Email: {t.jeter, minhvu}@ufl.edu, ri.alharbi@seu.edu.sa, mythai@cise.ufl.edu

Abstract. Decentralized Federated Learning (DFL) has emerged as a powerful paradigm for collaborative model training across distributed devices. However, this distributed nature introduces new security challenges, including the threat of selection attacks via neighbor deception (SAND). In this paper, we investigate vulnerabilities arising from malicious clients seeking to manipulate both the neighbor selection process and the data distribution of other participants. Employing a neighbor selection mechanism that utilizes a similarity metric, clients exchange statistical information to identify correlated neighbors. Our analysis reveals that a malicious client can exploit this mechanism by imitating a victim's statistical profile to maximize their similarity score, thereby securing their position as neighbors. Subsequently, they gain direct access to the victim's model update process. With this access, a malicious user can facilitate the injection of corrupted updates that result in misclassifications during the victim's training process. Our study underscores the importance of robust security measures in DFL and shed lights on potential countermeasures to mitigate the impact of SANDs.

Keywords: Decentralized Federated Learning · Neighbor Selection · Backdoor Attacks

1 Introduction

Federated Learning (FL) has emerged as a transformative paradigm in the domain of machine learning, enabling the collaborative training of models across distributed devices while preserving data privacy. In traditional machine learning approaches, data is centralized for model training, raising concerns about privacy breaches and data silos. FL addresses these concerns by allowing devices to collaboratively train models while keeping their data localized. This decentralized approach offers several compelling advantages, such as enhanced privacy preservation, reduced communication overhead, and the potential for improved model generalization. Because of these benefits, FL has encouraged the growth of numerous domains such as healthcare, autonomous driving, and mobile application development.

A natural progression from traditional FL is the evolution into Decentralized Federated Learning (DFL). In this advanced paradigm, devices not only contribute to model training but also play a role in network governance. The network forms a dynamic, interconnected graph where devices engage in localized model updates and information exchange. This decentralized structure amplifies the advantages of FL by enhancing scalability, robustness, and autonomy. Each device operates as an independent node, enabling efficient and distributed model training while reducing the reliance on central authorities. Although the employment of a DFL system mitigates server-based threats common in traditional FL [1–4], this decentralized architecture introduces new security challenges that need to be thoroughly explored and addressed.

As DFL gains traction, a crucial consideration is the emergence of novel security threats that exploit the inherent vulnerabilities of distributed systems. While traditional FL has laid the foundation for data privacy and model security, the decentralized variant introduces new avenues for potential attacks. One such uncharted territory is the **S**election **A**ttack via **N**eighbor **D**eception (SAND). In this attack scenario, a malicious participant manipulates the neighbor selection process and mimics the statistical attributes of legitimate users to secure a place within their network. By infiltrating their neighbors' circle, the attacker can unleash adversarial updates, undermining not only individual model training but also the collective learning process of the entire network. The implications of such attacks on model integrity and system performance are substantial and warrant comprehensive investigation and mitigation strategies. This paper delves into these unexplored attack scenarios, shedding light on their implications and proposing strategies to safeguard the integrity of decentralized FL systems.

Contributions. Our key contributions are as follows:

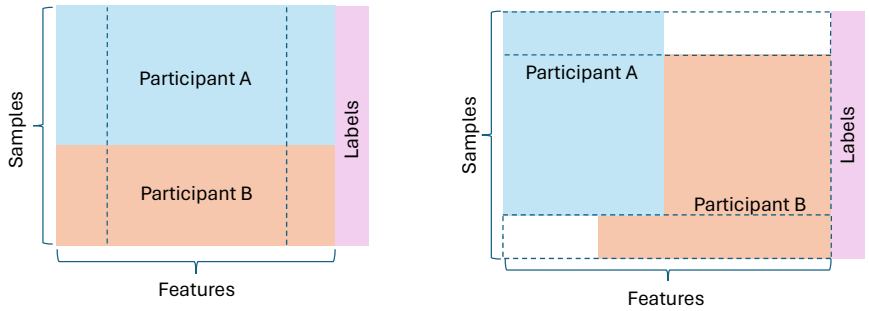
- We introduce and characterize SAND, a novel threat specific to decentralized FL. This attack exposes an unexplored vulnerability, wherein adversaries manipulate the neighbor selection process to compromise the collaborative learning environment. To the best of our knowledge, this is the first attack proposed within the context of decentralized FL.
- We thoroughly analyze the consequences of SAND on a victims' model while emphasizing the severity of the attack's impact as tainted updates propagate through the broader network.
- Building upon our analysis, the paper provides a general discussion regarding current robust defense mechanisms designed to mitigate backdoor attacks in FL. While discussing the possible limitations of these current methods, we also propose potential defenses designed specifically for DFL.

Organization. The paper's structure is as follows: Section 2 provides a primer on traditional FL and DFL. Section 3 presents our threat model, problem setup, and our proposed SAND attack. Section 4 presents an in-depth experimental analysis and results supporting our claims. Section 5 discusses related research on DFL and attacks mentioned in this paper. Section 6 is a brief discussion about possible countermeasures for our SAND attack. Section 7 concludes the paper, summarizing our key findings.

2 Preliminaries

2.1 Centralized Federated Learning

In traditional Federated Learning (FL), a centralized server is used to orchestrate the training of a global model between many clients. FL can be implemented horizontally (hFL) or vertically (vFL) depending on how the data is partitioned between clients. In a horizontal setting, clients maintain data consisting of the same features, but different samples (Figure 1a). The opposite is true for the vertical setting where clients maintain data consisting of the same samples, but different features (Figure 1b). Each style has been thoroughly challenged with attack scenarios such as client selection [1, 5], membership inference [6, 7], and data reconstruction [8, 9] that all undermine FL's privacy guarantee.



