FEDSHIELD: ADVERSARIAL HONEYPOT-GUIDED BEHAVIORAL CLIENT SCREENING FOR ROBUST FEDERATED LEARNING

Anonymous authors

ABSTRACT

Federated Learning (FL) enables collaborative training of the model between distributed clients without sharing raw data, providing significant privacy benefits. However, such training is highly vulnerable to backdoor attacks, where malicious clients insert hidden noise that compromises the global model while preserving apparent performance on clean data. Existing defenses struggle under realistic FL conditions, where benign updates exhibit high variability, and adaptive attackers can avoid statistical detection. To address these challenges, we propose FedSHIELD, a server side, honeypot guided behavioral client screening framework which evaluates client updates by measuring their reduction in loss on adversarially crafted honeypot samples. This approach enables direct detection of malicious updates without relying on layer wise statistics or prior assumptions about attack patterns. Suspicious updates are filtered out, while the remaining updates are clipped and perturbed with differential privacy noise to ensure robustness and privacy preservation. Extensive experiments on benchmark datasets such as MNIST, FashionMNIST, SVHN, and CIFAR-10 demonstrate that FedSHIELD achieves substantially lower backdoor attack success rates while maintaining high clean-data accuracy. FedSHIELD outperforms existing defenses across both IID and non-IID data distributions.

1 INTRODUCTION

Federated Learning (FL) is a distributed learning procedure that enables multiple clients to collectively train models without sharing actual data McMahan et al. (2017). This approach provides significant privacy benefits and has been applied across multiple sectors including healthcare Ahmed et al. (2025), artificial intelligence of things Kouda et al. (2025), finance Buyuktanir et al. (2025), intelligent transportation Pei et al. (2025), and agriculture Dey et al. (2025). However, decentralization in FL introduces security vulnerabilities, especially to backdoor, where malicious clients inject hidden behaviors into the federated model Ding et al. (2025), Orabi et al. (2025). Prior studies Gong et al. (2022), Zhang et al. (2023), Fang & Chen (2023) show that FL is highly sensitive to backdoor attacks, where attackers implant hidden noise into the model. Such backdoors allow the federated model to act accurately on typical inputs but produce compromised results when special triggers are present, thereby evading standard evaluation Wan et al. (2024).

Existing defense mechanisms in FL are generally categorized into anomaly detection and robust aggregation Lyu et al. (2022). Robust aggregation methods, including Bulyan Guerraoui et al. (2018), Krum Blanchard et al. (2017), Median and Trimmed Mean Yin et al. (2018), RLR Ozdayi et al. (2021), and FLTrust Cao et al. (2021), seek to minimize the influence of Compromised updates by refining the aggregation process. However, their effectiveness is often limited in non-IID client settings or when the number of malicious clients is high. Anomaly detection approaches, such as FLDetector Zhang et al. (2022), FLAME Nguyen et al. (2022), MASA Xu et al. (2025), MultiMetrics Huang et al. (2023), FedDMC Mu et al. (2024), FLTrust Cao et al. (2021), Snowball Qin et al. (2024), and FedDLAD Ding et al. (2025), attempt to filter suspicious clients based on statistical or clustering techniques.

Existing defenses in FL primarily rely on norm-based, statistical, or parameter similarity metrics to detect malicious client updates. Additionally, layer-wise or clustering-based approaches assume

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

structural similarity among benign updates, making them vulnerable to attacks that carefully mimic real patterns. Indirect trust assessments based on cosine similarity further limit their performance, as malicious updates can target specific backdoors without significantly deviating from benign ones. To overcome these challenges, FedSHIELD introduces a server-side, behavioral client screening mechanism that evaluates each update based on its actual effect on adversarially crafted honeypot samples, measured via reduction in loss. FedSHIELD dynamically generates honeypots and applies light adversarial variation. Suspicious updates are then filtered out, and the remaining updates are aggregated with clipping and differential privacy noise, ensuring robustness and privacy. This behavioral, loss-based screening does not rely on statistical assumptions or indirect similarity metrics, enabling accurate detection of adaptive attacks while maintaining high accuracy even under non-IID distributions.

Following are the summarized contributions of this work:

- We introduce FedSHIELD, a server-side defense mechanism for FL that leverages honeypot guided behavioral screening which evaluates client updates based on their actual impact on the loss of adversarially crafted honeypot samples. This approach enables direct detection of stealthy backdoor attacks without relying on statistical assumptions or parameter similarity procedures.
- The aggregation mechanism of FedSHIELD integrates client exclusion, norm gradient clipping, and differential privacy noise. which preserves the contributions of benign clients while mitigating the effect of malicious updates. This approach achieves consistent model performance and balance under heterogeneous, IID and non-IID distributions of data.
- FedSHIELD significantly reduces backdoor attack rates and achieves high accuracy on most benchmarks such as MNIST, FashionMNIST, SVHN, and CIFAR-10 datasets. FedSHIELD outperforms existing defense mechanisms across diverse scenarios, including high-malicious-client and non-IID settings. We further show that the suspicion score separates benign and malicious clients with AUROC ≥ 0.978 across four datasets, including honeypots selected from a different dataset (App. D.2). FedSHIELD remains effective against a scoring-aware adaptive attacker (App. D.5) and outperforms FLTrust and Sageflow under an identical auxiliary-data budget (App. D.6).

The rest of the paper is organized as: Section 2 cover related backdoor detection and aggregation methods, Section 3 cover Preliminaries, Section 4 focuses on FedSHIELD architecture, Section 5 reports experimental results, and finally the article is concluded in Section 6.

2 Related Work

Recent defenses against backdoors in FL are typically classified into client-level backdoor detection and robust aggregation.

Client-Level Backdoor Detection Methods These methods aim to filter and identify malicious clients before aggregation, analyzing structural or statistical properties of client updates. For instance, clustering-based methods such as K-means Li et al. (2021), and HDBSCAN-based detection in FLAME Nguyen et al. (2022) and binary tree clustering with noise as used in FedDMC Mu et al. (2024). Other approaches such as FLDetector Zhang et al. (2022) leverages update stability, while Huang et al. (2023) combines multiple distance-based metrics named MultiMetrics. FedDLAD Ding et al. (2025) introduces a dual-layer filtering strategy based on layer-wise statistics. These approaches analyze model behavior indirectly. However, these methods retain residual malicious updates or misclassify benign clients, especially under heterogeneous data distributions.

Robust Aggregation Methods These approaches focuses to modify the aggregation rules and mitigate the impact of malicious client updates. Several aggregation methods are developed such as Median and Trimmed Mean Yin et al. (2018), Krum Blanchard et al. (2017), and RLR Ozdayi et al. (2021) aim to overcome outliers during the aggregation process, whereas FLTrust Cao et al. (2021) assigns trust scores to client updates based on a server-held root dataset. While these techniques have demonstrated effectiveness in IID data settings, their performance often degraded in the presence of non-IID data distributions, limiting their applicability in realistic federated learning scenarios.

Prior defense mechanisms primarily rely on robust aggregation or statistical filtering to mitigate backdoor attacks. In contrast, FedSHIELD adopts a fundamentally different strategy by performing behavioral backdoor detection through adversarial honeypot evaluation. Specifically, the FedSHIELD measures the reduction in loss on the honeypot samples caused by each client update. This mechanism allows the server to directly assess the malicious intent of updates rather than relying on parameter similarity heuristics or statistical deviation. This loss-driven evaluation enables robust detection of both stealthy and obvious backdoor attacks under diverse data distributions.

3 Preliminaries and Problem Setting

3.1 Federated Learning

We consider a standard FL system comprising K clients coordinated by a central server. Each client k holds a private dataset $D_k = \{(x_{k,i}, y_{k,i})\}_{i=1}^{n_k}$. At round of communication t, the server broadcasts the global model θ^t to the clients where each client performs local training and computes an update:

$$\Delta_k^t = \theta_k^{t+1} - \theta^t, \quad \theta_k^{t+1} = \theta^t - \eta \nabla f(\theta^t, D_k), \tag{1}$$

where η is the learning rate and $f(\cdot)$ is the local loss. The server aggregates updates to obtain the next global model:

$$\theta^{t+1} = \theta^t + \frac{1}{K} \sum_{k=1}^{K} \Delta_k^t. \tag{2}$$

3.2 Threat Model: Backdoor Attacks

We assume a subset of clients $\mathcal{K}_{\text{mal}} \subset \{1, \ldots, K\}$ is malicious. These clients can inject poisoned samples $D_k^{\text{poison}} \subset D_k$ and manipulate local updates. Benign clients follow the protocol and remain uncompromised.

The attacker aims to implant a backdoor such that the global model behaves normally on clean inputs $x \in D_{\text{clean}}$, but outputs a target label y_{target} for poisoned inputs $x' \in D_{\text{poison}}$:

$$M_\theta(x) = \begin{cases} y, & x \in D_{\text{clean}}, \\ y_{\text{target}}, & x \in D_{\text{poison}}. \end{cases} \tag{3}$$

This is achieved by optimizing a joint objective balancing clean and poisoned losses:

$$\min_\theta \mathbb{E}_{(x,y)\sim D_{\text{clean}}} \ell(M_\theta(x), y) + \lambda \, \mathbb{E}_{(x', y_{\text{target}})\sim D_{\text{poison}}} \ell(M_\theta(x'), y_{\text{target}}), \tag{4}$$

where λ controls the attack strength.

