Less is More: Persistent Low-Frequency Backdoor Injection in Federated Learning

Pei Ye*, Yuqing Li*, Kun He*, Haoran Wang*, Ruiying Du*, Wei Wang†
* Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education,
School of Cyber Science and Engineering, Wuhan University, Wuhan, China
† Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong
*{yepeiii, li.yuqing, hekun, whr2023, duraying}@whu.edu.cn, †weiwa@cse.ust.hk

***Abstract*—Federated learning (FL) enables multiple clients to collaboratively train a machine learning model without sharing their local data. However, the distributed nature of FL makes it vulnerable to backdoor attacks from malicious clients. Most existing attack methods often assume that attackers can inject backdoors in every training round—a scenario that is both *unrealistic* and *inefficient* in real-world FL deployment. In this paper, we investigate why backdoor attacks become less effective under *low-frequency injection* and propose a novel attack paradigm for FL, called REinforced Memorization-based INterval backDoor attack (REMIND). REMIND optimizes the backdoor trigger via *task alignment* and *feature alignment*. Task alignment aligns backdoor and main task objectives to resist benign update suppression during non-attack rounds, while feature alignment guides poisoned samples to match the activation trajectory of target-class samples. This dual alignment enhances the backdoor's persistence and narrows the divergence between malicious and benign updates. With strong attack success rates established, we further analyze the advantages of low-frequency backdoor attacks, particularly their ability to improve robustness against defense mechanisms. Extensive evaluations on four benchmark datasets show that REMIND consistently outperforms eight state-of-the-art attack baselines under nine defense strategies.**

I. Introduction

In recent years, federated learning (FL) has achieved remarkable success as a collaborative paradigm in which multiple clients, such as smartphones or IoT devices, jointly train a shared model while keeping their data decentralized and private [1]–[5]. FL has enabled compelling results across diverse applications, including financial fraud detection [6], medical research [7], and green IIoT system [8]. Despite its significant potential, many studies [9]–[12] reveal that the distributed nature of FL provides a novel attack surface to backdoors. The independence of client-side training allows adversaries to compromise a subset of clients and inject malicious updates containing a backdoor trigger. The global model is hence manipulated and forced to classify any input patched with a trigger as a specific target label, all while retaining standard accuracy on benign samples. Backdoor attacks pose significant challenges to the reliability and security of FL applications [13], [14].

To date, much of the literature has explored backdoor attacks in FL under the implicit—and often unrealistic—assumption that the attackers can inject malicious updates in every training round. Prior works under this idealized condition primarily seek to improve the attack effectiveness [9], [15], [16] and stealthiness [10], [17], [18] through sophisticated techniques. However, a critical yet often overlooked issue is the *frequency* of attack injection, i.e., how often attackers can actually compromise training. Intuitively, reducing injection frequency may weaken attack effectiveness, but it also substantially reduces the attacker's risk of being detected by existing defense mechanisms. This dynamic introduces a compelling and underexplored *trade-off* between attack strength and stealth. Moreover, low-frequency injection scenarios are more representative of practical FL deployments, where malicious clients often face real-world limitations such as intermittent connectivity, limited availability, or participation restrictions [19], [20]. These constraints mean that adversaries cannot consistently submit poisoned updates in every round. It is hence imperative to examine backdoor attacks under more realistic and constrained injection frequencies.

When adapting existing backdoor attack techniques to these low-frequency injection settings, we observe *significant performance degradation*, evidenced by slower convergence, reduced attack success rates (ASR), and overall weakening attack effects (see Section III-D for empirical results). This performance drop primarily stems from the fact that backdoors injected during attack rounds are easily *overwritten* or *diluted* by the accumulation of benign client updates in subsequent non-attack rounds. As a consequence, maintaining persistent backdoor effects over time becomes increasingly difficult. These findings underscore the urgent need for new techniques capable of preserving both effectiveness and stealth for backdoor attacks in FL, even under limited injection opportunities.

In this paper, we present REMIND, a REinforced Memorization-based INterval backDoor attack framework tailored to the realities of low-frequency injection in practical FL deployments. Unlike prior methods that suffer from backdoor signal dilution, REMIND employs a dual alignment mechanism, which aligns poisoned and target-class samples

This research was supported in part by the National Key R&D Program of China under grant No. 2022YFB3102100, the National Natural Science Foundation of China under grant 62302343 and grant 62441237, the RGC GRF grant under the contract 16211123, and the RGC RIF grant under the contract R6021-20. (Corresponding author: Yuqing Li.)

across the model's decision boundary and feature space. This tactic not only aligns the objective of the backdoor task with the main task to prevent the backdoor from being diluted by benign updates, but also narrows the divergence between malicious and benign updates to further enhance the stealthiness of the backdoor attack. Beyond achieving strong attack effectiveness, REMIND further examines the advantages of low-frequency backdoor attacks, revealing their potential to improve attack efficiency through fewer attack rounds and enhance robustness against existing defense mechanisms. Specifically, our approach reduces the detectability of anomaly-detection-based countermeasures while simultaneously leading to substantial wasted computational overhead for defenders during non-attack rounds.

We conduct extensive experiments on widely-used benchmark datasets, including CIFAR-10 [21], CIFAR-100 [21], CINIC-10 [22] and Tiny-ImageNet [23], to evaluate the effectiveness of REMIND. We further compare it with eight state-of-the-art attack baselines under nine defense methods. The results suggest that REMIND outperforms all attack baselines, achieving a significant improvement in the speed of backdoor injection across all defenses. In addition, we examine the contributions of REMIND's components, validating the improved robustness of low-frequency injection against various defenses.

We summarize our main contributions as follows:

- We present the first systematic study of low-frequency backdoor attacks in FL, identifying a critical challenge posed by rapid backdoor overwriting by benign updates during non-attack rounds. Building on this insight, we propose REMIND, a novel reinforced memorization-based low-frequency attack framework that optimizes the backdoor trigger using task and feature alignment to enhance its effectiveness and persistence over time.
- Beyond demonstrating the effectiveness of REMIND under limited injection frequencies, we further reveal that low-frequency attacks can significantly undermine the *reliability* and *efficacy* of various defense mechanisms by lowering their detection rates and incurring unnecessary computational overhead.
- Extensive experiments on real-world datasets validate that REMIND remarkably improves the effectiveness and persistence of low-frequency injected backdoors compared to the state-of-the-art attack baselines. These findings expose new vulnerabilities in FL security and offer valuable perspectives for future research on developing resilient defenses.

II. Related Work

A. FL Backdoor Attacks

FL is susceptible to backdoor attacks due to its decentralized nature and lack of control over clients' local training processes [9], [10], [16], [18], [24], [25]. In these attacks, the attacker compromises a subset of participating clients and manipulates their local datasets to upload malicious updates, thereby injecting the backdoor into the global model. The proportion of compromised clients among all participants and the fraction of poisoned samples within a local dataset are referred to as the poisoned model rate (PMR) and poisoned data rate (PDR), respectively.

Existing FL backdoor attacks follow this basic backdoor injection paradigm and introduce various optimizations to enhance attack effectiveness or stealthiness. Model replacement (MR) [9] amplifies malicious local updates to dominate the aggregation process so as to maximize its impact on the global model. In contrast, LIE [17] prioritizes stealthiness by crafting malicious updates that closely resemble benign ones, achieving a balance between attack effectiveness and evasion of detection. Several works focus on improving trigger stealthiness through input manipulations. DBA [18] divides a complete trigger into multiple small patterns and distributes them across different clients, while FCBA [16] further explores combinatorial variations of these patterns to increase diversity and backdoor generalization. Alternative methods adopt parameter-level manipulations for attack stealthiness. 3DFed [10] perturbs malicious updates with noise, uploads decoy updates to confuse defenses, and embeds feedback-based indicators into parameters to adapt to the attack strategy. Neurotoxin [15] selectively modifies less critical model parameters for the main task, preserving benign performance while increasing backdoor persistence. Similarly, F3BA [24] flips task-irrelevant weights to maximize layer-wise activation shifts, achieving both stealth and strong attack performance.

However, these methods primarily focus on optimizing the trigger design or manipulating the *spatial patterns* of local updates, without accounting for the *temporal dimension* of the attack. Accordingly, their effectiveness deteriorates when the attack cannot be launched in every round.