(a) hFL (b) vFL

Figure 1: Two common styles of centralized FL.

2.2 Decentralized Federated Learning

Decentralized Federated Learning (DFL) offers a distributed approach to collaborative model training, differing from centralized FL. In a decentralized framework, clients collaboratively train a global model without relying on a central server for coordination. This approach enhances privacy by mitigating the need to share any information with a central entity. Moreover, DFL is inherently more robust to failures since there is no single point of control. DFL thus emerges as a privacy-preserving, fault-tolerant, and scalable paradigm that addresses some limitations of centralized FL [10].

In particular, each client independently initializes its local model and refines it through iterative updates based on its local dataset. The training process involves two key steps. Firstly, clients selectively communicate with a subset of the network, sharing information like model parameters. This communication is governed by metrics such as client performance, data distribution, or network relationships. Secondly, each client aggregates the information received from its chosen neighbors to update its local model. Common methods of FL aggregation, similar to centralized FL, are Federated Stochastic Gradient Descent (FedSGD)

or Federated Averaging (FedAvg). This decentralized process repeats iteratively until the models across clients converge to a collectively improved state, showcasing the power of collaborative learning without centralized control.

3 SAND - A Proposed Attack

3.1 Threat Model

We consider a decentralized Federated Learning (FL) scenario where K clients engage in communication within a network spanning T rounds. Each client, indexed as i, holds a private training data distribution denoted as $\mathcal{D}_i$, comprising model inputs x and corresponding training labels y. Furthermore, every client maintains a neural network f_i governed by parameters $\theta_i \in \mathbb{R}^d$. The training loss for client i is characterized by $\mathcal{L}(f_i(\theta_i, x), y) : \mathbb{R}^d \to \mathbb{R}$. The primary objective for clients is to leverage their individual data and models from fellow clients across the T communication rounds to converge towards a solution for the following optimization problem:

$$\min_{\theta \in \mathbb{R}^d} \mathcal{L}_{\mathcal{D}_i}(f_i(\theta)) := \min_{\theta \in \mathbb{R}^d} \mathbb{E}_{x_i, y_i \sim \mathcal{D}_i} \left[\mathcal{L}(f_i(\theta, x_i), y_i) \right] \tag{1}$$

for all $i = 1, \cdots, K$.

To attain the optimization defined in Equation 1, a client i in decentralized FL must identify suitable neighbors for collaborative training. Ideally, these neighbors should exhibit data distributions similar to that of client i. In this work, we consider two decentralized protocols that hinge on this fundamental principle, decentralized adaptive clustering (DAC) [11], and Random Gossip [12].

Following [11], DAC aims to elevate communication efficiency among clients sharing similar local data distributions. In this approach, each client computes a probability vector using a similarity measure, and this vector guides weighted sampling to select neighbors during each communication round. The inverse of the training loss serves as a proxy for similarity, and the resulting score S undergoes continual updates over T iterations, defined as follows:

$$s_{ij}^t = \frac{1}{\mathcal{L}(f_i(\theta_i, x_i)^t, d_j)} \tag{2}$$

Here, d_j represents the data of a client j. DAC encompasses a dynamic process that involves sampling a subset of clients, updating similarity probabilities, and aggregating models using the FedAvg algorithm. To address the presence of unsampled clients, an approximated similarity score is defined based on two-hop neighbors. The models are subsequently aggregated and trained locally through an iterative process. This strategic approach optimizes communication by fostering collaboration primarily between clients with comparable data distributions.

In contrast, the well-established Random Gossip communication method follows a different paradigm. Each client initiates with a randomly initialized model,

subject to updates through stochastic gradient descent on local data for a predefined number of local epochs. After completing its local update process, a client communicates its model parameters to a randomly chosen neighboring client. The receiving client patiently awaits a predefined number of models before performing averaging using the FedAvg algorithm. This communication cycle persists across all clients until convergence is achieved.

In our threat model, we designate one client out of the K clients within a DAC or Random Gossip-based network as malicious. The primary objective of this malicious client is to compromise the local model of a specific victim client situated within the network. Adding complexity to the scenario, the victim client is not obligated to select the malicious client as its neighbor during the collaborative learning process. Consequently, the malicious client faces a dual challenge–it must not only tamper with the model parameters but also manipulate the selection process itself. This implies that, beyond the tampering objectives, the attacker must meticulously design the parameters θ to mimic the victim's data distribution convincingly. In the subsequent sections, we will delve further into the details of our devised attack and explain how it accomplishes this goal.

3.2 Attack Principle

The attacker strategically aims to target and manipulate the local model of a victim client within the network. Specifically, the attacker is focused on tampering and compromising the victim's model through the injection of a backdoor.

In the context of a backdoor injection attack, the objective is to introduce a trigger into the victim's local model, causing it to exhibit a predefined behavior when presented with inputs containing the trigger. This form of backdoor attack within FL can be conceptualized as a multi-objective optimization problem, as expressed in the following formulation:

$$\theta_a = \arg\min_{\theta \in \mathbb{R}^d} \mathcal{L}_{\mathcal{D}}(f(\theta, x), y) + \mathcal{L}_{\mathcal{D}_p}(f(\theta, x), y), \tag{3}$$

where $\mathcal{D}$ and $\mathcal{D}_p$ denote the clean and poisoned datasets, respectively.

In decentralized FL, it is important to note that the victim client is not obligated to select the attacker's parameters for its model's update. Therefore, a successful attack must go beyond merely tampering with the parameters; it must induce the victim's model to opt for the manipulated parameters during the update. Given that the selection hinges on the performance of the attacker's parameters θ_a on the victim's data $\mathcal{D}_i$, the attacker aims to minimize $\mathcal{L}_{\mathcal{D}_i}(f_i(\theta_a))$. Since the attacker does not have access to $\mathcal{D}_i$, it minimizes $\mathcal{L}_{\mathcal{D}_i}(f_i(\theta_a))$ by relying on the victim's parameters shared during the training.