The server holds only one clean sample per class. These samples need not come from the clients' distribution: detection remains reliable when the honeypot is constructed from a different public dataset (App. D.2). This assumption is shared by FLTrust (Cao et al., 2021) and Sageflow (Park et al., 2021); we discuss its practicality in App. D.6.

3.3 Limitations and Motivation

Due to space constraints, a detailed discussion on the critical limitations of current defenses in realistic FL scenarios and the core motivation behind FedSHIELD is deferred to Appendix A.

Privacy-Preserving Honeypot Construction: A key design principle of FedSHIELD is that the server constructs the honeypot batch $\mathcal{H}_0$ entirely from a small, publicly available dataset $\mathcal{D}_{pub}$ (e.g., a held-out subset of CIFAR-10 or ImageNet), rather than from any client's private training data. This design choice is deliberate: since $\mathcal{D}_{pub}$ is explicitly outside the FL privacy boundary, no private client information is accessed or inferred by the server. This imposes a strictly weaker requirement than methods requiring server-side knowledge of the training distribution, and fully preserves the zero-knowledge privacy premise of federated learning.

4 METHODOLOGY

In this section, we present FedSHIELD, a server-side honeypot-based client filtering mechanism designed to defend FL systems against backdoor attacks. Unlike conventional defenses that rely on statistical anomaly detection or robust aggregation, FedSHIELD employs *behavioral testing* using adversarially crafted honeypots to identify malicious client updates while preserving benign contributions under heterogeneous data distributions.

FedSHIELD consists of four main components: (i) honeypot batch construction, (ii) adversarial trigger generation, (iii) suspicion scoring of client updates, and (iv) secure aggregation with gradient clipping and differential privacy (DP). Each component is described in detail below.

4.1 PROBLEM SETUP

Consider an FL system with K clients $\mathcal{K} = \{1, 2, \ldots, K\}$ at communication round t. Each client k holds a private dataset $\mathcal{D}_k$ and computes a local update Δ_k^t. The server maintains a global model θ^t and updates it as follows:

$$\theta^{t+1} = \theta^t + \Delta^{final}, \tag{5}$$

where Δ^{final} denotes the aggregated update after detecting and mitigating malicious clients.

4.2 HONEYPOT BATCH CONSTRUCTION

To identify malicious updates, the server constructs a honeypot batch $\mathcal{H}_0$ containing one representative sample per class from a small publicly available dataset $\mathcal{D}_{pub}$:

$$\mathcal{H}_0 = \{(x_c, y_c)\}_{c=1}^{C}, \quad (x_c, y_c) \sim \mathcal{D}_{pub}, \tag{6}$$

where $\mathcal{D}_{pub}$ denotes a small publicly available dataset (e.g., a held-out split of CIFAR-10 or ImageNet) that is entirely disjoint from all client training data $\{\mathcal{D}_k\}_{k=1}^{K}$, ensuring that the server accesses no private client information during honeypot construction and C is the number of classes. The initial reference loss on the unperturbed honeypot batch is computed using the current global model:

$$L_{ref} = \ell(\theta^t; \mathcal{H}_0), \tag{7}$$

where $\ell(\cdot)$ denotes the cross-entropy loss function. This reference loss serves as a baseline to detect imbalanced reduction caused by malicious updates as given in Algorithm 1 (see Appendix B.1).

4.3 ADVERSARIAL TRIGGER GENERATION

To amplify the sensitivity of the honeypot to malicious updates, small perturbations are applied to the batch using Projected Gradient Descent (PGD). Let $\delta^{(0)} = 0$ and iteratively update for T steps:

$$\delta^{(i)} = \Pi_{[-\epsilon,\epsilon]}\Big(\delta^{(i-1)} + \alpha \cdot \text{sign}(\nabla_\delta \ell(\theta^t; x_c + \delta^{(i-1)}, y_c))\Big), \quad i = 1, \ldots, T, \tag{8}$$

where $\Pi_{[-\epsilon,\epsilon]}$ projects each element of δ into $[-\epsilon, \epsilon]$, α is the PGD step size, and ϵ controls the maximum perturbation magnitude.

The final perturbed honeypot batch is defined as:

$$\mathcal{H} = \{(x_c + \delta_c^{(T)}, y_c)\}_{c=1}^{C}. \tag{9}$$

4.4 SUSPICION SCORING OF CLIENT UPDATES

For each client k, the server reconstructs the client model:

$$\hat{\theta}_k^t = \theta^t + \Delta_k^t. \tag{10}$$

The loss of this reconstructed model on the honeypot batch is:

$$L_k = \ell(\hat{\theta}_k^t; \mathcal{H}). \tag{11}$$

The suspicion score is then defined as the reduction in loss relative to the reference:

$$s_k = \max\{0, L_{ref} - L_k\}. \tag{12}$$

High s_k values indicate that the client update disproportionately reduces the honeypot loss, signaling potential malicious behavior. A malicious update minimizes the clean loss and the trigger-to-target loss jointly (Eq. 4). The PGD-perturbed honeypot lies in high-loss regions of the current global model, where such an update lowers the loss far more than an update trained on the clean objective alone. s_k measures this excess reduction. It depends on behavior rather than on update norms or similarity, so it is unaffected by stealthy updates that mimic benign statistics. Empirically, benign clients score 0.02–0.08 and malicious clients 0.86–0.98 across all datasets (App. D.2). Rather than applying a fixed exclusion fraction, FedSHIELD employs an adaptive threshold based on the statistical distribution of suspicion scores. Let μ_s and σ_s denote the mean and standard deviation of $\{s_k\}_{k=1}^{K}$. A client k is excluded only if its score exceeds:

$$\tau = \mu_s + \beta \cdot \sigma_s, \tag{13}$$

where $\beta > 0$ is a sensitivity parameter. The surviving client set is then:

$$\mathcal{S}_{keep} = \{k \mid s_k \leq \tau\}. \tag{14}$$

This adaptive thresholding ensures that benign edge-case clients with naturally higher honeypot loss reduction under non-IID distributions are not blindly excluded.

4.5 Update Clipping and Aggregation

To prevent individual updates from dominating the aggregation, each surviving update is clipped:

$$\tilde{\Delta}_k = \frac{\Delta_k^t}{\max(1, \|\Delta_k^t\|_2)}, \quad k \in \mathcal{S}_{keep}. \tag{15}$$

The updates are then aggregated uniformly:

$$\Delta^{agg} = \frac{1}{|\mathcal{S}_{keep}|} \sum_{k \in \mathcal{S}_{keep}} \tilde{\Delta}_k. \tag{16}$$

Screening is stateless: $\mathcal{H}$, s_k and τ are recomputed each round from that round's information only. First-time and hit-and-run clients are therefore screened exactly like long-standing ones (worked example in App. B.3). The resulting filtering error is at most 0.8% FPR and 0.8% FNR (App. D.3).

4.6 Differential Privacy Noise Addition

To provide client-level privacy, Gaussian noise is added to the aggregated update:

$$\Delta^{final} = \Delta^{agg} + \mathcal{N}(0, \sigma^2 I), \tag{17}$$

where σ controls the DP noise scale. Clipping followed by Gaussian noise gives client-level (ε, δ)-DP conditional on the retained set, with $\varepsilon \in [6.4, 14.1]$ over 200 rounds for $\sigma \in [1.0, 0.5]$ and $\delta = 10^{-5}$ (App. E.1).

4.7 Theoretical Rationale of FedSHIELD

FedSHIELD aims to mitigate backdoor attacks in FL by evaluating the behavioral response of client updates on a server-side honeypot (see Algorithm 2 in Appendix B.2 for the complete step-by-step procedure). We now formalize its theoretical properties.

1. Suspicion Score and Backdoor Sensitivity. Let the global model at round t be θ^t, and let the honeypot batch be

$$\mathcal{H} = \{(x_c + \delta_c^{(T)}, y_c)\}_{c=1}^{C}.$$

For a client k submitting update Δ_k^t, the reconstructed model is:

$$\hat{\theta}_k^t = \theta^t + \Delta_k^t. \tag{18}$$

Table 1: Comparison of Different Defenses Across Datasets ($\alpha = 0.5$ and $\alpha = 1$).