B. FL Backdoor Defenses

In this work, we consider three types of backdoor defenses in FL, categorized according to the stage at which the defense is applied [26]–[28]. The first type is pre-aggregation anomaly detection, which aims to identify and exclude malicious updates prior to aggregation. The second type is byzantine-robust aggregation that incorporates robust aggregation strategies to mitigate the impact of abnormal updates during aggregation. The third type is post-aggregation model refinement, seeking to remove backdoor effects from the global model after aggregation. We next introduce the underlying idea of each defense category along with representative methods.

Pre-aggregation anomaly detection. This type of defense mechanisms often assume that malicious and benign updates are distinguishable. Typically, they begin by computing certain metrics across clients, and then apply clustering algorithms or top-K selection strategies to detect anomalous ones. FLAME [29] computes the pairwise cosine similarities among local updates and employs HDBSCAN clustering algorithm [30] to identify outliers. Building upon this idea, FreqFed [31] transforms local updates into the frequency domain, as backdoor updates often exhibit distinct frequency patterns. Moreover, FLDetector [32] predicts the local update following the Cauchy median theorem [33] and deems those that deviate significantly from the prediction as malicious.

Byzantine-robust aggregation. This category of defense methods typically filters or weights the uploaded updates during aggregation to suppress the influence of a small number of abnormal updates. Krum [34] measures the Euclidean distance between each local update and its closest neighbors, and then chooses the one with the smallest total distance for aggregation. Based on this, Bulyan [35] combines multiple Krum-selected updates and further refines them using robust statistics to enhance its defense against adversarial outliers. To mitigate outliers at a finer granularity, Trimmed Mean [36] discards a fixed number of the largest and smallest values in each dimension before averaging, while Median [36] simply selects the median value across clients for each parameter.

Post-aggregation model refinement. These defense methods focus on adjusting the globally aggregated model to erase the backdoor without compromising main-task accuracy. One representative approach is differential privacy (DP) [37], which injects Gaussian noise into the global model to suppress potential backdoor behaviors. Alternative methods leverages knowledge distillation techniques. In FedDF [38], the aggregated model acts as a student that learns from the output logits of local models on clean data to overwrite malicious patterns. FedRAD [39] extends this idea by modifying the model aggregation strategy. It assigns weights to local updates based on the frequency with which a client's output becomes the median, thereby reducing the influence of abnormal clients. In a different direction, FedPruning [40] directly removes backdoor neurons by pruning those with low activation on benign samples, aiming to minimize backdoor capacity while preserving accuracy on the main task.

III. Preliminaries and Motivation

A. System Model

Consider a standard FL system with a central server coordinating N clients to collaboratively train a machine learning model without sharing their raw data. Each client i holds a local dataset $\mathcal{D}_i$. In every training round t, the server randomly selects a subset of clients $\mathcal{C}^t$ and distributes the current global model G^t to them. Each selected client i initializes its local model with G^t, performs E epochs of local training on $\mathcal{D}_i$, and then uploads the resulting model update Δ_i^t to the server. After collecting these local updates, the server aggregates them to produce a new global model $G^{t+1} = G^t + \frac{1}{|\mathcal{C}^t|}\sum_{i\in\mathcal{C}^t}\Delta_i^t$. If any client's local update Δ_i^t is malicious, the backdoor can be injected into the global model G^{t+1}.

B. Threat Model

Attacker's Capability. Following threat models in prior studies [12], [25], [41], we assume that the attacker can compromise a subset of participating clients, having full control over their local training processes. For clarity, we refer to the set of controlled clients as $\mathcal{C}_{mal}$. However, the attacker can neither control the server nor interfere with the aggregation process. Moreover, the attacker is unaware of the defense mechanisms implemented by the server.

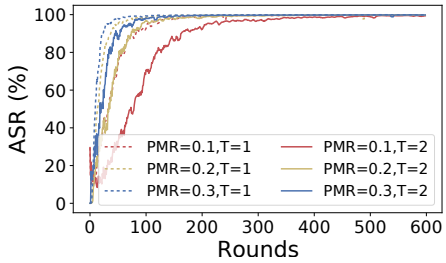


Fig. 1: The ASR (%) of Badnet on CIFAR-10 with varying PMRs and attack intervals.

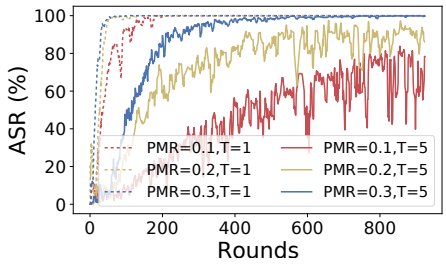


Fig. 2: The ASR (%) of F3BA on CINIC-10 with varying PMRs and larger attack intervals.

Attacker's Goal. We consider an attacker who aims to inject a backdoor into the global model while achieving the following two critical goals. 1) **Effectiveness**: The attacker must ensure that the compromised global model consistently classifies triggered samples as the attacker-specified label, even after any defensive processing. 2) **Stealthiness**: The backdoor injection process should be stealthy enough to keep classification accuracy on clean samples while simultaneously preventing the server from detecting malicious updates submitted by controlled clients. This twofold objectives render the backdoor attack to be both functionally potent and operationally concealed within FL frameworks.

C. Motivation

We now elaborate on the motivations to explore FL backdoor attacks under limited injection frequency.

Motivation 1: Low-frequency backdoor attacks are more accordant with realistic FL scenarios. In practice, FL clients often face challenges such as intermittent connectivity, dynamic availability or limited participation, preventing them from consistently uploading local updates [19], [42]. These practical constraints naturally apply to malicious clients, restricting their ability to inject poisoned updates in every round. Therefore, low-frequency backdoor attacks represent a more realistic threat model and warrant dedicated investigation.

Motivation 2: Low-frequency injection is more efficient for backdoor attacks. Existing FL backdoor attacks [10], [15], [17], [18], [25] typically inject malicious updates in every training round. However, the backdoor task is generally much simpler than the main task, which allows the adversary to gradually implant the backdoor into the global model *over time*, even with low attack frequencies. To illustrate, we implement Badnet attack [43] on CIFAR-10 dataset [21] and report the

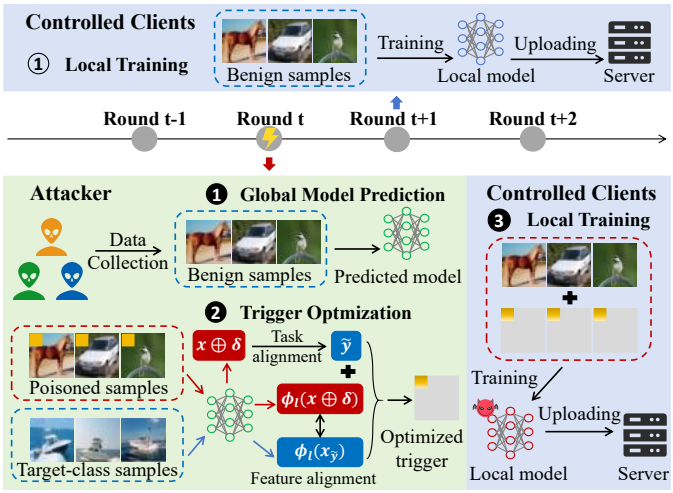


Fig. 3: An overview of REMIND architecture.

ASR while varying PMRs and attack intervals denoted by T (i.e., malicious updates are injected every T rounds). From Fig. 1, when the attacker launches an attack every other round (i.e., $T = 2$), Badnet achieves an ASR comparable to that of attacking every round (i.e., $T = 1$), substantially cutting the overall attack cost. This finding suggests that frequent attack injection may not be necessary; rather, low-frequency attacks provide a more efficient attack paradigm in FL.

D. Challenge

Although low-frequency attacks offer a more realistic and efficient attack solution, their performance may degrade significantly when applied with larger attack intervals or on more complex datasets. This is primarily attributed to the reduced opportunities for injecting malicious updates and the increased difficulty of executing the backdoor task under more challenging datasets. To underscore this point, we refer to Fig. 2 using F3BA attack [24] on CINIC-10 dataset [22]. Compared to the case where $T = 1$, F3BA under a low-frequency setting ($T = 5$) either fails to achieve a comparable ASR or exhibits much slower convergence. These results indicate that existing attack methods struggle to maintain stable performance under low-frequency injection scenarios, limiting the practical benefits of low-frequency attacks. Therefore, there is an urgent need to develop a novel attack capable of maintaining robust effectiveness despite low attack frequencies.