The most direct approach is to directly copy the victim's parameters, setting $\theta_a = \theta_i$, but such a strategy falls short of achieving the attack objective since it does not compromise the victim. Thus, our proposed attack injects the backdoor

by iteratively updating on top of the victim's parameters in each round:

$$\theta_a^0 = \theta_i^{(t)} \tag{4}$$

$$\theta_a^{(e+1)} = \theta_a^{(e)} - \mu \nabla_{\theta_a^{(e)}} \mathcal{L}_{\mathcal{D}_p}(f(\theta_a^{(e)}, x), y), \tag{5}$$

where $\mathcal{D}_p$ represents the poisoning dataset and e is the local epoch's index, respectively. The pseudocode of our proposed attack is provided in Algorithm 1.

Algorithm 1 Selection Attack via Neighbor Deception

Input: The poisoning dataset $\mathcal{D}_p$, victim's parameters $\theta_i^{(t)}$, the number of local epochs E, and the learning rate μ

$\theta_a = \theta_i^{(t)}$
for epoch 1 to E **do**
  $\theta_a = \theta_a - \mu \nabla_{\theta_a} \mathcal{L}_{\mathcal{D}_p}(f(\theta_a, x), y)$
end for
return θ_a

For illustrative purposes, Algorithm 2 describes the behavior of the victim in one round of training with DAC. Due to the copy step in Equation 4, the similarity s_{ia}^t between the attacker's model and the victim's local model is expected to be high. This results in a high sampling probability p_{ia}^t used in the next round of DAC. This translates to a higher chance that the tampered parameters of our attacker gets selected.

Algorithm 2 Victim in one communication round of DAC under SAND

Given: The prior probability p_i^{t-1}, The number of local epochs E, the number of neighbors M, and the parameters of other clients.

for epoch 1 to E **do**
  $\theta_i = \theta_i - \mu \nabla_{\theta_i} \mathcal{L}_{\mathcal{D}_i}(f(\theta_i, x), y)$
end for
Sample M clients among K clients with probabilities p_i^{t-1}
Compute similarity scores using (Eq. 2):
$s_i^t = [s_{i1}^t, s_{i2}^t, \ldots, s_{iK}^t]$ *// Due to (4), s_{ia}^t should be high*
Update the sampling probability:
$p_{ij}^t = e^{\tau s_i^t} / \left(\sum_{k=1}^K e^{\tau s_{ik}^t} \right)$
$\theta_i \longleftarrow$ FedAvg ($\{\theta_j, j$ is in M sampled clients$\}$)
return θ_i

To effectively realize the attack, we need to design the poisoning dataset $\mathcal{D}_p$, select the attacker local epochs E and fine-tune the learning rate μ. Intuitively, the higher the number of epochs and the learning rate, the more backdoors we inject into the malicious model θ_a. The trade-off is that it would make θ_a diverge

from θ_i, which results in a lower similarity and less chance of getting selected. We will elaborate in more details in the next experimental section.

4 Experimental Analysis

4.1 Experimental Setup

To rigorously assess the efficacy of our proposed attack, we conducted extensive experiments using 2 high-performance NVIDIA A100 and GeForce GPUs paired with 2 CPUs, each equipped with 40GB of RAM. In showcasing the versatility and robustness of our attack strategies, we strategically opted for widely recognized datasets within the realm of image classification. Our evaluation specifically leverages the CIFAR-10 dataset [13], a well-established benchmark for image recognition tasks featuring 10 classes and a total of 60,000 images. Additionally, we included the Fashion-MNIST dataset [14], comprising 10 classes and a diverse set of 70,000 fashion-related images. By applying our methodologies across these distinct datasets, our goal is to not only rigorously test the efficacy but also to showcase the scalability and applicability of our proposed attack strategies in diverse contexts within the realm of image classification.

4.2 Attack Configurations

To embrace the adversarial capabilities within our network, our attack is meticulously configured to empower the adversary. This strategic configuration enables the adversary to dynamically (1) scale up or down its training data, (2) modify its local learning rate, and (3) extend its training duration beyond the standard limit imposed on regular clients similar to an adversary in centralized FL [15]. These modifications easily integrate into the decentralized FL protocols, and their presence does not disrupt the normal behavior of the system; thus, we deem them as acceptable enhancements.

In our comprehensive experiments, we systematically vary the number of clients from 20 to 100, ensuring the presence of a single adversary in each case. Each experiment spans 100 rounds, maintaining a consistent learning rate of 0.0001. Notably, the adversary is granted the flexibility to extend its training duration by 10 additional epochs compared to the regular clients, all while adhering to the protocol. This nuanced approach allows us to thoroughly explore the impact of adversarial manipulations on the system's dynamics, ensuring a robust evaluation of our proposed attack strategies.

4.3 Evaluation Metrics

To comprehensively evaluate the effectiveness of our attack, we frame our assessment around two pivotal questions that form the foundation for our chosen metrics:

1. Relative Selection Frequency? We assess the average selection frequency of a malicious client in comparison to that of an arbitrary client. A higher average selection frequency suggests the successful compromise of the neighbor selection protocol by the malicious client, thereby enhancing its opportunities to propagate manipulated model updates to a victim.
2. Accuracy of Backdoor Model Updates? We measure the accuracy of the backdoor's integration into the victim's model. This metric serves as a key indicator of the attack's success rate, with higher accuracy values obviously signifying more effective incorporation of tampered updates.

Beyond these primary considerations, our analysis extends to the impact of specific hyperparameters, such as the ratio of backdoored data injected and the amount of training data utilized by the attacker.