Defense	CIFAR-10			SVHN			FashionMNIST			MNIST		
	ASR↓	ACC↑	OPS↑	ASR↓	ACC↑	OPS↑	ASR↓	ACC↑	OPS↑	ASR↓	ACC↑	OPS↑
$\alpha = 0.5$												
FedAvg	0.854	0.778	0	0.810	0.918	0	0.954	0.913	0	1.000	0.993	0
FoolsGold	0.836	**0.781**	+0.025	0.836	0.923	-0.027	0.896	0.898	+0.044	0.993	0.992	+0.006
Median	0.816	0.771	+0.035	0.832	0.916	-0.009	0.624	0.883	+0.313	0.998	**0.993**	+0.002
FLAME	0.783	0.728	+0.019	0.751	0.875	+0.026	0.442	0.883	+0.504	0.226	0.992	+0.773
MultiMetrics	0.786	0.721	+0.006	0.662	0.812	+0.067	0.009	0.882	+0.957	0.009	0.989	+0.987
SnowBall	0.722	0.723	+0.084	0.817	0.844	-0.089	0.017	0.870	+0.935	0.007	0.980	+0.980
RLR	0.739	0.715	+0.054	0.772	0.838	-0.040	0.003	0.862	+0.941	0.002	0.990	+0.995
FLTrust	0.507	0.775	+0.402	0.062	0.884	+0.886	0.006	0.802	+0.872	0.005	**0.993**	+0.995
Krum	0.021	0.456	+0.562	0.615	0.209	-0.532	0.009	0.782	+0.847	0.009	0.974	+0.972
FedDLAD	0.025	0.746	+0.930	0.044	0.895	+0.921	0.007	0.888	+0.965	0.003	0.992	+0.996
FedSHIELD	**0.001**	0.762	**0.978**	**0.011**	**0.933**	**0.990**	**0.001**	**0.903**	**0.988**	**0.001**	**0.993**	**0.999**
$\alpha = 1$												
FedAvg	0.846	0.783	0	0.843	0.924	0	0.967	0.913	0	1.000	0.990	0
FoolsGold	0.822	**0.785**	+0.031	0.837	0.929	+0.013	0.921	**0.907**	+0.041	0.917	0.988	+0.081
Median	0.764	0.767	+0.076	0.846	0.920	-0.008	0.923	0.896	+0.027	0.997	0.991	+0.004
FLAME	0.160	0.724	+0.736	0.773	0.882	+0.038	0.944	0.889	-0.003	0.992	0.984	+0.002
MultiMetrics	0.296	0.751	+0.609	0.861	0.872	-0.078	0.018	0.824	+0.884	0.032	0.992	+0.970
SnowBall	0.546	0.757	+0.321	0.503	0.837	+0.449	0.025	0.762	+0.809	0.001	0.990	+0.999
RLR	0.046	0.714	+0.858	0.768	0.874	+0.035	0.003	0.876	+0.956	0.002	0.989	+0.997
FLTrust	0.022	0.731	+0.908	0.091	0.868	+0.831	0.006	0.810	+0.881	0.012	0.988	+0.986
Krum	0.006	0.449	+0.566	0.870	0.797	-0.169	0.009	0.793	+0.859	0.967	0.958	+0.001
FedDLAD	0.038	0.751	+0.914	**0.003**	0.904	+0.975	0.007	0.882	+0.959	0.004	0.993	+0.999
FedSHIELD	**0.001**	0.752	**0.959**	**0.001**	**0.931**	**0.985**	**0.003**	0.906	**0.989**	**0.000**	**0.994**	**+1.00**

The honeypot loss is and the suspicion score are defined as:

$$L_k = \ell(\hat{\theta}_k^t; \mathcal{H}), \tag{19}$$

$$s_k = \max(0, L_{\text{ref}} - L_k), \tag{20}$$

where $L_{\text{ref}} = \ell(\theta^t; \mathcal{H})$.

Observation 1: For a benign client, Δ_k^t is unlikely to overfit the adversarial honeypot, yielding $s_k \approx 0$. For a malicious client carrying a backdoor, Δ_k^t reduces the honeypot loss disproportionately, giving $s_k > 0$. Observation 1 is supported empirically in App. D.2.

2. Client Filtering Analysis. Let $\mathcal{S}_{\text{mal}}$ be the set of malicious clients, and $\mathcal{S}_{\text{keep}}$ the surviving clients after filtering. By excluding clients whose scores exceed the adaptive threshold $\tau = \mu_s + \beta \cdot \sigma_s$, FedSHIELD is designed such that:

$$\mathcal{S}_{\text{mal}} \cap \mathcal{S}_{\text{keep}} \approx \emptyset, \tag{21}$$

$$\mathcal{S}_{\text{benign}} \subseteq \mathcal{S}_{\text{keep}}. \tag{22}$$

This ensures that most malicious updates are excluded while preserving benign contributions.

3. Bounded Influence via Clipping. To further limit the impact of any individual update, we clip the L_2 norm:

$$\tilde{\Delta}_k = \frac{\Delta_k^t}{\max(1, \|\Delta_k^t\|_2)}, \quad k \in \mathcal{S}_{\text{keep}}. \tag{23}$$

This guarantees:

$$\|\tilde{\Delta}_k\|_2 \leq 1, \quad \forall k \in \mathcal{S}_{\text{keep}}, \tag{24}$$

which prevents unusually large updates from dominating the aggregation.

4. Aggregation and Convergence. The final aggregated update is

$$\Delta^{\text{agg}} = \frac{1}{|\mathcal{S}_{\text{keep}}|} \sum_{k \in \mathcal{S}_{\text{keep}}} \tilde{\Delta}_k, \tag{25}$$

and with optional differential privacy noise

$$\Delta^{\text{final}} = \Delta^{\text{agg}} + \mathcal{N}(0, \sigma^2 I), \tag{26}$$

the update satisfies client-level (ε, δ)-DP conditional on the retained set (App. E.1).

Proposition 1 (Convergence). Let F be L-smooth and bounded below by F^*. Suppose the mean of the retained benign updates satisfies $\Delta_{ben}^t = -\eta\,(\nabla F(\theta^t) + b^t + \xi^t)$ with $\|b^t\| \le B$, $\mathbb{E}[\xi^t \mid \theta^t] = 0$ and $\mathbb{E}\|\xi^t\|^2 \le V$, and let $\rho_t = |\mathcal{S}_{keep} \cap \mathcal{S}_{mal}|/|\mathcal{S}_{keep}| \le \rho$. Then for $\eta \le 1/(6L)$, the iterates of FedSHIELD satisfy

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\nabla F(\theta^t)\|^2 \le \frac{4\,(F(\theta^0) - F^*)}{\eta T} + 3\Big(B + \frac{2\rho}{\eta}\Big)^2 + 6L\eta V + \frac{2Ld\sigma^2}{\eta}.$$

FedSHIELD thus converges to a neighborhood of a stationary point whose radius depends on the benign bias B, the DP noise σ, and the fraction ρ of retained updates that are malicious. This fraction is small in practice, since fewer than 1% of malicious updates survive filtering (App. D.3). The proof is given in App. E.2.

5. Backdoor Suppression Analysis. Let $B(\theta)$ denote the expected backdoor effect of model θ on poisoned inputs. Rather than assuming a linear decomposition of the backdoor effect across client updates, FedSHIELD provides suppression through two complementary mechanisms. First, by filtering clients with disproportionately high honeypot loss reduction, the set $\mathcal{S}_{\text{keep}}$ is constructed to exclude the dominant sources of backdoor influence. Second, gradient clipping bounds the per-update contribution, limiting the magnitude of any residual backdoor signal that may persist in surviving updates. Formally, for a global model updated only from $\mathcal{S}_{\text{keep}}$:

$$B(\theta^t + \Delta^{\text{final}}) \le B(\theta^t) + \frac{|\mathcal{S}_{\text{keep}} \cap \mathcal{S}_{\text{mal}}|}{|\mathcal{S}_{\text{keep}}|} \cdot B_{\max}, \tag{27}$$

where $B_{\max}$ is the maximum backdoor effect of any single client update, and $|\mathcal{S}_{\text{keep}} \cap \mathcal{S}_{\text{mal}}|$ denotes the number of malicious clients that survive filtering. Since FedSHIELD is designed to exclude the most suspicious clients, this term is expected to remain small in practice, thereby suppressing the residual backdoor effect without requiring linearity assumptions on the loss landscape.

5 Results

We perform comprehensive evaluations to assess how well the FedSHIELD framework mitigates different backdoor attacks, while ensuring strong performance on clean data. The evaluation includes comparisons with several baseline defense methods and ablation studies to assess the impact of individual components. Performance is measured using metrics that capture both attack mitigation and overall model accuracy, providing a comprehensive assessment of the FedSHIELD robustness and reliability.

5.1 Experimental Setup

We simulate 50 clients, of which 20% are malicious and implant a backdoor mapping label "0" to "5"; half of the clients are sampled each round. We evaluate three backdoor attacks: CBA (Bagdasaryan et al., 2020), where every malicious client uses the full trigger; DBA (Xie et al., 2019), where the trigger is split across malicious clients; and SPA (Wang et al., 2020), which combines model replacement with PGD-constrained updates, as well as a scoring-aware adaptive attacker (App. D.5). We report attack success rate (ASR), clean accuracy (ACC) and the overall performance score (OPS) (Huang et al., 2023), which combines both relative to FedAvg. Full configurations, metric definitions and hyperparameters are given in App. C.

5.2 Analysis of Experimental Results

Performance under Non-IID Data To simulate practical FL scenarios, we split the MNIST, Fashion-MNIST, SVHN, and CIFAR-10 datasets using a Dirichlet distribution Li et al. (2022) with parameters $\alpha = 0.5$ and $\alpha = 1$, which controls the level of data heterogeneity among clients. Table 1, presents the performance of various defenses under these non-IID settings.

Methods such as Median and FoolsGold show limited effectiveness, with Median being susceptible to malicious updates in heterogeneous data settings, and FoolsGold relying on the assumption of similar

Table 2: Performance under different attack settings with IID data.