IV. Methodology

To address these challenges, we propose REMIND, a persistent low-frequency backdoor attack framework. We begin with a system overview, and elaborate on how it reinforces backdoor memorization with adverse effects on defenses.

A. Overview

Fig. 3 depicts the main architecture of REMIND, which proceeds in rounds of model synchronization. It operates under two cases depending on whether an attack is launched in the current round. In each **non-attack round**, the attacker takes no action, while each controlled client selected by the server, performs normal training and uploads its benign model updates for aggregation (①). In each **attack round**, the attack process consists of two phases. 1) *Trigger optimization phase*: the attacker first predicts the next-round global model using the combined datasets of all controlled clients (❶), and then optimizes the trigger by sequentially aligning the poisoned and target-class samples both in the feature space and at the decision boundary of the predicted model. (❷). 2) *Poisoned local training phase*: each selected controlled client poisons its local data using the optimized trigger, performs local training on this poisoned data (❸), and finally uploads its malicious updates to the server.

B. REMIND Design

In FL, backdoor attacks typically inject triggers into local training data and upload the poisoned updates to compromise the global model [9], [10], [15]–[18], [24]. However, under low-frequency injection scenarios, existing methods become less effective as the implanted backdoor is gradually diluted or erased by benign updates. To address this challenge, REMIND employs a reinforced memorization loss that adaptively optimizes the trigger on a predicted global model. This enables the backdoor to persist across rounds and allows the global model to effectively *"remind"* itself of the backdoor behavior, even when the attack is not continuously applied.

Local training process. In each round t, the server randomly selects a subset of clients $\mathcal{C}^t$ as participants and broadcasts them the global model G^t for E epochs of local training.

If no attack is launched in the current round, the local training on each selected controlled client $i \in \mathcal{C}_{mal} \cap \mathcal{C}^t$ proceeds as normal, following the standard objective:

$$w_i^{*,t} = \underset{w_i^t}{\operatorname{argmin}} \, \mathbb{E}_{(x,y)\sim\mathcal{D}_i}[\mathcal{L}(x, y; w_i^t)], \tag{1}$$

where w_i^t denotes its local model initialized by global model G^t. The local dataset $\mathcal{D}_i = \{(x, y)\}$ consists of data samples x and their corresponding labels y. $\mathcal{L}(\cdot)$ represents the standard prediction loss function (e.g., cross-entropy), which measures the discrepancy between the model's output and the label.

If an attack is launched in this round, the attack process can be formulated as follows:

$$\begin{aligned} w_i^{*,t} = \underset{w_i^t}{\operatorname{argmin}} \, & \mathbb{E}_{(x,y)\sim\mathcal{D}_{i,mal}}[\mathcal{L}(x \oplus \delta^{*,t}, \tilde{y}; w_i^t)] + \\ & \mathbb{E}_{(x,y)\sim\mathcal{D}_{i,nor}}[\mathcal{L}(x, y; w_i^t)] \\ s.t.\ \delta^{*,t} = \underset{\delta}{\operatorname{argmin}} \, & \mathbb{E}_{(x,y)\sim\mathcal{D}_{att}} RL_\delta(x, y; G^t), \end{aligned} \tag{2}$$

where $RL_\delta(\cdot)$ is the loss function adopted by the attacker to optimize the trigger δ. To achieve a strong attack effect, the attacker collects the local data of all controlled clients for trigger optimization, i.e., $\mathcal{D}_{att} = \cup_{i\in\mathcal{C}_{mal}} \mathcal{D}_i$. For each selected controlled client $i \in \mathcal{C}_{mal} \cap \mathcal{C}^t$, the poisoned local training jointly optimizes two terms in Eq. (2). The first term corresponds to the backdoor loss on malicious samples $\mathcal{D}_{i,mal}$, while the second term ensures that the model retains utility on benign samples $\mathcal{D}_{i,nor}$. Furthermore, the malicious and benign samples satisfy $\mathcal{D}_{i,mal} \cup \mathcal{D}_{i,nor} = \mathcal{D}_i$ and $\mathcal{D}_{i,mal} \cap \mathcal{D}_{i,nor} = \varnothing$.

Reinforced memorization loss. During non-attack rounds, backdoors in the global model tend to be weakened due to the dilution of benign updates. To address this, REMIND adaptively optimizes the trigger before launching attack, by mapping poisoned samples closer to the target label $\tilde{y}$ in current global model's decision boundary. This achieves *task alignment*, where we align the objective of backdoor task with that of main task. This way, even in non-attack rounds, backdoors are not be suppressed by benign updates, thereby maintaining the presence in the global model. Specifically, $RL_\delta(\cdot)$ is defined as:

$$RL_\delta(x, y; G^t) = \mathcal{L}(x \oplus \delta, \tilde{y}; G^t). \tag{3}$$

To further enhance the stealthiness, REMIND performs layer-wise *feature alignment* sequentially during trigger optimization. Accordingly, $RL_\delta(\cdot)$ is extended to:

$$\begin{aligned} RL_\delta(x, y; G^t) &= \mathcal{L}(x \oplus \delta, \tilde{y}; G^t) \\ &+ \sum_{l=1}^{L} \lambda_l \|\phi_l(x \oplus \delta; G^t) - \phi_l(x_{\tilde{y}}; G^t)\|^2. \end{aligned} \tag{4}$$

Here, $\phi_l(x \oplus \delta; G^t)$ denotes the poisoned input's intermediate feature at the l-th layer, and it is aligned with the average feature of the corresponding layer from target-class samples, i.e., $\phi_l(x_{\tilde{y}}; G^t) = \frac{1}{|\mathcal{D}_{\tilde{y}}|} \sum_{(x,y)\in\mathcal{D}_{\tilde{y}}} \phi_l(x; G^t)$, where $\mathcal{D}_{\tilde{y}} = \{(x, y) \in \mathcal{D}_{att} | y = \tilde{y}\}$. The hyperparameter λ_l balances the effect of the two alignments. By aligning layer-wise features, poisoned samples are guided to match the activation trajectory of target-class samples, which in turn narrows the discrepancy between the malicious and benign updates.

Global model prediction module. Building on such reinforced memorization strategy, REMIND further develops a global model prediction module to maintain the association between poisoned samples and the target label during non-attack rounds. The core idea is that under low-frequency injection scenarios, the effectiveness of the attack heavily depends on how well the trigger *generalizes* to future global models aggregated from benign updates. Hence, REMIND estimates the next-round global model in advance and optimizes the trigger on this predicted model. To be more precise, the attacker predicts the global model as follows:

$$\hat{G}^{t+1} = \underset{G^t}{\operatorname{argmin}}\, \mathbb{E}_{(x,y)\sim\mathcal{D}_{att}} \mathcal{L}(x, y; G^t). \tag{5}$$

The algorithm of REMIND. The details of REMIND are illustrated in Algorithm 1. In each training round, the server selects the participating clients and sends them the global model (line 2). If this is an attack round, the attacker first predicts the next-round global model, then optimizes the trigger and distributes it to the controlled clients selected for this round (line 4-6). Subsequently, each selected controlled client injects the optimized trigger into its local dataset and conducts poisoned local training (lines 10-11). After that, the malicious model update is generated and uploaded to the server (lines 14-15). If this is a non-attack round, the controlled clients perform normal local training as other benign selected clients and send the benign updates to the server (lines 18-21). Finally, the server aggregates these received local updates to build the new global model (lines 22-23).

Algorithm 1: The workflow of REMIND

Input: Controlled clients $\mathcal{C}_{mal}$, client i's dataset $\mathcal{D}_i$, training rounds $\mathcal{T}$, attack rounds $\mathcal{T}_{att}$, global model G.
Output: Backdoored global model $G^{|\mathcal{T}|}$.