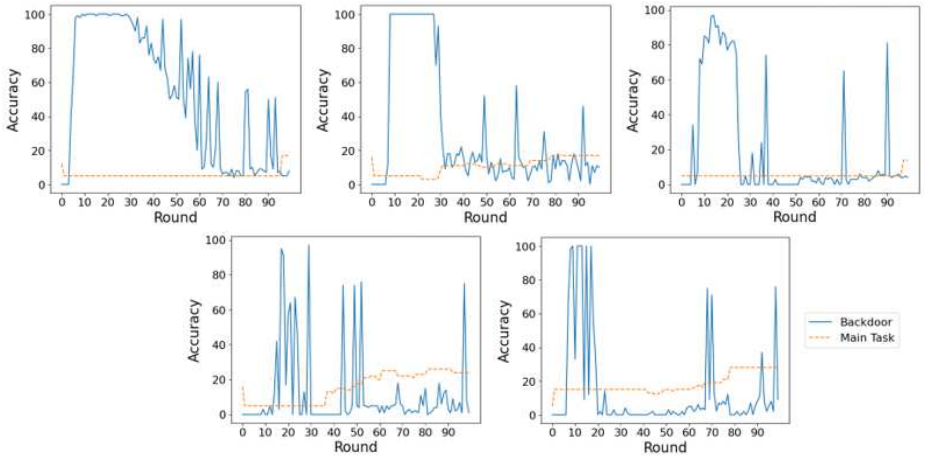
4.4 Experimental Results

In each experiment, we track how often the malicious client is selected as a neighbor compared to an arbitrary client in the network. Preliminary results are presented in Table 1 showing that even with varying numbers of clients, the malicious client is selected as a neighbor more times than an arbitrary client. We would like to emphasize that the malicious client is selected as a neighbor to the targeted victim, but the number of times selected that are presented are for the entire network. Therefore, we show that the malicious client can become a neighbor to a targeted victim *and* get selected by other clients in the network more times than an arbitrary client. As previously stated, the Gossip protocol is completely random so it is expected for the selections to differ each run which is why in some cases an arbitrary client may be selected more times than our malicious client.

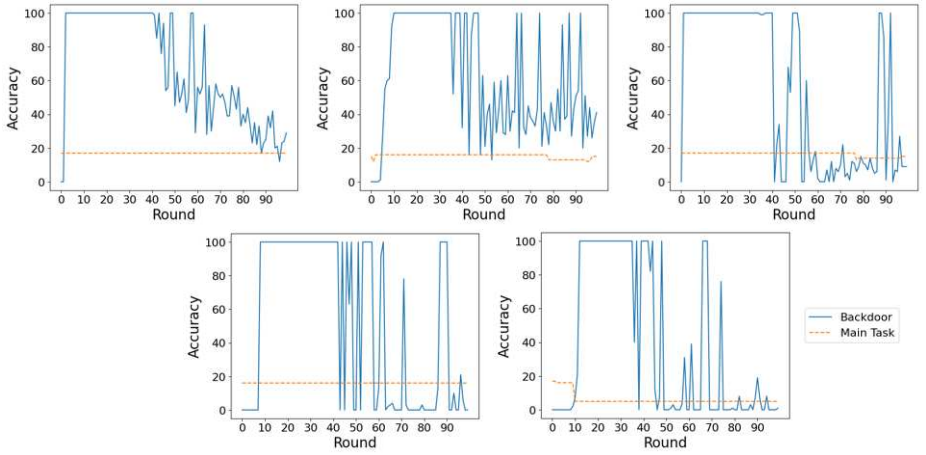
No. of Clients	MS (CIFAR-10)	AS (CIFAR-10)	MS (F-MNIST)	AS (F-MNIST)
20	**D:392**, **G:528**	D:335, G:504	**D:419**, G:493	D:296, **G:507**
40	**D:5,459**, **G:519**	D:4,052, G:471	**D:5,306**, G:493	D:3,929, **G:506**
60	**D:17,581**, **G:535**	D:16,349, G:501	**D:22,094**, **G:478**	D:19,073, G:472
80	**D:46,406**, G:501	D:45,409, **G:515**	**D:48,358**, G:498	D:37,461, **G:515**
100	**D:101,792**, G:486	D:82,816, **G:504**	**D:96,723**, **G:520**	D:93,254, G:508

Table 1: Malicious Selection (MS) vs. Arbitrary Selection (AS) w.r.t CIFAR-10 and FashionMNIST. *Note:* D = *DAC* and G = *Gossip*.

Initial Configurations: Figures 2 and 3 illustrate the effectiveness of our attacker injecting the backdoor into a victims' models through propagated updates, showcasing outcomes under varying numbers of clients and our initial attack configurations detailed in Section 4.2. Experimenting with 20, 40, 60, 80, and 100 clients, employing the DAC and Gossip protocols for CIFAR-10 and FashionMNIST, respectively, provides a comprehensive view of our attack's initial impact.

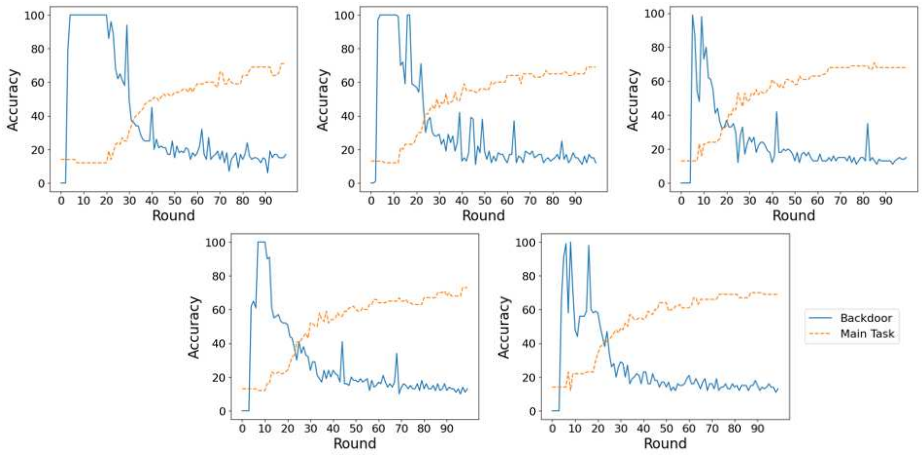


(a) DAC protocol.