Defense	SPA		CBA		DBA		OPS↑
	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	
FedAvg	0.764	0.801	0.866	0.802	0.860	0.800	0
FoolsGold	0.801	0.790	0.852	0.797	0.796	0.788	+0.002
Median	0.005	0.791	0.851	0.801	0.819	0.794	+0.346
SnowBall	0.013	0.802	0.317	0.796	0.249	0.795	+0.772
RLR	0.003	0.792	0.274	0.738	0.023	0.739	+0.829
MultiMetrics	0.002	0.795	0.050	0.788	0.233	0.775	+0.871
Krum	0.016	0.679	0.011	0.695	0.006	0.655	+0.831
FLAME	0.004	0.788	0.009	0.772	**0.002**	0.782	+0.969
FedDLAD	0.002	0.795	0.033	0.793	**0.002**	0.791	**+0.976**
FLTrust	**0.001**	0.784	0.803	0.780	0.543	0.760	+0.447
FedSHIELD	**0.001**	**0.817**	**0.006**	**0.812**	0.003	**0.804**	+1.00

Table 3: Ablation Study of FedSHIELD Components Across Datasets ($\alpha = 0.5$ and $\alpha = 1$)

Variant	MNIST		FashionMNIST		SVHN		CIFAR-10	
	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑
$\alpha = 0.5$								
Honeypot Only	0.081	0.981	0.094	0.871	0.192	0.892	0.221	0.742
Adversarial Honeypot	0.041	0.986	0.053	0.884	0.091	0.907	0.103	0.754
Adv. Honeypot + Behavioral Scoring	0.012	0.991	0.018	0.894	0.034	0.921	0.046	0.758
Adv. Honeypot + Scoring + Filtering	0.004	0.992	0.006	0.899	0.019	0.928	0.013	0.760
FedSHIELD	**0.001**	**0.993**	**0.001**	**0.903**	**0.011**	**0.933**	**0.001**	**0.762**
$\alpha = 1$								
Honeypot Only	0.063	0.983	0.082	0.879	0.171	0.901	0.198	0.726
Adversarial Honeypot	0.029	0.988	0.041	0.892	0.079	0.915	0.091	0.732
Adv. Honeypot + Behavioral Scoring	0.010	0.991	0.015	0.901	0.028	0.926	0.039	0.745
Adv. Honeypot + Scoring + Filtering	0.003	0.993	0.005	0.904	0.012	0.930	0.009	0.747
FedSHIELD	**0.000**	**0.994**	**0.003**	**0.906**	**0.001**	**0.931**	**0.001**	**0.752**

malicious updates, which does not always hold. Krum provides moderate protection on MNIST and FashionMNIST but struggles on SVHN and CIFAR-10. RLR reduces the ASR significantly on MNIST and FashionMNIST while maintaining accuracy, but its performance deteriorates on more complex datasets. FLTrust achieves good defense on MNIST but experiences a noticeable drop in ACC on FashionMNIST. FLAME mitigates some negative effects of malicious updates, yet it slightly compromises benign performance. MultiMetrics and SnowBall perform reasonably on MNIST and FashionMNIST, but their defense weakens on SVHN and CIFAR-10. FedDLAD demonstrates competitively better defense across all datasets, achieving very low ASR while maintaining high ACC. Notably, on SVHN, it reduces ASR to 0.044 with minimal accuracy loss, leading to high OPS scores. The proposed FedSHIELD outperforms all other methods, achieving the lowest ASR across all datasets while preserving strong ACC, indicating its superior capability to defend against backdoor attacks under non-IID data distributions.

Performance under IID Data. We also evaluate defenses in the IID scenario on CIFAR-10 against CBA, DBA, and SPA attacks, as summarized in Table 2. Krum effectively lowers ASR to around 1–2% across attack types, but at the cost of reduced ACC. Median, MultiMetrics, and SnowBall show stronger defense against SPA than against CBA and DBA. RLR and FLTrust are less effective against CBA, with ASR increasing substantially. In contrast, FLAME and FedDLAD maintain a stable defense performance, with FedDLAD slightly outperforming FLAME in ACC while keeping ASR extremely low. Across both settings, among the baselines, FedDLAD achieves the highest OPS, demonstrating its ability to suppress backdoor attacks without sacrificing benign accuracy. FedSHIELD further improves upon this, consistently achieving near-zero ASR and the highest OPS across all attack scenarios, highlighting its effectiveness as a robust defense mechanism in federated learning.

5.3 Additional Analyses

Stability across seeds. To further assess the stability of FedSHIELD, we additionally repeat its experiments over three random seeds; the standard deviations are small in all settings (App. D.1).

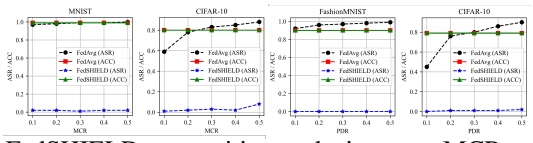

Figure 1: FedSHIELD competitive analysis across MCRs and PDRs.

Matched auxiliary data. Given the same budget of one sample per class, FedSHIELD reduces ASR to 0.1%, versus 41–52% for FLTrust and Sageflow on CIFAR-10 (App. D.6). The gain therefore comes from how the data are used, not merely from having them.

Adaptive attacker. An attacker that explicitly penalizes its own honeypot-loss reduction raises ASR only to at most 4.7%, and its suspicion score remains well above the benign range (App. D.5).

Heterogeneity. Under extreme label skew ($\alpha = 0.1$), benign FPR rises to at most 3.9% while FNR is unaffected (App. D.4).

Computational cost. FedSHIELD adds modest server overhead: with 1000 clients it requires 0.20 s per round, versus 0.44 s for FLAME, 1.37 s for FLTrust and 40.2 s for Krum (App. D.7).

5.4 Ablation Study

We perform an architectural ablation study to quantify the contribution of each component in FedSHIELD. Variants include: Honeypot Only: Clean honeypot evaluation without adversarial perturbation or filtering. Adversarial Honeypot: PGD-perturbed honeypot batch, no behavioral scoring. Adversarial Honeypot + Behavioral Scoring: Updates ranked by honeypot loss reduction, without filtering. Adversarial Honeypot + Scoring + Filtering: Suspicious clients excluded and updates clipped. FedSHIELD (Full): Complete framework with adversarial honeypots, behavioral scoring, filtering, clipping, and DP. Table 3 shows ASR and ACC across datasets. The ablation study shows that using Honeypot Only provides limited backdoor suppression, while adding Adversarial perturbations reduces ASR but cannot fully eliminate malicious updates. Incorporating Behavioral scoring further lowers ASR by detecting updates that disproportionately affect the honeypot, Filtering and clipping improve robustness on complex datasets. The complete FedSHIELD framework consistently achieves the lowest ASR while maintaining high accuracy, highlighting the importance of each component.

5.5 Impact of MCR and PDR

Figure 1 illustrates the robustness of FedSHIELD under varying Malicious Client Ratio (MCR) and Poisoned Data Ratio (PDR). On MNIST, ASR remains extremely low (0.01–0.02) while ACC stays constant at 0.99 across all MCR values. For CIFAR-10, ASR ranges from 0.01 to 0.08 and ACC is stable at 0.80, indicating effective mitigation even with increasing malicious clients. On FashionMNIST, ASR stays near to 0 and ACC remains 0.90 for all PDR levels, while CIFAR-10 shows a slight ASR increase to 0.02 at the highest PDR, with ACC same at 0.79. These results confirm that FedSHIELD consistently neutralizes backdoor attacks while preserving clean-data performance.

6 Conclusion

Federated Learning is highly susceptible to backdoor attacks, especially under non-IID data distributions and adaptive adversaries, where existing defenses based on statistical analysis and robust aggregation often fail. In this work, we proposed FedSHIELD, a server-side honeypot-guided behavioral defense that detects malicious client updates by evaluating their impact on the loss of adversarially crafted honeypot samples. By filtering suspicious clients and applying robust aggregation with gradient clipping and differential privacy noise, FedSHIELD effectively mitigates backdoor attacks while preserving clean-data performance. Experimental results on multiple benchmark datasets demonstrate that FedSHIELD consistently outperforms existing defenses across diverse federated learning scenarios, highlighting its practicality and robustness for secure FL deployments. Future work will extend FedSHIELD to asynchronous client participation and heterogeneous model architectures in large-scale federated learning systems; current limitations are discussed in Appendix G.

AI DISCLOSURE

In this work, we used generative AI tools to assist with parts of the theoretical analysis and to support the interpretation of the additional experimental results reported in the appendix. We have not used generative AI tools to generate synthetic data sets, propose or refine hypotheses, design the research methodology or experiments, implement methods, or clean and reformat data sets, and translation and qualitative or thematic data analysis are not applicable to this work. We also used generative AI tools to edit the paper to improve readability, to suggest a structure for the appendix, and to format tables. All AI-assisted work has been reviewed and checked by the authors. We take responsibility for the final content of this work, including text, claims and artifacts.

ETHICS STATEMENT

This work aims to improve the security of federated learning by defending against backdoor attacks, which is intended to benefit applications in privacy-sensitive domains such as healthcare and finance. All experiments use publicly available benchmark datasets (MNIST, FashionMNIST, SVHN, CIFAR-10, and, for cross-dataset honeypots, CIFAR-100 and DeepFashion); no human subjects, personal data or private client data were involved. FedSHIELD constructs its honeypot exclusively from public data and never accesses client training data. We study existing backdoor attacks and an adaptive attack variant solely to evaluate defenses; these attacks are already known in the literature, and our contribution is defensive. We acknowledge that behavioral screening exclude benign clients with highly heterogeneous data, which affect fairness for underrepresented participants; we quantify this effect in Appendix D.4 and discuss it further in Appendices F and G. The authors have read and adhere to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

We provide all details needed to reproduce our results. The complete FedSHIELD procedure, including honeypot construction and client filtering, is given in Algorithms 1 and 2 (Appendix B). Datasets, model architectures, data partitioning, honeypot sources, attack configurations, baselines, evaluation metrics and all hyperparameters (including random seeds) are specified in Appendix C. Results over multiple random seeds are reported in Appendix D.1. The assumptions and complete proof of our theoretical result are given in Appendix E.2, and the privacy accounting is detailed in Appendix E.1. Our source code will be made publicly available upon acceptance.