1 **for** *each* $t \in \mathcal{T}$ **do**
  // Server
2 Select participating clients $\mathcal{C}^t$ and send the current global model G^t to them;
3 **if** $t \in \mathcal{T}_{att}$ **then**
    // Attacker
4 Predict the next round global model $\hat{G}^{t+1}$ by Eq. (5);
5 Optimize the trigger $\delta^{*,t}$ by Eq. (2) and Eq. (4);
6 Distribute $\delta^{*,t}$ to the selected controlled clients $i \in \mathcal{C}_{mal} \cap \mathcal{C}^t$;
    // Selected clients
7 **for** *each client* $i \in \mathcal{C}^t$ **do**
8 $w_i^t \leftarrow G^t$;
9 **if** $i \in \mathcal{C}_{mal}$ **then**
      // controlled clients
10 Poison $\mathcal{D}_i$ with the received trigger $\delta^{*,t}$;
11 Update local model $w_i^{*,t}$ by Eq. (2);
12 **else**
13 Update local model $w_i^{*,t}$ by Eq. (1);
14 $\Delta_i^t \leftarrow w_i^{*,t} - w_i^t$;
15 Upload the update Δ_i^t to the server;
16 **else**
    // Selected clients
17 **for** *each client* $i \in \mathcal{C}^t$ **do**
18 $w_i^t \leftarrow G^t$;
19 Update local model $w_i^{*,t}$ by Eq. (1);
20 $\Delta_i^t \leftarrow w_i^{*,t} - w_i^t$;
21 Upload the benign update Δ_i^t to the server;
    // Server
22 Receive local updates Δ_i^t from every client $i \in \mathcal{C}^t$;
23 $G^{t+1} \leftarrow G^t + \frac{1}{|\mathcal{C}^t|} \sum_{i\in\mathcal{C}^t} \Delta_i^t$;

C. Adverse Effects of Low-frequency Attack on Defense

As discussed in Section II-B, backdoor defense mechanisms in FL can be deployed before, during, or after the aggregation process in each training round. However, when backdoor attacks are launched intermittently rather than in every round, applying these defenses *indiscriminately* across all rounds may suffer from unintended adverse effects. Once REMIND maintains high attack effectiveness, we next conduct a detailed analysis of how different categories of defense mechanisms interact under such low-frequency injection scenarios.

1) Pre-Aggregation Anomaly Detection: This category of defenses focuses on detecting malicious updates before each round of aggregation [29]–[32], with detection effectiveness quantified by detection probability and false alarm rate.

Reduced detection probability. Low-frequency injection reduces the number of attack rounds, thereby decreasing the risk of detection. Formally, we assume that the probability of successfully detecting malicious updates in a round is p and the attacker launches an attack every T rounds. Over a total

of M rounds, the expected number of attack rounds is $\lfloor \frac{M}{T} \rfloor$. Given the independence across rounds, the overall probability of successful detection is $P_{detect} = 1 - (1-p)^{\lfloor \frac{M}{T} \rfloor}$. This suggests that P_{detect} decreases monotonically as T increases. In other words, by reducing the attack frequency, REMIND diminishes the likelihood of malicious clients being detected by defense mechanisms.

Increased false alarm rate. To eliminate malicious updates as thoroughly as possible, detection mechanisms often exclude all updates that deviate from the center of the majority updates μ beyond a threshold τ. A false alarm occurs when the benign update Δ_b satisfies $\|\Delta_b - \mu\| > \tau$. In attack rounds, the presence of a few significantly different malicious updates allows μ to be located among benign updates. However, during non-attack rounds, the differences among benign updates increase due to variations in local data distributions. As a result, the distance between a benign update Δ_b and the center μ may exceed the threshold τ, leading to a higher false alarm rate. This exacerbates the *fairness concerns* in FL [44]–[46], as benign clients with highly skewed data distributions or limited local data are more likely to be misclassified as malicious, and thus unfairly excluded from training participation.

2) Byzantine-Robust Aggregation: This type of defense methods tends to prioritize updates that conform to the majority, and disregard valid yet divergent contributions during aggregation [34]–[36]. This *conservative* strategy can impair the performance of the global model on the main task.

Decreased main task accuracy. Since backdoor attacks aim to preserve the model performance on the main task, such defenses respond indiscriminately across both attack and non-attack rounds. While eliminating backdoor information, they inadvertently exclude outlier benign updates, leading to a large drop in main task accuracy. In this sense, low-frequency attacks allow attackers to induce the same level of disruption with significantly lower attack costs.

3) Post-Aggregation Model Refinement: These defenses employ *computationally intensive* techniques, like knowledge distillation [38] and neuron pruning [40], to refine the potentially poisoned global model after each round of aggregation.

Wasted resources. Due to the inability to distinguish between attack and non-attack rounds, this type of defenses only indiscriminately applies model refinement in every round. As a result, substantial resources are unnecessarily consumed by the defender during non-attack rounds, leading to huge computation overhead.

V. Experiments

A. Experimental Setup

Implementation details. We conduct experiments using Pytorch 2.2.1 and CUDA 12.2 on a GPU server equipped with Intel Xeon Gold 6133 CPU, 256GB RAM and six NVIDIA GeForce RTX 4090 GPUs, each with 24GB of VRAM.

Datasets and models. We evaluate REMIND on four real-world dataset with varying levels of complexity, including CIFAR-10 [21], CIFAR-100 [21], CINIC-10 [22], and Tiny-Imagenet [23]. Table I presents the statistics of these datasets.

TABLE I: Statistics of the datasets.

Datasets	#Train	#Test	#Features	#Classes
CIFAR-10	50K	10K	1024	10
CIFAR-100	50K	10K	1024	100
CINIC-10	90K	90K	1024	10
Tiny-Imagenet	100K	10K	4096	200

We conduct training using ResNet-18 [47] with approximately 2.7 million parameters.

Attack baselines. To validate REMIND, we introduce eight state-of-the-art FL backdoor attack methods for comparison:

- *Badnet* [43] performs the standard local training with Trojan data containing backdoors, and poisons the global model.
- *MR* [9] amplifies the malicious local updates to dominate the aggregation process.
- *LIE* [17] restricts the malicious local updates during training to ensure they remain within the distribution of benign ones.
- *Neurotoxin* [15] selectively perturbs the main task-irrelevant parameters in the local update.
- *DBA* [18] splits the original trigger into fragments, which are then assigned to each controlled client.
- *FCBA* [16] also adopts trigger splitting method, but each controlled client uses a combination of these fragments.
- *3DFed* [10] integrates indicators into the local update and uses the feedback to guide the attack in the next round.
- *F3BA* [24] flips main task-irrelevant model parameters during trigger embedding to maximize activation differences.

Defense methods. We examine the robustness of the attack methods under nine FL backdoor defenses across three categories: 1) pre-aggregation anomaly detection: FLAME [29] and FreqFed [31]; 2) byzantine-robust aggregation: Trimmed mean [36], Median [36], and Bulyan [35]; 3) post-aggregation model refinement: DP [37], FedPruning [40], FedDF [38], and FedRAD [39].

FL settings. By default, we set the number of clients $N = 100$, where only 20 clients are selected out of them randomly in every round. Each selected client conducts $E = 3$ epochs of local training using a SGD optimizer with a learning rate of 0.01 and a momentum of 0.9. Following previous studies [24], [48], we consider the non-IID data setting across clients, where local datasets are constructed using Dirichlet distribution [49] with a concentration parameter of 0.5 for CIFAR-10, CINIC-10 and Tiny-ImageNet, and 0.3 for CIFAR-100.

Attack settings. We randomly select a PMR proportion of all clients as malicious clients controlled by the attacker, where each malicious client's local training data contains a PDR fraction of poisoned samples. By default, PMR and PDR are set to 0.2 and 0.1, respectively. We design the trigger as a square located at the upper left corner of the image, and maintain the same trigger size across all attacks to ensure fair comparison. In low-frequency attack scenarios, the compromised clients upload malicious updates every T rounds, while submitting benign updates in other rounds. Unless otherwise specified, we set $T = 5$.

Evaluation metrics. According to previous works [10], [29], we employ two key metrics to assess the effectiveness of

TABLE II: The BA/MA (%) of REMIND and the baselines under different defense methods on CIFAR-10.