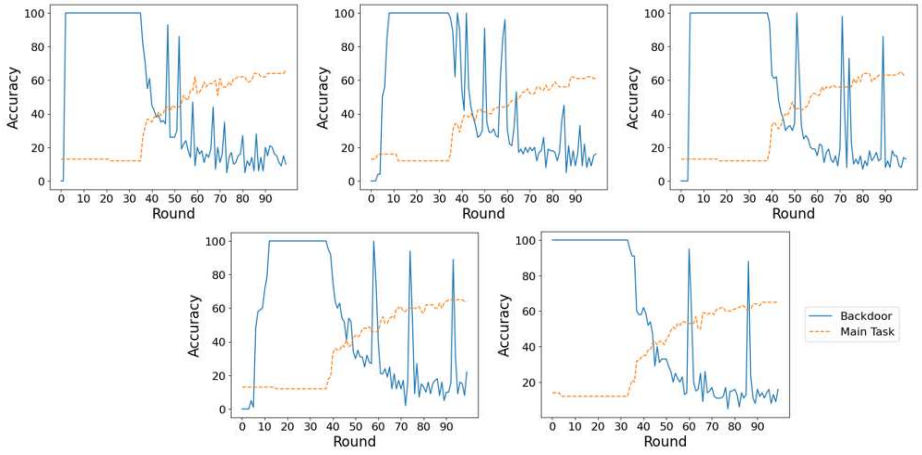


(b) Gossip protocol.

Figure 2: Backdoor Accuracy with Varying Number of Clients on CIFAR-10.



(a) DAC protocol.



(b) Gossip protocol.

Figure 3: Backdoor Accuracy with Varying Number of Clients on FashionMNIST.

For CIFAR-10, our results indicate that the injected backdoor by an attacker achieves testing accuracy exceeding 60%, with instances approaching or reaching 100%. This high misclassification rate underscores the success of the attack. For FashionMNIST, the injected backdoor attains 100% testing accuracy initially. However, as the rounds progress, the main task accuracy gradually rises, eventually surpassing the backdoor accuracy. This suggests that the global model eventually memorizes the main task at a higher rate before converging.

In both cases, the backdoor accuracy exhibits fluctuations over time due to the malicious client's interactions with other neighbors. Despite not being selected by the victim every round, our results demonstrate the backdoor's persistent presence within the network, highlighting the resilience of our attack strategy.

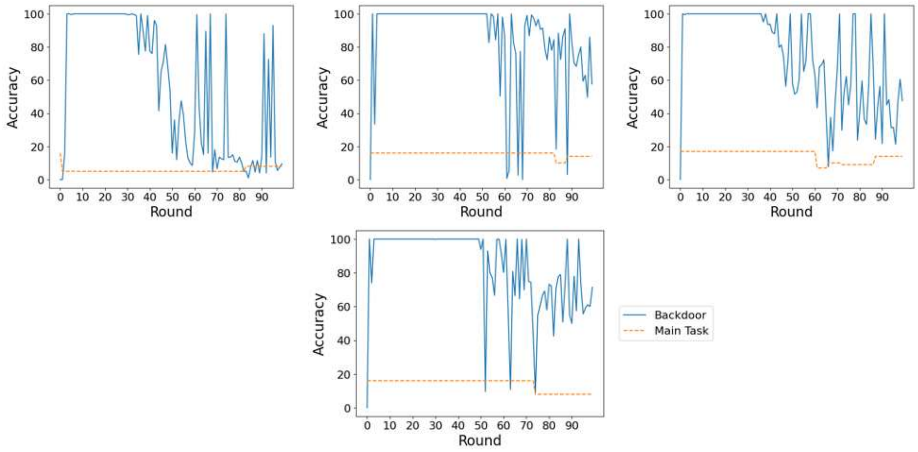
Effects of Training Set Size: Adhering to the decentralized network protocol, our adversary possesses the capability to expand its training data, intensifying the memorization of its backdoor by the victim's model. To evaluate these effects systematically, we incrementally increase the adversary's training set size by 200, 300, 400, and 500 data points, respectively. The outcomes, depicted in Figure 4, within the DAC protocol reveal two key observations:

1. The backdoor attains 100% accuracy at a faster rate with larger training datasets.
2. The backdoor persists in the model with higher accuracy in later rounds than previously before on both CIFAR-10 and FashionMNIST.