REFERENCES

Syed Thouheed Ahmed, AC Kaladevi, Achyut Shankar, Fayez Alqahtani, et al. Privacy enhanced edge-ai healthcare devices authentication: a federated learning approach. *IEEE Transactions on Consumer Electronics*, 2025.

Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International conference on artificial intelligence and statistics*, pp. 2938–2948. PMLR, 2020.

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30, 2017.

Busra Buyuktanir, Şahsene Altinkaya, Gozde Karatas Baydogmus, and Kazim Yildiz. Federated learning in intrusion detection: advancements, applications, and future directions. *Cluster Computing*, 28(7):473, 2025.

Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. In *Proceedings of NDSS*, 2021.

Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

Tanushree Dey, Somnath Bera, Anwesha Mukherjee, Debashis De, and Rajkumar Buyya. Flyer: Federated learning-based crop yield prediction for agriculture 5.0. *IEEE Transactions on Artificial Intelligence*, 2025.

Binbin Ding, Penghui Yang, and Sheng-Jun Huang. Feddlad: A federated learning dual-layer anomaly detection framework for enhancing resilience against backdoor attacks. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pp. 5021–5029, 2025.

Pei Fang and Jinghui Chen. On the vulnerability of backdoor defenses for federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11800–11808, 2023.

Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. The limitations of federated learning in sybil settings. In *23rd International symposium on research in attacks, intrusions and defenses (RAID 2020)*, pp. 301–316, 2020.

Xueluan Gong, Yanjiao Chen, Qian Wang, and Weihan Kong. Backdoor attacks and defenses in federated learning: State-of-the-art, taxonomy, and future directions. *IEEE Wireless Communications*, 30(2):114–121, 2022.

Rachid Guerraoui, Sébastien Rouault, et al. The hidden vulnerability of distributed learning in byzantium. In *International conference on machine learning*, pp. 3521–3530. PMLR, 2018.

Siquan Huang, Yijiang Li, Chong Chen, Leyu Shi, and Ying Gao. Multi-metrics adaptively identifies backdoors in federated learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4652–4662, 2023.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pp. 5132–5143. PMLR, 2020.

Mohamed Amine Kouda, Badis Djamaa, and Ali Yachir. An efficient federated learning solution for the artificial intelligence of things. *Future Generation Computer Systems*, 163:107533, 2025.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images.(2009), 2009.

Dongcheng Li, W Eric Wong, Wei Wang, Yao Yao, and Matthew Chau. Detection and mitigation of label-flipping attacks in federated learning systems with kpca and k-means. In *2021 8th International Conference on Dependable Systems and Their Applications (DSA)*, pp. 551–559. IEEE, 2021.

Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pp. 965–978. IEEE, 2022.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HJxNAnVtDS`.

Lingjuan Lyu, Han Yu, Xingjun Ma, Chen Chen, Lichao Sun, Jun Zhao, Qiang Yang, and Philip S Yu. Privacy and robustness in federated learning: Attacks and defenses. *IEEE transactions on neural networks and learning systems*, 35(7):8726–8746, 2022.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pp. 263–275. IEEE, 2017.

Xutong Mu, Ke Cheng, Yulong Shen, Xiaoxiao Li, Zhao Chang, Tao Zhang, and Xindi Ma. Feddmc: Efficient and robust federated learning via detecting malicious clients. *IEEE Transactions on Dependable and Secure Computing*, 21(6):5259–5274, 2024.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 7. Granada, 2011.

Thien Duc Nguyen, Phillip Rieger, Huili Chen, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Shaza Zeitouni, et al. {FLAME}: Taming backdoors in federated learning. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 1415–1432, 2022.

Menna Mamdouh Orabi, Osama Emam, and Hanan Fahmy. Adapting security and decentralized knowledge enhancement in federated learning using blockchain technology: literature review. *Journal of Big Data*, 12(1):55, 2025.

Mustafa Safa Ozdayi, Murat Kantarcioglu, and Yulia R Gel. Defending against backdoors in federated learning with robust learning rate. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 9268–9276, 2021.

Jungwuk Park, Dong-Jun Han, Minseok Choi, and Jaekyun Moon. Sageflow: Robust federated learning against both stragglers and adversaries. *Advances in neural information processing systems*, 34:840–851, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Jiaming Pei, Marwan Omar, Maryam Mohamed Al Dabel, Shahid Mumtaz, and Wei Liu. Federated few-shot learning with intelligent transportation cross-regional adaptation. *IEEE Transactions on Intelligent Transportation Systems*, 2025.

Zhen Qin, Feiyi Chen, Chen Zhi, Xueqiang Yan, and Shuiguang Deng. Resisting backdoor attacks in federated learning via bidirectional elections and individual perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 14677–14685, 2024.

Yichen Wan, Youyang Qu, Wei Ni, Yong Xiang, Longxiang Gao, and Ekram Hossain. Data and model poisoning backdoor attacks on wireless federated learning, and the defense mechanisms: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 26(3):1861–1897, 2024.

Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. *Advances in neural information processing systems*, 33:16070–16084, 2020.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *International conference on learning representations*, 2019.

Jiahao Xu, Zikai Zhang, and Rui Hu. Identify backdoored model in federated learning via individual unlearning. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 7960–7969. IEEE, 2025.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International conference on machine learning*, pp. 5650–5659. Pmlr, 2018.

Hangfan Zhang, Jinyuan Jia, Jinghui Chen, Lu Lin, and Dinghao Wu. A3fl: Adversarially adaptive backdoor attacks to federated learning. *Advances in neural information processing systems*, 36: 61213–61233, 2023.

Zaixi Zhang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Fldetector: Defending federated learning against model poisoning attacks via detecting malicious clients. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 2545–2555, 2022.

A MOTIVATION

Current defenses against backdoor attacks in FL rely heavily on robust aggregation or statistical anomaly detection to identify malicious client updates. These approaches can be effective under simplified conditions, but they have critical limitations in realistic FL scenarios (Gong et al., 2022). In non-IID settings, benign client updates naturally vary in direction and magnitude, which makes it hard to detect malicious behavior apart from legitimate data heterogeneity. Adaptive attackers can also craft updates that look statistically normal and so bypass these defenses. As a result, malicious updates go undetected while benign clients may be wrongly excluded, which harms both robustness and global model performance.

To address these challenges, FedSHIELD evaluates client behavior using server-side honeypot samples. Instead of relying on layer-wise statistics, norms or similarity heuristics, the server measures the actual effect of each client update on the loss of adversarially perturbed honeypot inputs. This loss-driven evaluation is a direct indicator of malicious behavior. It lets FedSHIELD filter compromised updates while keeping the contributions of benign clients, including under heterogeneous (non-IID) data and stealthy adaptive attacks (Appendix D).

B ALGORITHMIC DETAILS

B.1 HONEYPOT BATCH CONSTRUCTION

Algorithm 1 Honeypot Adversarial Perturbation Generation

Require: Honeypot batch $\mathcal{H}_0 = \{(x_c, y_c)\}_{c=1}^{C}$, global model θ^t, PGD steps T, step size α, perturbation budget ϵ
Ensure: Perturbed honeypot batch $\mathcal{H}$
1: Initialize $\delta^{(0)} \leftarrow 0$
2: **for** $i = 1$ to T **do**
3: $\quad \delta^{(i)} \leftarrow \Pi_{[-\epsilon,\epsilon]}\big(\delta^{(i-1)} + \alpha \cdot \mathrm{sign}(\nabla_\delta \ell(\theta^t; x_c + \delta^{(i-1)}, y_c))\big)$
4: **end for**
5: $\mathcal{H} \leftarrow \{(x_c + \delta_c^{(T)}, y_c)\}_{c=1}^{C}$
6: **return** $\mathcal{H}$

B.2 FEDSHIELD ALGORITHM

Algorithm 2 FedSHIELD: Honeypot-Based Client Filtering (one round)

Require: Global model θ^t, client updates $\{\Delta_k^t\}_{k=1}^{K}$, number of classes C, PGD steps T, step size α, perturbation budget ϵ, sensitivity parameter β, DP noise σ
Ensure: Aggregated update Δ^{final}
1: Construct perturbed honeypot batch $\mathcal{H}$ (Algorithm 1)
2: Compute reference loss $L_{ref} = \ell(\theta^t; \mathcal{H})$
3: **for** $k = 1$ to K **do**
4: $\quad$ Reconstruct client model $\hat{\theta}_k^t = \theta^t + \Delta_k^t$
5: $\quad$ Compute honeypot loss $L_k = \ell(\hat{\theta}_k^t; \mathcal{H})$
6: $\quad$ Compute suspicion score $s_k = \max(0, L_{ref} - L_k)$
7: **end for**
8: Compute $\mu_s = \frac{1}{K}\sum_k s_k$, $\sigma_s = \mathrm{std}(\{s_k\})$
9: Set $\tau = \mu_s + \beta \cdot \sigma_s$
10: Select $\mathcal{S}_{keep} = \{k \mid s_k \le \tau\}$
11: **for all** $k \in \mathcal{S}_{keep}$ **do**
12: $\quad$ Clip updates $\tilde{\Delta}_k = \Delta_k^t / \max(1, \|\Delta_k^t\|_2)$
13: **end for**
14: Aggregate updates $\Delta^{agg} = \frac{1}{|\mathcal{S}_{keep}|}\sum_{k \in \mathcal{S}_{keep}} \tilde{\Delta}_k$
15: Add DP noise $\Delta^{final} = \Delta^{agg} + \mathcal{N}(0, \sigma^2 I)$
16: **return** Δ^{final}

B.3 Stateless Screening: A Worked Example

Every quantity in Algorithm 2 ($\mathcal{H}$, L_{ref}, L_k, s_k, τ) is recomputed each round from that round's information only, so FedSHIELD requires no participation history. This distinguishes it from defenses that accumulate per-client statistics across rounds, which cannot judge a client they have not observed before.