Defense \ Attack	Badnet	MR	LIE	Neurotoxin	DBA	FCBA	3DFed	F3BA	**REMIND**
FedAvg	92.19/88.01	97.18/87.99	86.55/87.92	89.29/87.79	6.38/87.88	47.8/87.65	99.24/88.97	94.61/86.64	**99.63**/88.81
FLAME	96.11/87.18	97.97/85.73	94.85/86.49	94.34/84.11	5.21/85.43	69.36/85.77	2.23/84.23	15.69/83.56	**99.98**/85.66
FreqFed	93.36/87.11	94.73/86.33	94.6/86.36	92.39/85.08	3.48/85.58	67.2/86.74	4.04/85.38	9.18/84.1	**100**/86.87
Trimmed Mean	93.87/88.54	93.73/88.55	90.08/88.1	86.51/87.85	2.15/88.4	37.33/88.28	99.15/88.91	95.45/87.5	**99.99**/88.64
Median	79.24/85.22	83.64/84.76	70.23/86.15	63.09/85.61	3.17/85.63	12.44/85.6	95.03/87.34	64.3/84.51	**98.99**/85.11
Bulyan	65.03/82.17	59.35/80.74	51.28/81.13	36.78/81.87	2.49/81.51	16.11/82.23	82.35/84.97	5.05/80.68	**95.29**/82.02
DP	91.57/86.71	94.79/86.78	82.79/86.7	84.44/86.05	3.42/86.17	42.02/86.07	99.02/88.43	86.79/84.94	**99.41**/86.38
FedDF	87.74/84.27	92.76/84.74	78.75/85.59	87.66/85.67	2.54/86.13	36.02/84.04	98.61/85.28	85.1/84.96	**99.12**/86.02
FedRAD	78.94/83.25	67.09/82.71	77.97/83.09	73.09/82.46	2.43/83.27	23.27/82.85	94.39/84.77	66.81/80.95	**99.77**/83.33
FedPruning	86.13/85.33	93.65/84.23	81.31/84.98	79.01/85.75	2.54/85.12	34.25/85.88	97.97/84.78	92.64/84.67	**99.46**/85.16

TABLE III: The BA/MA (%) of REMIND and the baselines under different defense methods on CIFAR-100.

Defense \ Attack	Badnet	MR	LIE	Neurotoxin	DBA	FCBA	3DFed	F3BA	**REMIND**
FedAvg	89.71/62.45	98.55/62.83	89.33/63.28	91.55/63.28	19.34/63.63	65.83/63.15	99.95/64.76	95.1/60.54	**99.98**/63.47
FLAME	92.92/56.87	94.67/57.91	94.54/57.42	94.04/57.35	19.42/57.64	64.5/57.9	0.66/53.2	1.72/54.11	**100**/57.5
FreqFed	95.8/57.57	90.6/58.61	93.58/56.33	93.38/55.25	13.76/57.54	52.4/56.71	0.68/54.77	1.43/54.25	**98.65**/57.93
Trimmed Mean	91.6/62.38	90.62/62.51	85.19/62.85	87.31/61.27	19.78/63.08	62.11/62.08	99.95/64.91	96.93/60.43	**99.99**/63.39
Median	73.58/57.18	74.52/57.44	77.17/57.19	55.78/57.12	3.44/58.36	23.91/57.26	92.77/61.4	46.39/53.92	**98.73**/57.22
Bulyan	60.94/50.47	36.78/50.05	62.11/48.26	39.57/48.57	6.04/49.94	21.62/50.72	31.24/53.78	0.94/45.72	**98.47**/49.55
DP	92.25/61.25	96.14/61.77	84.41/62.65	88.67/61.67	26.82/60.55	56.82/61.41	99.83/64.31	94.75/59.27	**99.94**/61.8
FedDF	89.23/60.37	93.5/58.25	86.19/60.67	89.12/61.3	20.33/60.81	56.71/60.76	99.92/64.42	94.98/57.91	**99.94**/61.67
FedRAD	87.29/61.44	80.43/60.23	78.56/60.87	84.89/59.81	24.01/60.55	48.49/59.97	99.8/61.43	89.71/57.6	**99.94**/61.79
FedPruning	90.33/62.25	97.76/62.57	83.55/62.28	88.31/61.14	16.23/62.19	53.83/61.32	99.76/64.67	97.11/60.59	**99.95**/62.26

TABLE IV: The BA/MA (%) of REMIND and the baselines under different defense methods on CINIC-10.

Defense \ Attack	Badnet	MR	LIE	Neurotoxin	DBA	FCBA	3DFed	F3BA	**REMIND**
FedAvg	96.39/73.76	96.24/72.42	92.95/75.05	93.65/75.27	4.33/75.14	82.77/74.9	99.64/77.66	89.22/73.89	**99.87**/75.21
FLAME	92.87/71.77	94.91/72.39	75.38/71.09	91.37/72.94	3.03/73.37	53.64/73.48	39.61/75.12	5.54/71.23	**99.98**/75.46
FreqFed	89.7/73.24	93.85/73.39	92.17/73.67	18.44/73.41	3.81/73.25	64.1/73.27	40.55/72.96	3.28/73.21	**99.89**/74.68
Trimmed Mean	94.25/74.09	94.38/73.55	94.99/73.43	95.37/74.72	3.39/75.82	24.3/74.08	99.73/77.14	97.53/74.52	**99.97**/74.89
Median	78.44/71.26	83.92/71.39	80.82/70.58	63.81/70.08	3.88/70.94	6.34/71.49	95.85/76.06	67.87/71.21	**98.75**/71.47
Bulyan	41.93/67.41	24.37/66.86	24.16/65.29	3.35/64.54	5.16/67.35	6.79/66.53	82.72/70.53	3.10/62.87	**94.39**/67.33
DP	94.04/72.47	95.71/72.72	89.33/74.75	95.38/74.71	3.96/72.85	36.83/73.14	99.09/76.32	92.65/71.99	**99.96**/74.14
FedDF	94.89/74.5	98.9/74.27	90.92/73.15	95.99/75.37	4.83/74.67	80.83/75.12	99.53*/77.56	79.34/72.93	**99.47**/75.22
FedRAD	91.51/73.49	92.81/71.24	87.24/72.77	90.61/72.22	8.03/71.07	49.1/71.07	98.01*/76.57	78.17/71.9	**97.85**/73.08
FedPruning	85.29/73.8	98.07/71.41	56.47/73.22	92.17/72.42	3.64/71.88	7.2/72.13	99.62*/75.32	92.23/72.95	**99.18**/73.43

TABLE V: The BA/MA (%) of REMIND and the baselines under different defense methods on Tiny-Imagenet.

Defense \ Attack	Badnet	MR	LIE	Neurotoxin	DBA	FCBA	3DFed	F3BA	**REMIND**
FedAvg	90.17/41.39	95.71/41.21	84.96/41.08	83.66/41.77	11.97/40.91	69.25/40.97	99.82/42.24	82.06/38.35	**99.9**/41.97
FLAME	90.19/35.57	0.51/37.02	90.76/35.89	92.31/36.52	7.42/36.45	45.75/36.71	0.17/34.12	0.68/33.41	**98.45**/36.5
FreqFed	78.12/36.22	0.46/36.47	69.85/36.66	84.61/36.32	0.52/36.52	18.66/35.92	0.23/34.51	0.39/33.26	**99.06**/36.43
Trimmed Mean	79.45/40.76	91.63/41.23	86.16/41.32	81.37/40.47	7.69/40.53	65.16/40.13	99.69*/42.24	94.11/38.07	**99.06**/40.61
Median	37.79/37.24	46.53/37.35	42.53/37.78	60.92/37.23	0.55/37.58	11.24/37.45	84.24/41.37	10.67/34.13	**90.61**/37.53
Bulyan	49.9/32.41	0.68/32.35	62.73/32.12	4.94/32.85	0.25/31.25	0.35/31.81	1.72/36.61	0.43/30.25	**73.12**/32.3
DP	90.93/41.03	96.38/41.26	81.06/41.21	86.59/41.03	10.77/40.91	74.24/41.09	99.71*/42.28	85.52/38.42	**99.05**/41.72
FedDF	89.41/41.55	93.51/40.78	81.91/41.17	83.7/41.09	8.41/40.98	72.09/40.15	99.54*/43.16	91.48/39.01	**99.03**/40.32
FedRAD	89.25/39.89	89.32/40.97	85.13/39.65	83.54/40.15	1.08/39.27	67.44/39.5	99.23*/40.9	87.27/38.38	**99.16**/40.69
FedPruning	90.09/41.23	94.47/41.23	83.36/41.38	90.54/41.44	9.66/41.57	78.5/41.25	99.6/41.45	91.26/39.04	**99.86**/41.4

REMIND. Specifically, Main task Accuracy (MA) refers to the proportion of correctly predicted samples out of the total test samples, serving as a measure of the model's performance on the main task. Backdoor Accuracy (BA) is the proportion of poisoned test samples that are successfully predicted as the target label, the same as the previously mentioned ASR.