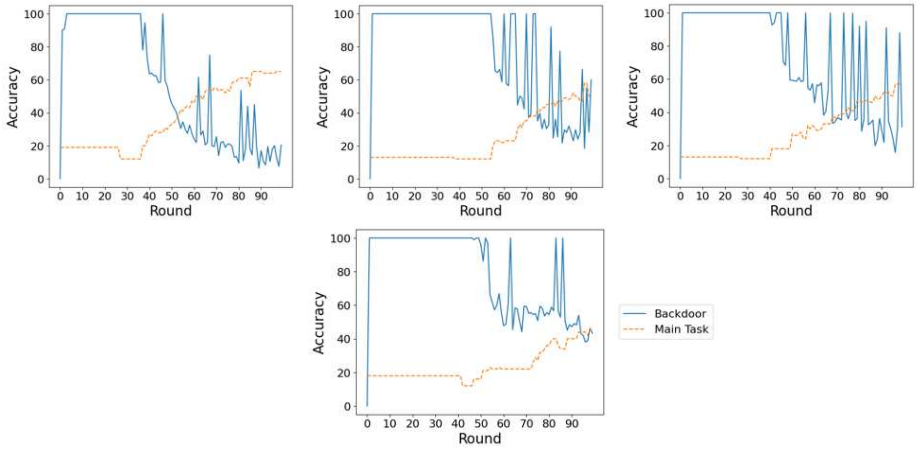
Notably, when expanding the FashionMNIST training dataset by an additional 500 data points, the main task accuracy fails to surpass the backdoor accuracy. This significant enhancement ensures the consistent dominance of our backdoor over the main task, as demonstrated in contrast to Figure 3 with our initial configurations.

Effects of Backdoor Ratio and l_2 Norm: To ensure a comprehensive evaluation of the attack's optimal performance, we utilize 500 additional data points from the previous experiment when investigating the impact of the backdoor ratio. In these experiments, we examine the backdoor accuracy across ratios of 0.1, 0.3, 0.5, 0.7, and 0.9. Modifying the attack further, we leverage the l_2 norm to enforce smaller and more consistent magnitudes in the weights of the attacker. The results, depicted in Figures 5 and 6, showcase the attack's effectiveness within the DAC and Gossip protocols at various injection levels.

In particular, we observe a consistent behavior of the backdoor within the Gossip protocol on the CIFAR-10 and FashionMNIST, where its persistence remains virtually identical across each injection level. Despite fluctuations in backdoor accuracy in later rounds, it consistently dominates the main classification task in the victims' model. Additionally, Figure 6 demonstrates that even with the adversary utilizing only 10% injected backdoor data for training in DAC, the backdoor achieves 100% accuracy and maintains relatively high accuracy throughout the training process compared to the main task on FashionMNIST.



(a) CIFAR-10



(b) FashionMNIST

Figure 4: Effects of Increasing Training Size on Backdoor Accuracy within DAC.

5 Related Works

Decentralized Federated Learning. Despite the prevalent assumption in numerous recent works that their decentralized networks exhibit a ring topology [16–20], we argue that relying solely on the assumption of a specific topology in a decentralized network may lead to suboptimal performance. This oversight arises from neglecting the critical aspect of the neighbor selection process, which forms the basis for the assumed topology. We meticulously consider and account for the behavior of the neighbor selection process, recognizing that the effectiveness of a decentralized system is inherently tied to how neighbors are strategically chosen.

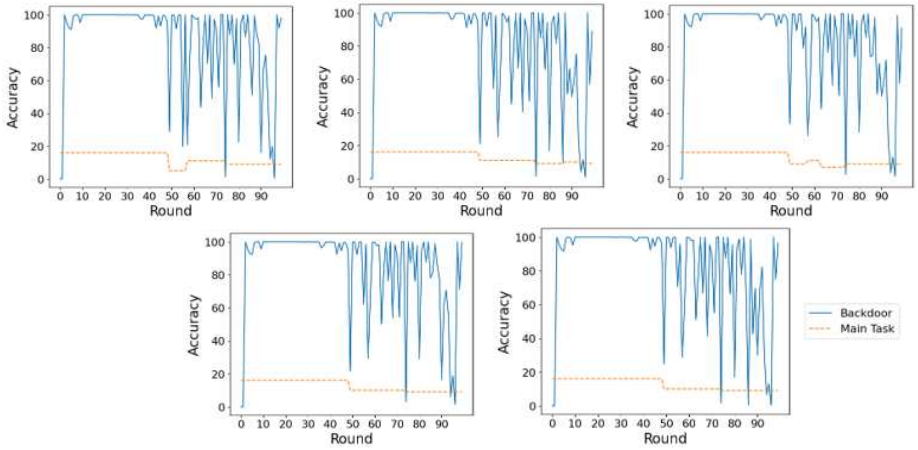
Backdoor Attacks in FL. A multitude of backdoor attacks have been introduced in the context of federated learning (FL), addressing various complexities such as model replacement [15, 21], model poisoning [22], relationships between model parameters and backdoors [23,24], and the utilization of distributed frameworks for backdoor dissemination [25]. Despite their diversity, these backdoor attack variants primarily target the centralized FL setting.

In contrast, our work emphasizes a backdoor attack in the domain of decentralized FL, eliminating the reliance on a central server. While the fundamental concepts of these attacks could be adapted to our decentralized setting, the translation is not necessarily straightforward due to factors such as the neighbor selection process. We have shown that only by precise application of the backdoor can the attacker (1) be selected as a neighbor to the the targeted victim and (2) affect the victims' model update process with the backdoor injected model updates. Another significant departure from recent works is our use of a single attacker instead of many. Even in the presence of a single attacker, our scenario has demonstrated remarkable effectiveness in disrupting the training of a targeted victim, showcasing the robustness and impact of our proposed approach.