Scenario. Consider a cross-device deployment with $K = 50$ registered devices, of which 25 are sampled per round. An adversary registers a new device, is sampled once in round $t = 47$, submits a single poisoned update Δ_k^{47}, and does not participate again. In the same round, a benign device that has also just joined is sampled for the first time. The server has no prior record of either device.

How round 47 is screened.

1. *Honeypot and reference loss.* The server perturbs $\mathcal{H}$ against the current global model (Algorithm 1) and computes $L_{ref} = \ell(\theta^{47}; \mathcal{H})$. This depends only on θ^{47}, not on which devices participate.

2. *Per-update scoring.* For each participant, the server evaluates $\hat{\theta}_k^{47} = \theta^{47} + \Delta_k^{47}$ on $\mathcal{H}$ and sets $s_k = \max(0,\, L_{ref} - L_k)$. Using the typical CIFAR-10 values in Table 6, the new benign device scores $s \approx 0.06$, like the other benign participants, while the adversarial device scores $s \approx 0.91$.

3. *Cohort-based threshold.* For illustration, suppose the other 24 participants are benign with $s \approx 0.06$. Then $\mu_s \approx 0.094$, $\sigma_s \approx 0.167$ and $\tau_{47} = \mu_s + 0.7\,\sigma_s \approx 0.21$. The adversarial update ($0.91 > 0.21$) is excluded, and the new benign device ($0.06 < 0.21$) is kept.

Neither decision uses any information from earlier rounds. The one-time attacker is filtered the moment it participates, and the benign newcomer is not penalized for lacking a track record. A single attacker in a round is, if anything, easier to isolate, because it stands out more clearly against the cohort statistics. All per-round statistics in Appendix D are computed under this stateless evaluation. They therefore apply directly to newly joining and hit-and-run clients.

C Extended Experimental Setup

C.1 Datasets and Models

We use four benchmark datasets: FashionMNIST (Xiao et al., 2017), CIFAR-10 (Krizhevsky et al., 2009), MNIST (Deng, 2012) and SVHN (Netzer et al., 2011). They differ in data distribution and complexity. For FashionMNIST and MNIST we use a CNN with five convolutional layers, followed by four fully connected layers for FashionMNIST and two for MNIST. For CIFAR-10 and SVHN we use VGG9. Non-IID partitions are generated with a Dirichlet distribution with $\alpha \in \{0.5, 1\}$ (and $\alpha = 0.1$ for the stress test in Appendix D.4).

C.2 Honeypot Data Source

Default (Same-Dist.). The server samples one representative image per class from a held-out public split $\mathcal{D}_{pub}$, strictly disjoint from all client training partitions. FedSHIELD therefore never accesses client private data.

Cross-dataset honeypots (Cross-Dist.). To test whether FedSHIELD depends on the honeypot matching the client distribution, we also construct the honeypot from a different public dataset, again taking one sample per class: CIFAR-100 for CIFAR-10, MNIST for SVHN, DeepFashion for FashionMNIST, and SVHN for MNIST. For the digit tasks the label spaces coincide, so samples are used directly for the corresponding digit. For CIFAR-10 and FashionMNIST, each target class is represented by a semantically related class of the source dataset. All honeypot images are resized and converted to the target resolution and number of channels.

C.3 Attack Configurations

We evaluate FedSHIELD against three backdoor attacks: the Distributed Backdoor Attack (DBA) (Xie et al., 2019), the Centralized Backdoor Attack (CBA) (Bagdasaryan et al., 2020), and the Single-shot with PGD Attack (SPA) (Wang et al., 2020).

CBA. Every malicious client has full knowledge of the trigger, a square pattern embedded in the input samples. This is the conventional backdoor scenario with complete trigger access.

DBA. The full trigger is split into segments, and each malicious client holds only its local piece. We use a cross-shaped trigger, with each client holding one quarter of the pattern. This increases stealthiness and makes detection less likely.

SPA. This attack combines model replacement with projected gradient descent (PGD). Malicious clients scale their update so as to replace the global model with their local model during aggregation, while PGD keeps the update norm constrained to evade Euclidean-distance-based anomaly detection.

Adaptive attack. We additionally consider a scoring-aware variant of CBA that explicitly minimizes its own suspicion score (Appendix D.5).

Default threat setting. Of the $K = 50$ clients, 20% are malicious by default. Malicious clients relabel samples of class "0" as class "5" (with the trigger applied) to implant the backdoor.

C.4 Implementation and Evaluation Metrics

All experiments are implemented in PyTorch (Paszke et al., 2019). In each round the server samples half of the clients uniformly at random, and the global aggregation learning rate is set to 1.

We report three metrics. Attack Success Rate (ASR) is the fraction of trigger-stamped inputs classified as the attacker's target label; lower ASR means a more effective defense. Accuracy (ACC) is the accuracy on clean test inputs. Overall Performance Score (OPS), following Huang et al. (2023), combines both relative to FedAvg:

$$\mathrm{OPS} = \frac{\mathrm{DACC} - \mathrm{BACC}}{\mathrm{BACC}} - \frac{\mathrm{DASR} - \mathrm{BASR}}{\mathrm{BASR}}, \tag{28}$$

where DACC and DASR are the ACC and ASR of the evaluated defense, and BACC and BASR are those of FedAvg.

C.5 Baseline Defenses

We compare FedSHIELD with FedAvg (McMahan et al., 2017) and the following state-of-the-art defenses, grouped by whether they use server-side data:

- **No server-side data:** Krum (Blanchard et al., 2017), Median (Yin et al., 2018), RLR (Ozdayi et al., 2021), FoolsGold (Fung et al., 2020), MultiMetrics (Huang et al., 2023), FLAME (Nguyen et al., 2022), SnowBall (Qin et al., 2024) and FedDLAD (Ding et al., 2025).

- **Server-side (auxiliary) data:** FLTrust (Cao et al., 2021) and Sageflow (Park et al., 2021). FedSHIELD belongs to this group.

FedAvg is the undefended reference. All baselines were re-implemented and tuned following the recommendations of their original papers. A comparison within the server-data group under an identical data budget is given in Appendix D.6.

C.6 HYPERPARAMETERS

Table 4: Hyperparameters used for all main experiments.

Component	Setting
Number of clients K / participation per round	50 / 50% sampled uniformly
Communication rounds	200
Local epochs / batch size	10 / 256
Malicious clients / backdoor target	20% / label "0" $\rightarrow$ "5"
Local optimizer	SGD, learning rate 0.001
Global aggregation learning rate	1
Honeypot size	1 representative sample per class
Honeypot PGD (ℓ_∞, Alg. 1)	3 steps, step $1/255$, budget $\epsilon = 4/255$
Sensitivity threshold β (Eq. (13))	0.7
Update clipping norm	1
DP noise scale σ	$\{0.5,\ 0.75,\ 1.0\}$
Random seeds	$\{1,\ 42,\ 123\}$

D ADDITIONAL EXPERIMENTAL RESULTS

D.1 STATISTICAL ROBUSTNESS ACROSS RANDOM SEEDS

Table 5 reports FedSHIELD's main results as mean $\pm$ standard deviation over three seeds $\{1, 42, 123\}$, for all four datasets and both heterogeneity levels. Standard deviations are small everywhere (at most 1.0 percentage point in ACC), so the results in Table 1 are not an artifact of a particular seed.

Table 5: FedSHIELD results over three random seeds (mean $\pm$ std). ASR and ACC in %.

	CIFAR-10			**SVHN**			**FashionMNIST**			**MNIST**		
α	ASR↓	ACC↑	OPS↑	ASR↓	ACC↑	OPS↑	ASR↓	ACC↑	OPS↑	ASR↓	ACC↑	OPS↑
0.5	0.1±0.0	76.2±0.2	0.978±0.001	1.1±0.3	93.3±0.5	0.990±0.000	0.1±0.0	90.3±0.7	0.988±0.003	0.1±0.4	99.3±0.1	0.999±0.000
1.0	0.1±0.0	75.2±1.0	0.959±0.006	0.1±0.0	93.1±0.2	0.985±0.004	0.3±0.1	90.6±0.6	0.989±0.002	0.0±0.0	99.4±0.1	1.000±0.000

D.2 SEPARATION BETWEEN BENIGN AND MALICIOUS CLIENTS

This subsection examines the central premise of FedSHIELD (Section 4.7, Observation 1): the suspicion score s_k (Eq. (12)) separates malicious updates from benign ones.