B. Attack Performance

REMIND improves backdoor accuracy. Tables II-V present the performance of REMIND and the baselines under various defenses on CIFAR-10, CIFAR-100, CINIC-10 and Tiny-Imagenet, respectively. We observe that REMIND consistently achieves the highest BA across all cases. For instance, under the FreqFed defense on Tiny-Imagenet, REMIND attains a BA of 99.06%, while the next best attack, Neurotoxin, reaches only 84.61%. Under the Bulyan defense, REMIND maintains a high BA of 73.12%, outperforming all other baselines, which do not exceed 65%. Moreover, under all defense methods across four datasets (except for Bulyan on Tiny-ImageNet), REMIND achieves a BA of 90% or higher, exceeding 99% in most cases. This shows that REMIND effectively sustains strong attack effectiveness, even with low attack frequencies.

REMIND accelerates model convergence. To intuitively

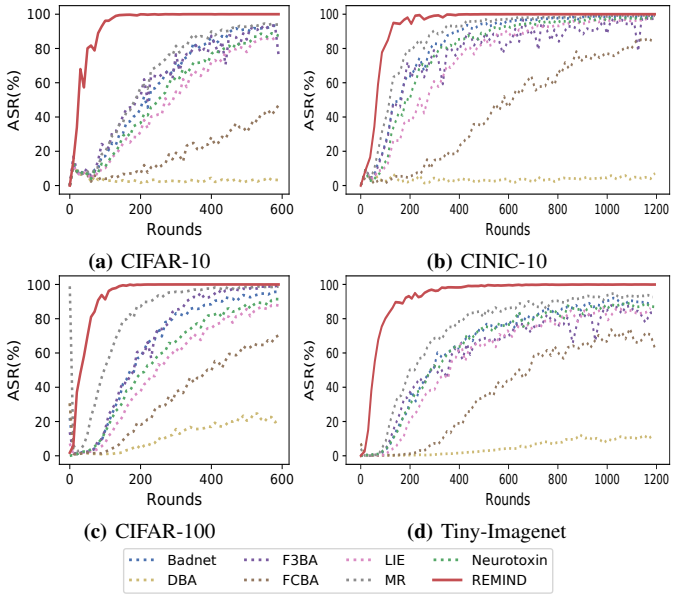


Fig. 4: The convergence curves of REMIND and the baselines.

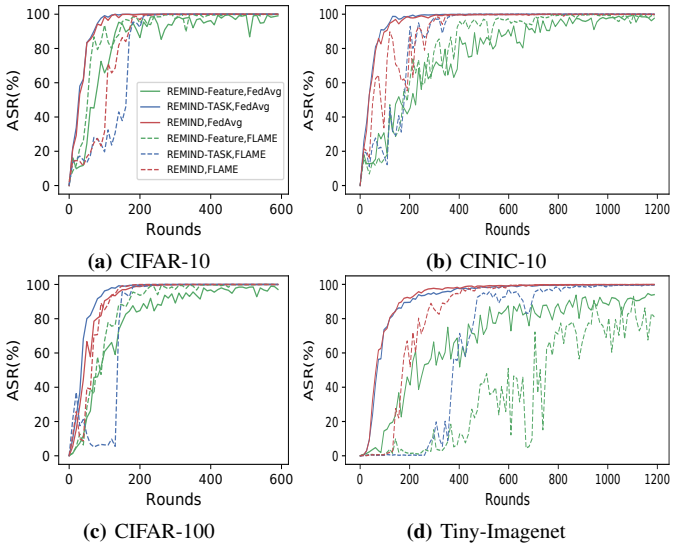


Fig. 5: The ASR (%) of REMIND with different reinforced memorization loss under FedAvg and FLAME.

TABLE VI: The BA/MA (%) of REMIND and REMIND w/o the prediction module with varying attack intervals.

Methods	T=1	T=3	T=5
	CIFAR-10		
REMIND w/o prediction module	93.77/88.72	92.8/88.64	91.84/88.35
REMIND	**99.98**/88.57	**99.53**/88.23	**99.63**/88.81
	CIFAR-100		
REMIND w/o prediction module	96.04/63.12	95.67/63.42	93.83/63.41
REMIND	**100**/63.53	**99.82**/63.28	**99.98**/63.47
	CINIC-10		
REMIND w/o prediction module	93.54/75.32	92.63/75.68	92.13/75.24
REMIND	**99.94**/75.13	**99.98**/75.77	**99.87**/75.21
	Tiny-Imagenet		
REMIND w/o prediction module	95.61/41.44	94.87/41.38	96.3/41.11
REMIND	**99.43**/41.63	**99.74**/41.58	**99.9**/41.97

demonstrate the backdoor injection speed, we compare REMIND with the baseline attacks across training rounds under the defense-free setting in Fig. 4. The results demonstrate that REMIND exhibits the fastest convergence across all datasets while the baselines either converge more slowly or plateau at a lower BA. By employing adaptive triggers to reinforce the memorization of the backdoor, REMIND enables rapid accumulation of the backdoor effect in the global model.

It is worth noting that we do not include the convergence curve of 3DFed in Fig. 4. Our experiments reveal that sometimes 3DFed exhibits faster convergence than REMIND and even outperforms it, as indicated by the entries marked with "*" in Tables IV and V. Upon reimplementation based on the official code, we enable 3DFed to first generate a single malicious update by training on $\mathcal{D}_{att}$, and then produces multiple poisoned updates by injecting noise into this base update. This practice implicitly increases the amount of poisoned samples. Hence, the accelerated convergence and better performance of 3DFed is expected, while this also makes it more likely to be detected by certain defense methods.

C. Ablation Study

To evaluate the effectiveness of each key component in REMIND, we conduct an ablation study on different datasets.

Effectiveness of reinforced memorization loss. In REMIND, the reinforced memorization loss performs task alignment and feature alignment simultaneously. To validate the roles of the two alignments, we compare REMIND with its two variants: REMIND-Feature, which only implements the feature alignment, and REMIND-Task, which solely enforces the task alignment. For REMIND, we assign equal weights to the two types of alignment. Fig. 5 illustrates the results of the three attacks under FedAvg and FLAME. It can be observed that without any defenses, REMIND and REMIND-Task yield comparable performance and significantly outperform REMIND-Feature. This indicates that task alignment can effectively enhance the attack effectiveness in low-frequency injection scenarios. However, in the FLAME setting, REMIND-Task performs much poorer than REMIND, and even underperforms REMIND-Feature on CIFAR-10 and CIFAR-100. This proves that feature alignment in REMIND plays a critical role in enhancing attack stealthiness and is indispensable.

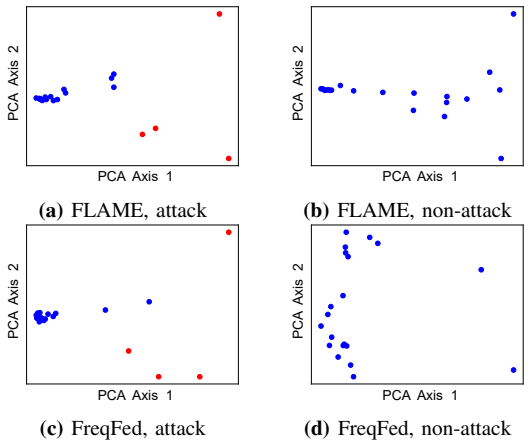
Effectiveness of prediction module. We verify the effectiveness of the prediction module by comparing REMIND with and without this component. As reported in Table VI, removing the prediction module results in a 3.6%–7.79% drop in BA across different attack intervals. In the experiments, we observe that the BA fluctuates without the prediction module. In contrast, REMIND with the prediction module converges more stably to a higher BA, highlighting its crucial role in preserving backdoor effect under low-frequency attacks.