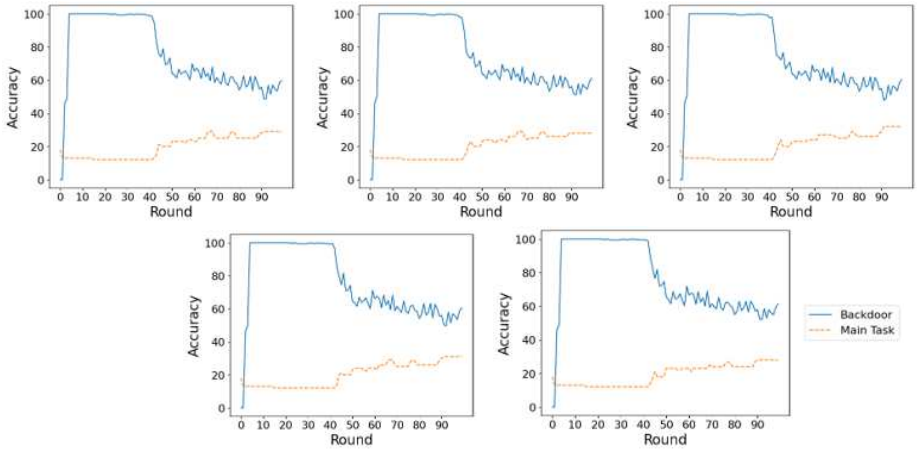
6 Countermeasures Discussion

Current Defenses. Existing defenses in centralized FL predominantly delve into techniques such as clipping and smoothing applied to a global model and its parameters. These methods enable the detection of backdoors, offering certifiable robustness guarantees [26]. Complementary defense strategies in the centralized FL landscape involve a comprehensive blend of techniques, including clipping, smoothing, noise addition, and clustering. These combined approaches aim to not only identify but also eliminate backdoors [27, 28].

Transitioning to a decentralized FL context, the integration of these defenses exposes the distinct challenges arising from the absence of a central server and the decentralized nature of the network. Notably, strategies like clipping and smoothing, which were conventionally applied at the central server, must now be adapted and implemented across each individual client within the decentralized network. Numerous factors come into play in determining the success of



(a) CIFAR-10



(b) FashionMNIST

Figure 5: Effects of Backdoor Ratio on Backdoor Accuracy within Gossip.

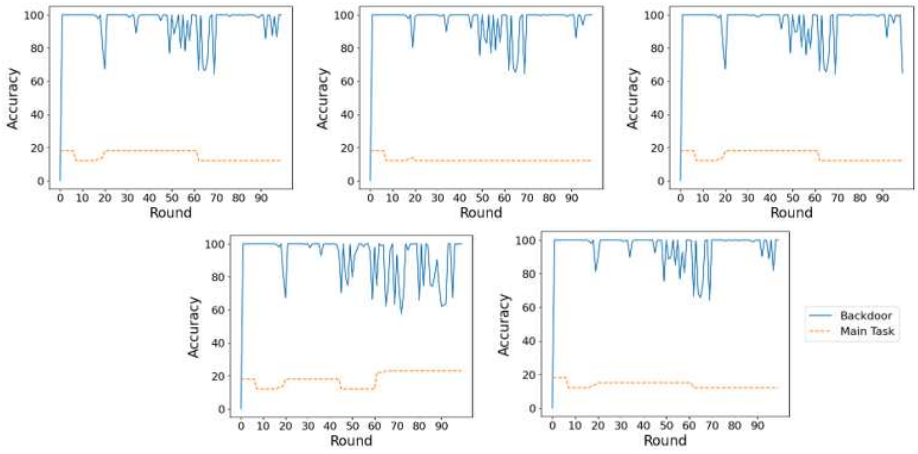


Figure 6: Effects of Backdoor Ratio on Backdoor Accuracy in DAC on Fashion-MNIST.

these defenses within a decentralized environment. Local capabilities of individual clients, the potential impact of defenses on data distributions, and the reciprocal influence of possibly altered distributions on neighbor selections are pivotal considerations. The combination of these factors emphasizes the complexity of translating and optimizing defenses for the decentralized setting. It is evident that a straightforward translation from centralized to decentralized is not feasible, and ensuring the efficacy of these defenses in the decentralized context requires a more thoughtful adaptation process.

Our Proposal. We propose a holistic defense strategy utilizing randomness and network reshuffling. Incorporating randomness into the neighbor selection process disrupts an adversary's ability to guarantee proximity to a targeted victim. A central challenge in implementing this randomness is the need to preserve the similarity between neighboring users while introducing the necessary degree of unpredictability. This method strategically optimizes the neighbor selection process, finding a balance that enhances system performance while preventing adversarial manipulation.

Additionally, reshuffling neighbors at defined epochs counteracts an adversary's potential isolation attacks on a targeted victim. The frequency of reshuffling, denoted by the number of epochs, plays a pivotal role in balancing the impact of adversarial attacks with the stability of the network. Opting for a smaller number of epochs to trigger reshuffling may diminish the effectiveness of adversarial attacks but risks network destabilization due to frequent adjustments. On the other hand, a larger number of epochs may maintain stability, but could inadvertently create more opportunities for successful attacks. Determining the optimal number of epochs to trigger reshuffling, thereby striking a delicate balance between network stability and defense against attacks, stands as a crucial aspect of implementing this robust defense mechanism.

7 Conclusion

In conclusion, our exploration into Decentralized Federated Learning (DFL) has exposed a novel backdoor attack scenario that challenges the conventional assumptions and defenses within this domain. By meticulously examining the vulnerabilities stemming from a malicious client manipulating the neighbor selection process, our study sheds light on the complexities of security challenges in DFL. The Selection Attack via Neighbor Deception (SAND) represents a significant threat, allowing adversaries to compromise both the neighbor selection process and the data distribution of other participants. Our extensive experiments provide compelling evidence of the effectiveness and robustness of the proposed SAND attack. The attacker's ability to strategically become a neighbor to a targeted victim, coupled with the injection of corrupted updates, leads to a domino effect that undermines the victim's model training and, subsequently, the quality of the global model. Furthermore, the comprehensive analysis of SAND's consequences underscores the severity of this attack and the need for robust security measures in DFL. As our work unveils a new dimension of security challenges in FL, it prompts a reevaluation of existing assumptions and defenses, inspiring further exploration into the landscape of DFL security.

Acknowledgement

This work was supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korea government (MOTIE) (RS-2023-00303559, Study on developing cyber-physical attack response system and security management system to maximize real-time distributed resource availability). This material is also supported by the National Science Foundation under grants CNS-1935923 and IIS-2416606.