Intuition. A malicious client optimizes the joint objective in Eq. (4), which minimizes the clean loss and the trigger-to-target loss at the same time. The honeypot batch is adversarially perturbed with PGD towards high-loss regions of the current global model (Algorithm 1). An update trained on the joint objective lowers the loss on these perturbed points by much more than an update trained only on the clean objective, and s_k measures exactly this excess reduction. The score depends only on how the update behaves on the honeypot, not on the update's norm, direction or similarity to other clients. It is therefore unaffected by the statistical camouflage that defeats norm- and similarity-based defenses. We verify this premise empirically below; a formal characterization is left to future work.

Setup. We record s_k for every participating client over 200 rounds (non-IID, $\alpha = 0.5$), with both honeypot sources described in Appendix C.2.

Across all four datasets and both honeypot sources, benign scores stay low (mean 0.02–0.08) and malicious scores stay high (mean 0.86–0.98), with AUROC ≥ 0.978 in every setting. Benign scores rise only slightly when the honeypot comes from a different dataset (e.g. $0.06 \rightarrow 0.08$ on CIFAR-10). Two conclusions follow. First, benign updates that improve generalization do not produce a large honeypot-loss reduction, even when the honeypot is close to the training distribution. Second, the separation comes from the backdoor objective itself rather than from overlap between the honeypot and the training data.

Table 6: Suspicion score s_k of benign vs. malicious clients (non-IID, $\alpha = 0.5$, 200 rounds) and the AUROC of s_k as a detector.

Dataset	Honeypot source	Client	Mean s_k	Std s_k	AUROC
CIFAR-10	Same-Dist.	Benign	0.06	0.03	0.987
		Malicious	0.91	0.05	
	Cross-Dist. (CIFAR-100)	Benign	0.08	0.04	0.978
		Malicious	0.86	0.07	
SVHN	Same-Dist.	Benign	0.04	0.02	0.992
		Malicious	0.97	0.04	
	Cross-Dist. (MNIST)	Benign	0.05	0.03	0.985
		Malicious	0.98	0.06	
FashionMNIST	Same-Dist.	Benign	0.03	0.02	0.995
		Malicious	0.97	0.03	
	Cross-Dist. (DeepFashion)	Benign	0.04	0.02	0.989
		Malicious	0.96	0.05	
MNIST	Same-Dist.	Benign	0.02	0.01	0.994
		Malicious	0.95	0.03	
	Cross-Dist. (SVHN)	Benign	0.03	0.02	0.991
		Malicious	0.92	0.04	

D.3 Filtering Error Rates

Table 7 reports the false-positive rate (FPR: benign clients removed) and the false-negative rate (FNR: malicious clients retained) of the filtering step (lines 8–10 of Algorithm 2), pooled over all filtering decisions made during the 200 training rounds.

Table 7: FPR and FNR (%) of the FedSHIELD filtering step, pooled over 200 rounds (non-IID, $\alpha = 0.5$).

Metric	CIFAR-10	SVHN	FashionMNIST	MNIST
FPR	0.80	0.60	0.01	0.00
FNR	0.10	0.80	0.03	0.00

FPR and FNR are each at most 0.8% on every dataset. The adaptive threshold τ is re-estimated each round from that round's cohort, yet it yields reliable filtering decisions throughout training.

D.4 Benign Clients Under Extreme Label Skew

A benign client whose local distribution is far from the global one could change the honeypot loss atypically and be mistaken for an attacker. To test this boundary case, we add a benign subgroup with extreme label skew (Dirichlet $\alpha = 0.1$, giving clients with 1–2 dominant classes) and compare its FPR with that of normal non-IID benign clients ($\alpha = 0.5$), at the deployed threshold $\beta = 0.7$.

Table 8: FPR of normal ($\alpha = 0.5$) and extreme-skew ($\alpha = 0.1$) benign clients, and FNR of malicious clients (%, pooled over 200 rounds).

Dataset	Dominant classes (skewed)	FPR, α=0.5	FPR, α=0.1	FNR
CIFAR-10	cat, dog	0.80	3.94	0.10
SVHN	2, 9	0.60	2.43	0.80
FashionMNIST	Coat, Bag	0.01	1.98	0.03
MNIST	5, 8	0.00	1.93	0.00

Extreme skew raises the FPR on every dataset, to at most 3.94% on CIFAR-10, so this is a real boundary condition of the adaptive threshold. Even so, more than 96% of extreme-skew benign clients are retained and the FNR is unchanged. The effect does not appear in the population-level statistics of Table 7 because such clients are a small minority. A relaxed threshold for clients with unusually narrow class support could close this gap; we leave this to future work and note it in Appendix G.

D.5 Robustness to an Adaptive, Scoring-Aware Attacker

We consider an attacker that knows the FedSHIELD scoring rule and explicitly tries to keep its suspicion score low while injecting the backdoor. The server-side honeypot is not visible to clients, so the attacker builds a proxy honeypot $\hat{\mathcal{H}}$ from its own clean data and adds an evasion penalty to the CBA objective (Eq. (4)):

$$\min_{\theta} \; \mathbb{E}_{(x,y)\sim D_{\text{clean}}} \ell(M_\theta(x), y) + \lambda \, \mathbb{E}_{(x',y_t)\sim D_{\text{poison}}} \ell(M_\theta(x'), y_t) + \gamma \cdot \max\big(0, \; \hat{L}_{ref} - \ell(M_\theta; \hat{\mathcal{H}})\big), \tag{29}$$

where γ sets the strength of the evasion term; $\gamma = 0$ recovers the standard CBA attack.

Table 9: FedSHIELD against the adaptive attacker of Eq. (29) (non-IID, $\alpha = 0.5$). ASR and ACC in %.

Dataset	Defense	γ	ASR↓	ACC↑	Mean s_k (malicious)
CIFAR-10	FedAvg	0.0	85.4	77.8	–
	FedSHIELD	0.0	0.1	76.2	0.91
	FedSHIELD	1.0	1.8	75.9	0.82
	FedSHIELD	2.0	4.1	75.5	0.71
SVHN	FedAvg	0.0	81.0	91.8	–
	FedSHIELD	0.0	1.1	93.3	0.93
	FedSHIELD	1.0	2.1	83.9	0.89
	FedSHIELD	2.0	4.7	83.5	0.83
FashionMNIST	FedAvg	0.0	95.4	91.3	–
	FedSHIELD	0.0	0.1	90.3	0.94
	FedSHIELD	1.0	1.9	88.3	0.86
	FedSHIELD	2.0	2.4	88.0	0.78
MNIST	FedAvg	0.0	100.0	99.3	–
	FedSHIELD	0.0	0.1	99.3	0.95
	FedSHIELD	1.0	0.6	99.2	0.89
	FedSHIELD	2.0	1.3	99.1	0.81

A larger γ lowers the attackers' suspicion score and raises ASR, so the evasion term has a measurable effect. Even at $\gamma = 2$, however, ASR stays at or below 4.7% on all datasets, against 81–100% without defense. The malicious mean s_k (0.71–0.83) also remains far above the benign range in Table 6 (0.02–0.08): the detection signal is weakened but not defeated. Clean accuracy is essentially unchanged on CIFAR-10, FashionMNIST and MNIST. The attacker considered here does not observe the server's honeypot; stronger attackers remain future work.

D.6 Comparison Under a Matched Auxiliary-Data Budget

FedSHIELD, FLTrust and Sageflow all use a small server-side dataset, while the other baselines use none (Appendix C.5). To isolate the effect of the mechanism from the effect of the data, we compare the three auxiliary-data defenses under an identical budget of one clean sample per class. This is FedSHIELD's default setting.

Table 10: Auxiliary-data defenses under a matched budget of 1 sample/class (non-IID, $\alpha = 0.5$). ASR and ACC in %.

Dataset	Defense	ASR↓	ACC↑	OPS↑
CIFAR-10	FedAvg (no defense)	85.4	77.8	0.000
	FLTrust	41.0	77.5	0.516
	Sageflow	51.5	73.9	0.347
	FedSHIELD	0.1	76.1	0.978
MNIST	FedAvg (no defense)	100.0	99.3	0.000
	FLTrust	19.2	99.2	0.807
	Sageflow	25.3	99.1	0.745
	FedSHIELD	0.1	99.3	0.999

With the same auxiliary resource, FedSHIELD lowers ASR to 0.1%, compared with 41–52% (CIFAR-10) and 19–25% (MNIST) for FLTrust and Sageflow, at comparable clean accuracy.

Why the same data are used more effectively. FLTrust uses the auxiliary data to compute a server reference update and weights each client by the cosine similarity of its update to that reference. Sageflow evaluates each client model on the auxiliary data, filters out models with high output entropy, and weights the rest by their loss (loss-weighted averaging). In both cases, a backdoored update that also fits the clean task well looks benign: it points in a direction similar to the reference, and its loss on clean data is low. FedSHIELD instead uses the data to probe the behavior that the backdoor objective adds. The PGD-perturbed honeypot lies in high-loss regions of the current model, and an update trained on the joint clean-plus-trigger objective (Eq. (4)) lowers the loss there by much more than a benign update (Table 6). FedSHIELD therefore flags updates that are too good on the perturbed probe, whereas FLTrust and Sageflow reject updates that are too dissimilar or too poor on clean data. Stealthy backdoors are designed to avoid exactly the latter. Both baselines were originally designed for larger auxiliary sets: FLTrust uses a root dataset of 100 samples by default, and Sageflow specifies its public data as a fraction of the training set (1–2%). Both are much larger than one sample per class, so our matched setting is stricter for these baselines than their native operating points.