D. Adverse Effects on Defense

We next validate the adverse effects induced by REMIND on all three defense categories.

TABLE VII: The TPR (%) under pre-aggregation anomaly detection methods with varying attack intervals.

Datasets	CIFAR-10	CIFAR-100	CINIC-10	Tiny-Imagenet
	FLAME			
T=1	30.88	90.42	44.21	98.92
T=3	9.79	21.38	13.25	28.38
T=5	**5.92**$_{\downarrow 24.96}$	**11.29**$_{\downarrow 79.13}$	**8.67**$_{\downarrow 35.54}$	**10.04**$_{\downarrow 88.88}$
	FreqFed			
T=1	34.5	85.96	28.75	66.17
T=3	11.71	18.33	13.33	13.88
T=5	**5.79**$_{\downarrow 28.71}$	**11.75**$_{\downarrow 74.21}$	**8.21**$_{\downarrow 20.54}$	**7.75**$_{\downarrow 58.42}$



(a) FLAME, attack **(b)** FLAME, non-attack **(c)** FreqFed, attack **(d)** FreqFed, non-attack

Fig. 6: Visualization of participating clients' similarity metrics, where red and blue represent malicious and benign clients, respectively.

Reduced detection probability. In this study, we employ the true positive rate (TPR) following previous works [29], [50] to measure the detection probability. Specifically, TPR is defined as the proportion of malicious clients successfully identified in each round to the total number of malicious clients involved. A higher TPR indicates a stronger ability of the defense to detect malicious clients. The reported TPR is averaged over the entire training process. As shown in Table VII, the TPR of two pre-aggregation anomaly detection methods, i.e., FLAME and FreqFed, gradually decreases as the attack interval T increases. For example, when $T = 5$, the TPR drops by 20.54%–88.88% compared to attacking in every round. This indicates that less frequent attacks are less likely to be detected, thereby improving the stealthiness of the backdoor.

Increased false alarm rate. To visualize the distribution of client-wise metrics in FLAME and FreqFed, we apply principal component analysis (PCA) [51] to project high-dimensional metrics into a two-dimensional space. The resulting visualizations are presented in Fig. 6. We observe that during attack rounds, the presence of highly deviating malicious clients causes benign clients to appear more tightly clustered, greatly reducing the likelihood of being detected by anomaly detection methods. While during non-attack rounds, the inherent variations among benign clients become more prominent, leading to some of them being mistakenly identified as malicious, consequently increasing false alarm rate.

Decreased main task accuracy. The MA of REMIND under three byzantine-robust aggregation methods are shown in Table VIII, where "No defense" refers to the setting without any defenses, serving as a baseline for comparison. We find that the MA degradation caused by defenses remains similar across different attack intervals. For instance, under the Bulyan defense on Tiny-ImageNet, the MA decreases by 9.41%, 9.39%, and 9.67% for three attack intervals $T = 1, 3, 5$, respectively. This suggests that these defenses fail to effectively distinguish between attack and non-attack rounds, and consistently exclude some benign updates from aggregation, thus resulting in a comparable loss in MA regardless of whether an attack occurs.

TABLE VIII: The MA (%) under byzantine-robust aggregation methods with varying attack intervals.

Datasets	CIFAR-10	CIFAR-100	CINIC-10	Tiny-Imagenet
No defense	88.81	63.47	75.21	41.97
	Trimmed Mean			
T=1	88.64$_{\downarrow 0.17}$	63.41$_{\downarrow 0.06}$	74.92$_{\downarrow 0.29}$	40.65$_{\downarrow 1.32}$
T=3	88.65$_{\downarrow 0.16}$	63.38$_{\downarrow 0.09}$	74.91$_{\downarrow 0.3}$	40.63$_{\downarrow 1.34}$
T=5	88.64$_{\downarrow 0.17}$	63.39$_{\downarrow 0.08}$	74.89$_{\downarrow 0.32}$	40.61$_{\downarrow 1.36}$
	Median			
T=1	85.28$_{\downarrow 3.53}$	57.36$_{\downarrow 6.11}$	71.49$_{\downarrow 3.72}$	37.54$_{\downarrow 4.43}$
T=3	85.27$_{\downarrow 3.54}$	57.24$_{\downarrow 6.23}$	71.49$_{\downarrow 3.72}$	37.66$_{\downarrow 4.31}$
T=5	85.11$_{\downarrow 3.7}$	57.22$_{\downarrow 6.25}$	71.47$_{\downarrow 3.74}$	37.53$_{\downarrow 4.44}$
	Bulyan			
T=1	82.14$_{\downarrow 6.67}$	49.83$_{\downarrow 13.64}$	67.39$_{\downarrow 7.82}$	32.56$_{\downarrow 9.41}$
T=3	82.12$_{\downarrow 6.69}$	49.67$_{\downarrow 13.8}$	67.35$_{\downarrow 7.86}$	32.58$_{\downarrow 9.39}$
T=5	82.02$_{\downarrow 6.79}$	49.55$_{\downarrow 13.92}$	67.33$_{\downarrow 7.88}$	32.3$_{\downarrow 9.67}$

TABLE IX: The per-round time cost (s) of different defense methods.

Datasets	CIFAR-10	CIFAR-100	CINIC-10	Tiny-Imagenet
Pre-aggregation	0.3	0.25	0.37	0.34
In-aggregation	1.73	1.77	1.83	1.73
FedDF	9.42$_{\times 5.45}$	8.89$_{\times 5.02}$	7.25$_{\times 3.96}$	13.97$_{\times 8.08}$
FedRAD	58.72$_{\times 33.94}$	55.81$_{\times 31.53}$	43.24$_{\times 23.63}$	76.92$_{\times 44.46}$
FedPruning	109.5$_{\times 63.29}$	148$_{\times 83.62}$	1211$_{\times 661.75}$	68.93$_{\times 39.84}$

Wasted resources. We present the average time per round for three types of defenses in Table IX, where "Pre-aggregation" and "In-aggregation" represent the average values of the first two categories, i.e., pre-aggregation anomaly detection and byzantine-robust aggregation. The results show that post-aggregation model refinement methods, namely, FedDF, FedRAD and FedPruing, incur significantly high execution time, approximately 3.96–661.75× that of the first two types. Hence, these post-aggregation techniques tend to waste substantial resources during non-attack rounds under low-frequency attacks.

VI. Conclusion

In this paper, we present REMIND, the first effective low-frequency backdoor attack framework in FL. To maintain a high ASR under low injection frequencies, REMIND reinforces backdoor memorization by aligning features and outputs of the poisoned and target-class samples. We further analyze its potential to improve attack efficiency and enhance robustness against defense methods. Extensive experiments validate the superiority of REMIND over existing attack baselines in terms of both attack effectiveness and persistence, thereby posing a new security threat to FL.

REFERENCES

[1] B. McMahan and D. Ramage, "Federated learning: Collaborative machine learning without centralized training data," Google AI, 2017. [Online]. Available: https://research.google/blog/federated-learning-collaborative-machine-learning-without-centralized-training-data/

[2] A. Yazdinejad, A. Dehghantanha, H. Karimipour, G. Srivastava, and R. M. Parizi, "A robust privacy-preserving federated learning model against model poisoning attacks," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 6693 – 6708, 2024.

[3] R. Ye, R. Ge, X. Zhu, J. Chai, D. Yaxin, Y. Liu, Y. Wang, and S. Chen, "Fedllm-bench: Realistic benchmarks for federated learning of large language models," in *Proc. NeurIPS*, 2024, pp. 111 106–111 130.

[4] M. Ye, W. Shen, B. Du, E. Snezhko, V. Kovalev, and P. C. Yuen, "Vertical federated learning for effectiveness, security, applicability: A survey," *ACM Computing Surveys*, vol. 57, no. 9, pp. 1–32, 2025.

[5] N. Yan, Y. Li, J. Chen, X. Wang, J. Hong, K. He, and W. Wang, "Efficient and straggler-resistant homomorphic encryption for heterogeneous federated learning," in *Proc. IEEE INFOCOM*, 2024, pp. 791–800.