References

1. T. Nguyen, P. Thai, J. Tre'R, T. N. Dinh, and M. T. Thai, "Blockchain-based secure client selection in federated learning," in *2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 1–9, IEEE, 2022.
2. J. Tre'R and M. T. Thai, "Privacy analysis of federated learning via dishonest servers," in *2023 IEEE 9th Intl Conference on Big Data Security on Cloud (BigDataSecurity)*, pp. 24–29, IEEE, 2023.
3. J. Tre'R, T. Nguyen, R. Alharbi, and M. T. Thai, "Oasis: offsetting active reconstruction attacks in federated learning," in *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1004–1015, IEEE, 2024.
4. M. N. Vu, J. Tre'R, R. Alharbi, and M. T. Thai, "Active data reconstruction attacks in vertical federated learning," in *2023 IEEE International Conference on Big Data (BigData)*, pp. 1374–1379, IEEE, 2023.
5. J. Gao, B. Hou, X. Guo, Z. Liu, Y. Zhang, K. Chen, and J. Li, "Secure aggregation is insecure: Category inference attack on federated learning," *IEEE Transactions on Dependable and Secure Computing*, 2021.

6. J. Li, N. Li, and B. Ribeiro, "Effective passive membership inference attacks in federated learning against overparameterized models," in *The Eleventh International Conference on Learning Representations*, 2022.
7. T. Nguyen, P. Lai, K. Tran, N. Phan, and M. T. Thai, "Active membership inference attack under local differential privacy in federated learning," in *International Conference on Artificial Intelligence and Statistics*, pp. 5714–5730, PMLR, 2023.
8. J. C. Zhao, A. Sharma, A. R. Elkordy, Y. H. Ezzeldin, S. Avestimehr, and S. Bagchi, "Loki: Large-scale data reconstruction attack against federated learning through model manipulation," in *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 30–30, IEEE Computer Society, 2023.
9. J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16937–16947, 2020.
10. L. Yuan, Z. Wang, L. Sun, S. Y. Philip, and C. G. Brinton, "Decentralized federated learning: A survey and perspective," *IEEE Internet of Things Journal*, 2024.
11. E. L. Zec, E. Ekblom, M. Willbo, O. Mogren, and S. Girdzijauskas, "Decentralized adaptive clustering of deep nets is beneficial for client collaboration," in *International Workshop on Trustworthy Federated Learning*, pp. 59–71, 2022.
12. Z. Tang, S. Shi, B. Li, and X. Chu, "Gossipfl: A decentralized federated learning framework with sparsified and adaptive communication," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 3, pp. 909–922, 2022.
13. A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.
14. H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.
15. E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International conference on artificial intelligence and statistics*, pp. 2938–2948, PMLR, 2020.
16. X. Lian, W. Zhang, C. Zhang, and J. Liu, "Asynchronous decentralized parallel stochastic gradient descent," in *International Conference on Machine Learning*, pp. 3043–3052, PMLR, 2018.
17. X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," *Advances in neural information processing systems*, vol. 30, 2017.
18. Y. Shi, L. Shen, K. Wei, Y. Sun, B. Yuan, X. Wang, and D. Tao, "Improving the model consistency of decentralized federated learning," in *International Conference on Machine Learning*, pp. 31269–31291, PMLR, 2023.
19. T. Vogels, L. He, A. Koloskova, S. P. Karimireddy, T. Lin, S. U. Stich, and M. Jaggi, "Relaysum for decentralized deep learning on heterogeneous data," *Advances in Neural Information Processing Systems*, vol. 34, pp. 28004–28015, 2021.
20. M. Bornstein, T. Rabbani, E. Z. Wang, A. Bedi, and F. Huang, "Swift: Rapid decentralized federated learning via wait-free model communication," in *The Eleventh International Conference on Learning Representations*, 2022.
21. 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. 16070–16084, 2020.

22. X. Lyu, Y. Han, W. Wang, J. Liu, B. Wang, J. Liu, and X. Zhang, "Poisoning with cerberus: stealthy and colluded backdoor attack against federated learning," in *Thirty-Seventh AAAI Conference on Artificial Intelligence*, 2023.
23. Y. Dai and S. Li, "Chameleon: Adapting to peer images for planting durable backdoors in federated learning," in *International Conference on Machine Learning*, pp. 6712–6725, PMLR, 2023.
24. Z. Zhang, A. Panda, L. Song, Y. Yang, M. Mahoney, P. Mittal, R. Kannan, and J. Gonzalez, "Neurotoxin: Durable backdoors in federated learning," in *International Conference on Machine Learning*, pp. 26429–26446, PMLR, 2022.
25. C. Xie, K. Huang, P. Y. Chen, and B. Li, "Dba: Distributed backdoor attacks against federated learning," in *8th International Conference on Learning Representations, ICLR 2020*, 2020.
26. C. Xie, M. Chen, P.-Y. Chen, and B. Li, "Crfl: Certifiably robust federated learning against backdoor attacks," in *International Conference on Machine Learning*, pp. 11372–11382, PMLR, 2021.
27. P. Rieger, T. D. Nguyen, M. Miettinen, and A.-R. Sadeghi, "Deepsight: Mitigating backdoor attacks in federated learning through deep model inspection," in *Network and Distributed Systems Security*, 2022.
28. T. D. Nguyen, P. Rieger, R. De Viti, H. Chen, B. B. Brandenburg, H. Yalame, H. Möllering, H. Fereidooni, S. Marchal, M. Miettinen, *et al.*, "{FLAME}: Taming backdoors in federated learning," in *31st USENIX Security Symposium (USENIX Security 22)*, pp. 1415–1432, 2022.