Practicality of the auxiliary-data assumption. (i) The budget is tiny: one sample per class, i.e. 10 images for a 10-class task. (ii) The data need not match the clients' distribution: separation remains strong with cross-dataset honeypots (AUROC ≥ 0.978, Table 6), so generic public images of the right semantic categories suffice. (iii) Assuming a small clean server dataset is standard among comparably motivated defenses (FLTrust, Sageflow).

D.7 Time Complexity of Different Methods Across Clients

Table 11 reports the computation time of various FL aggregation methods for client sizes ranging from 5 to 1000. FedAvg consistently exhibits the lowest time complexity, demonstrating excellent efficiency and scalability. Methods such as RLR and MultiMetrics achieve slightly higher computation times but remain much faster than robust defenses like Krum, whose time grows sharply with the number of clients. FLTrust and FoolsGold show moderate times for small client sets but scale poorly as client numbers increase. Notably, FedSHIELD maintains low time complexity and scales efficiently to large client sizes: at 1000 clients it is faster than most robust defenses (e.g. FLAME, Median, FLTrust and Krum) and comparable to SnowBall and FedDLAD, while still providing strong defense performance.

Table 11: Running time (s) of different defense methods, where c represents the number of clients.

Method	c5	c10	c25	c50	c100	c1000
FedAvg	0.0001±0.0003	0.0001±0.0000	0.0001±0.0001	0.0013±0.0005	0.0065±0.0060	0.0410±0.0379
RLR	0.0006±0.0005	0.0005±0.0010	0.0007±0.0003	0.0019±0.0003	0.0262±0.0168	0.0516±0.0497
MultiMetrics	0.0008±0.0005	0.0016±0.0018	0.0028±0.0044	0.0063±0.0008	0.0130±0.0019	0.2860±0.2164
FedSHIELD	0.0012±0.0006	0.0016±0.0047	0.0055±0.0065	0.0089±0.0003	0.0111±0.0026	0.1962±0.0483
SnowBall	0.0007±0.0005	0.0016±0.0048	0.0032±0.0063	0.0061±0.0005	0.0120±0.0011	0.1521±0.0167
FedDLAD	0.0009±0.0005	0.0016±0.0048	0.0032±0.0063	0.0067±0.0010	0.0128±0.0013	0.2001±0.0481
FLAME	0.0017±0.0005	0.0031±0.0052	0.0079±0.0079	0.0142±0.0010	0.0273±0.0015	0.4378±0.0775
Median	0.0017±0.0006	0.0058±0.0064	0.0095±0.0078	0.0165±0.0006	0.0763±0.0203	0.5503±0.0652
FLTrust	0.0047±0.0011	0.0072±0.0068	0.0165±0.0086	0.0288±0.0011	0.0622±0.0065	1.3700±0.9231
FoolsGold	0.0015±0.0009	0.0169±0.0279	0.0057±0.0061	0.0079±0.0003	0.0198±0.0015	0.6343±0.1776
Krum	0.0013±0.0007	0.0065±0.0061	0.0205±0.0064	0.0736±0.0028	0.3548±0.0304	40.230±5.1113

E Privacy and Theoretical Assumptions

E.1 Privacy Accounting

After filtering, each retained update is clipped to $\|\tilde{\Delta}_k\|_2 \leq 1$ and Gaussian noise $\mathcal{N}(0, \sigma^2 I)$ is added to their mean (Algorithm 2, lines 11–15).

Assumption. We analyze the Gaussian mechanism conditional on the retained set $\mathcal{S}_{keep}$, i.e. we treat the filtering decision as fixed. Under this assumption, replacing one client's update changes the mean by at most $\Delta_2 = 1/|\mathcal{S}_{keep}|$. We use the smallest retained-set size observed in any round, $|\mathcal{S}_{keep}| = 12$, as a worst case, together with $\delta = 10^{-5}$. We stress that the filtering step is itself data-dependent. A fully unconditional guarantee would also have to account for the effect of one client on the threshold τ and hence on $\mathcal{S}_{keep}$. We do not claim such a guarantee.

Per-round and composed guarantees. The per-round guarantee follows from the classical Gaussian mechanism, $\varepsilon = (\Delta_2/\sigma)\sqrt{2\ln(1.25/\delta)}$. For all $T = 200$ rounds, we compose with Rényi DP (Mironov, 2017): the Gaussian mechanism with noise multiplier $z = \sigma/\Delta_2$ is $(\alpha, \alpha/2z^2)$-RDP, the RDP guarantees add over T rounds, and we convert to (ε, δ)-DP via $\varepsilon = \min_{\alpha>1} T\alpha/(2z^2) + \ln(1/\delta)/(\alpha - 1)$. We apply *no* privacy amplification from the 50% client sampling, so the composed values are conservative.

Table 12: Client-level privacy guarantees, conditional on the retained set ($|\mathcal{S}_{keep}| \geq 12$, $\delta = 10^{-5}$).

σ	Noise multiplier z	Per-round ε	ε after 200 rounds (RDP, no amplification)
0.50	6	0.81	14.1
0.75	9	0.54	8.8
1.00	12	0.40	6.4

E.2 Proof of Proposition 1 and Assumptions of Eq. (27)

Proof of Proposition 1. Let Δ_{ben} and Δ_{mal} denote the means of the retained benign and malicious clipped updates. Then $\Delta^{agg} = (1 - \rho_t)\Delta_{ben} + \rho_t\Delta_{mal} = \Delta_{ben} + \rho_t(\Delta_{mal} - \Delta_{ben})$. Since every clipped update has norm at most 1, so do both means, hence $\|\Delta_{mal} - \Delta_{ben}\| \leq 2$. We can therefore write $\Delta^{agg} = -\eta g^t$ with $g^t = \nabla F(\theta^t) + \tilde{b}^t + \xi^t$ and $\|\tilde{b}^t\| \leq B + 2\rho/\eta =: B'$. The server step is $\theta^{t+1} = \theta^t - \eta g^t + n^t$ with $n^t \sim \mathcal{N}(0, \sigma^2 I_d)$ independent of everything else. By L-smoothness,

$$\mathbb{E}F(\theta^{t+1}) \leq F(\theta^t) - \eta\,\mathbb{E}\langle\nabla F, g^t\rangle + \tfrac{L\eta^2}{2}\mathbb{E}\|g^t\|^2 + \tfrac{Ld\sigma^2}{2}.$$

Using $\mathbb{E}[\xi^t \mid \theta^t] = 0$ and $\langle a, b\rangle \geq -\frac{1}{2}\|a\|^2 - \frac{1}{2}\|b\|^2$, $\mathbb{E}\langle\nabla F, g^t\rangle \geq \frac{1}{2}\|\nabla F\|^2 - \frac{1}{2}B'^2$; and $\mathbb{E}\|g^t\|^2 \leq 3\|\nabla F\|^2 + 3B'^2 + 3V$. With $\eta \leq 1/(6L)$, we have $\frac{3L\eta^2}{2} \leq \frac{\eta}{4}$, so

$$\mathbb{E}F(\theta^{t+1}) \leq F(\theta^t) - \tfrac{\eta}{4}\|\nabla F(\theta^t)\|^2 + \tfrac{3\eta}{4}B'^2 + \tfrac{3L\eta^2}{2}V + \tfrac{Ld\sigma^2}{2}.$$

Summing over $t = 0, \ldots, T-1$, using $F(\theta^T) \geq F^*$ and multiplying by $4/(\eta T)$ gives the claim. □

The assumption on Δ_{ben} is standard for FedAvg-type methods (Li et al., 2020; Karimireddy et al., 2020): b^t collects the bias from multiple local steps under non-IID data and from clipping, and ξ^t the stochastic-gradient noise.

Backdoor suppression (Eq. (27)). The bound assumes (B1) a finite per-update backdoor contribution $B_{\max}$, and (B2) that $|\mathcal{S}_{keep} \cap \mathcal{S}_{mal}|$, the number of malicious clients that survive filtering, is small. (B2) is an empirical premise. It is supported by the measured FNR of at most 0.8% (Table 7) and by the low ASR ($\leq 4.7\%$) under the adaptive attacker (Appendix D.5).

F Broader Impact

FedSHIELD enhances the security of FL in sensitive domains such as healthcare and finance by mitigating backdoor attacks without accessing private client data. This promotes trust and can accelerate the adoption of privacy-preserving, collaborative AI. Behavioral screening, however, carries inherent risks. Wrongly filtering benign clients (false positives), particularly under highly heterogeneous data, could exclude diverse data perspectives and degrade model fairness or utility for underrepresented groups. We quantify this risk in Appendix D.4: clients with extremely narrow class support are filtered at a rate of up to 3.9%. In addition, server-side model reconstruction adds computational overhead and a larger energy footprint compared with standard aggregation such as FedAvg.

G Limitations

Although FedSHIELD performs strongly across diverse FL scenarios, the sensitivity parameter β of the adaptive threshold may require tuning across FL configurations to balance false positives and false negatives. Benign clients with extremely narrow class support are filtered slightly more often (FPR up to 3.9%, Appendix D.4), and FedSHIELD requires a small amount of public data from the task's categories. Finally, extending FedSHIELD to asynchronous settings, such as clients dropping out mid-round or joining asynchronously, remains an interesting direction for future work.