[6] T. Awosika, R. M. Shukla, and B. Pranggono, "Transparency and privacy: the role of explainable ai and federated learning in financial fraud detection," *IEEE Access*, vol. 12, pp. 64 551–64 560, 2024.

[7] F. Cremonesi, M. Vesin, S. Cansiz, Y. Bouillard, I. Balelli, L. Innocenti, R. Taiello, S. Silva, S.-S. Ayed, M. Önen *et al.*, "Fed-biomed: Open, transparent and trusted federated learning for real-world healthcare applications," in *Federated Learning Systems: Towards Privacy-Preserving Distributed AI*. Springer, 2025, pp. 19–41.

[8] V. K. Quy, D. C. Nguyen, D. Van Anh, and N. M. Quy, "Federated learning for green and sustainable 6g iiot applications," *Internet of Things*, vol. 25, p. 101061, 2024.

[9] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proc. AISTATS*, 2020, pp. 2938–2948.

[10] H. Li, Q. Ye, H. Hu, J. Li, L. Wang, C. Fang, and J. Shi, "3dfed: Adaptive and extensible framework for covert backdoor attack in federated learning," in *Proc. IEEE S&P*, 2023, pp. 1893–1907.

[11] H. Zhang, J. Jia, J. Chen, L. Lin, and D. Wu, "A3fl: Adversarially adaptive backdoor attacks to federated learning," in *Proc. NeurIPS*, 2023.

[12] M. Li, W. Wan, Y. Ning, S. Hu, L. Xue, L. Y. Zhang, and Y. Wang, "Darkfed: A data-free backdoor attack in federated learning," in *Proc. IJCAI*, 2024, pp. 4443–4451.

[13] R. Kumar, G. Ebbrecht, J. Farooq, W. Wei, Y. Mao, and J. Chen, "Secfeddrive: Securing federated learning for autonomous driving against backdoor attacks," in *Proc. IEEE CNS*, 2024, pp. 1–6.

[14] M. A. Khan, Y. Chandio, E. Bagdasarian, and F. Anwar, "Compromising federated medical ai-backdoor risks in prompt learning," in *Proc. ACM SenSys*, 2025, pp. 630–631.

[15] Z. Zhang, A. Panda, L. Song, Y. Yang, M. Mahoney, P. Mittal, R. Kannan, and J. Gonzalez, "Neurotoxin: Durable backdoors in federated learning," in *Proc. ICML*, 2022, pp. 26 429–26 446.

[16] T. Liu, Y. Zhang, Z. Feng, Z. Yang, C. Xu, D. Man, and W. Yang, "Beyond traditional threats: A persistent backdoor attack on federated learning," in *Proc. AAAI*, 2024, pp. 21 359–21 367.

[17] G. Baruch, M. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," in *Proc. NeurIPS*, 2019.

[18] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "Dba: Distributed backdoor attacks against federated learning," in *Proc. ICLR*, 2019, pp. 1–19.

[19] M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao, "Heterogeneous federated learning: State-of-the-art and research challenges," *ACM Computing Surveys*, vol. 56, no. 3, pp. 1–44, 2023.

[20] J. Pei, W. Liu, J. Li, L. Wang, and C. Liu, "A review of federated learning methods in heterogeneous scenarios," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 3, pp. 5983–5999, 2024.

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

[22] L. N. Darlow, E. J. Crowley, A. Antoniou, and A. J. Storkey, "Cinic-10 is not imagenet or cifar-10," *arXiv preprint arXiv:1810.03505*, 2018.

[23] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.

[24] P. Fang and J. Chen, "On the vulnerability of backdoor defenses for federated learning," in *Proc. AAAI*, 2023, pp. 11 800–11 808.

[25] Y. Li, Y. Zhao, C. Zhu, and J. Zhang, "Infighting in the dark: Multi-label backdoor attack in federated learning," in *Proc. IEEE/CVF CVPR*, 2025, pp. 25 770–25 779.

[26] T. D. Nguyen, T. Nguyen, P. Le Nguyen, H. H. Pham, K. D. Doan, and K.-S. Wong, "Backdoor attacks and defenses in federated learning: Survey, challenges and future research directions," *Engineering Applications of Artificial Intelligence*, vol. 127, p. 107166, 2024.

[27] Y. Wan, Y. Qu, W. Ni, Y. Xiang, L. Gao, and E. Hossain, "Data and model poisoning backdoor attacks on wireless federated learning, and the defense mechanisms: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 3, pp. 1861–1897, 2024.

[28] P. Ye, Y. Li, K. He, Q. Li, T. Qin, X. Wang, K. Yang, C. Zhang, and J. Chen, "Breaking the illusion: A critical study of backdoor defense in federated learning with non-iid data," *IEEE Transactions on Information Forensics and Security,*, pp. 1–16, 2025.

[29] 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 *Proc. USENIX Security*, 2022, pp. 1415–1432.

[30] L. McInnes, J. Healy, S. Astels *et al.*, "hdbscan: Hierarchical density based clustering." *J. Open Source Softw.*, vol. 2, no. 11, p. 205, 2017.

[31] H. Fereidooni, A. Pegoraro, P. Rieger, A. Dmitrienko, and A.-R. Sadeghi, "Freqfed: A frequency analysis-based approach for mitigating poisoning attacks in federated learning," in *Proc. NDSS*, 2024.

[32] Z. Zhang, X. Cao, J. Jia, and N. Z. Gong, "Fldetector: Defending federated learning against model poisoning attacks via detecting malicious clients," in *Proc. ACM SIGKDD*, 2022, pp. 2545–2555.

[33] S. Lang, *A Second Course in Calculus*. Addison-Wesley Publishing Company, 1968, vol. 4197.

[34] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. NeurIPS*, 2017, pp. 119–129.

[35] R. Guerraoui, S. Rouault *et al.*, "The hidden vulnerability of distributed learning in byzantium," in *Proc. ICML*, 2018, pp. 3521–3530.

[36] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. ICML*, 2018, pp. 5650–5659.

[37] M. Naseri, J. Hayes, and E. De Cristofaro, "Local and central differential privacy for robustness and privacy in federated learning," in *Proc. NDSS*, 2022.

[38] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *Proc. NeurIPS*, 2020.

[39] S. P. Sturluson, S. Trew, L. Muñoz-González, M. Grama, J. Passerat-Palmbach, D. Rueckert, and A. Alansary, "Fedrad: Federated robust adaptive distillation," *arXiv preprint arXiv:2112.01405*, 2021.

[40] C. Wu, X. Yang, S. Zhu, and P. Mitra, "Mitigating backdoor attacks in federated learning," *arXiv preprint arXiv:2011.01767*, 2020.

[41] J. Feng, Y. Lai, H. Sun, and B. Ren, "Sadba: Self-adaptive distributed backdoor attack against federated learning," in *Proc. AAAI*, 2025.

[42] C. Chen, T. Liao, X. Deng, Z. Wu, S. Huang, and Z. Zheng, "Advances in robust federated learning: A survey with heterogeneity considerations," *IEEE Transactions on Big Data*, vol. 11, no. 3, pp. 1548–1567, 2025.

[43] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," *arXiv preprint arXiv:1708.06733*, 2017.

[44] W. Huang, M. Ye, Z. Shi, G. Wan, H. Li, B. Du, and Q. Yang, "Federated learning for generalization, robustness, fairness: A survey and benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 9387–9406, 2024.

[45] M. Badar, S. Sikdar, W. Nejdl, and M. Fisichella, "Fairtrade: Achieving pareto-optimal trade-offs between balanced accuracy and fairness in federated learning," in *Proc. AAAI*, 2024, pp. 10 962–10 970.

[46] S. Ghazi, S. Farzi, and A. Nikoofard, "Federated learning for all: A reinforcement learning-based approach for ensuring fairness in client selection," *IEEE Access*, vol. 13, pp. 118 515–118 535, 2025.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, 2016, pp. 770–778.

[48] 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 *Proc. AAAI*, 2023, pp. 9020–9028.

[49] T. Minka, "Estimating a dirichlet distribution," 2000.

[50] S. Li and Y. Dai, "Backdoorindicator: Leveraging ood data for proactive backdoor detection in federated learning," in *Proc. USENIX Security*, 2024, pp. 4193–4210.

[51] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37–52, 1987.