

Article

SecureFedGuard: Authenticated and Backdoor-Resilient Federated Learning with Dual-View Gradient Forensics

Tuli Chen ¹, Yantao Li ² and Shu Gong ^{1,*}¹ School of Management, Guangdong University of Science and Technology, Dongguan 523070, China² Faculty of Data Science, City University of Macau, Taipa, Macau

* Correspondence: gongshu@gdust.edu.cn

Abstract

Federated learning (FL) enables collaborative model training without centralizing raw data, yet practical deployments remain vulnerable to security threats such as Byzantine model poisoning, stealthy backdoor implantation, and integrity attacks that exploit the opacity of client updates. This paper presents SecureFedGuard, a security-centric FL framework that introduces a novel combination of (i) dual-view update authentication that binds each client update to a lightweight stochastic gradient fingerprint, enabling server-side integrity screening without accessing client data, and (ii) backdoor-resilient aggregation driven by cross-round spectral forensics and adaptive coordinate-wise trimming guided by an estimated benign subspace. SecureFedGuard is designed to be compatible with secure aggregation and does not require trusted hardware, public datasets for pretraining, or expensive per-client verification. We provide a simple robustness analysis that clarifies when benign updates dominate the estimated subspace under mixed benign/malicious participation. Experiments on real FL benchmarks (vision and language) under diverse threat models show that SecureFedGuard substantially improves clean accuracy and backdoor attack success rate compared with strong baselines, while adding modest communication and computation overhead. These results suggest a practical path toward integrity-preserving and backdoor-resistant FL without weakening the privacy boundary between clients and the server.

Keywords: machinelearning; backdoor attack; spiking neural network

1. Introduction

Federated learning (FL) enables many clients (e.g., mobile devices or organizations) to collaboratively train a shared model without centralizing raw training data [1,2]. This design mitigates direct data exposure, and is often combined with cryptographic secure aggregation so that the server observes only aggregated updates rather than individual client deltas [3]. Despite these advantages, practical FL deployments remain highly vulnerable to security threats: malicious clients can inject Byzantine updates that derail training, or implant stealthy backdoors that preserve clean utility while enforcing attacker-chosen behaviors at inference time [4–6]. The core difficulty is that the server must coordinate learning from updates it cannot fully trust, under data heterogeneity and limited observability.

A large body of work addresses Byzantine robustness by replacing naïve averaging with robust aggregation rules, such as coordinate-wise robust estimators and distance-based selection mechanisms [4,5,7]. However, recent evidence suggests that (i) non-IID client distributions and adaptive attackers can degrade the effectiveness of classic robust rules,



Academic Editor: Aryya Gangopadhyay

Received: 22 January 2026

Revised: 12 February 2026

Accepted: 13 February 2026

Published: 28 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

and (ii) designing yet another aggregation heuristic may be insufficient without additional security signals [8–10]. In parallel, backdoor attacks in FL have become increasingly stealthy by concentrating poisoning into critical layers or carefully scaling updates (model replacement), which can bypass both naïve defenses and several robust aggregators [11]. Existing backdoor defenses often assume extra capabilities (e.g., clean reference data, strong client-side validation, or plaintext access to all updates), or operate myopically per round and thus struggle against persistent, cross-round attackers [12–15].

This paper targets an increasingly relevant deployment regime: FL systems that must remain privacy-preserving, scalable, and secure simultaneously. Privacy mechanisms such as secure aggregation can prevent the server from inspecting individual updates, which complicates traditional outlier filtering; meanwhile, integrity threats demand that malicious contributions be detected or attenuated without violating the privacy boundary [3,16,17]. Moreover, backdoor defenses must keep clean accuracy high under heterogeneous data, while suppressing targeted misbehavior under strong, coordinated attackers [6,18,19]. These constraints motivate a defense that goes beyond a single robust estimator and instead couples lightweight integrity screening with geometry-aware, cross-round anomaly suppression.

Our approach. We propose SecureFedGuard, a security-centric FL framework that combines two orthogonal mechanisms. First, we introduce dual-view update authentication (DVA), a lightweight trajectory-consistency test based on compact linear sketches: each client transmits a sketch of its final update and a sketch of its cumulative local gradient trace. The server uses these two views to detect integrity-evasive attacks that craft arbitrary vectors without following a plausible local optimization path. Second, to counter optimization-based poisoning and stealthy backdoors that may pass such consistency checks, we introduce cross-round spectral forensics with residual clipping: the server estimates a benign subspace each round, measures residual energies of client updates outside that subspace, and maintains a cross-round persistence memory to down-weight clients that repeatedly deviate. These signals are then integrated into an adaptive robust aggregation rule that preserves benign learning while suppressing persistent anomalous directions.

Key contributions. SecureFedGuard contributes: (1) a new sketch-based integrity screening mechanism (DVA) that is data-free and adds only low-bandwidth overhead, addressing integrity-evasive Byzantine behaviors; (2) a cross-round geometric defense that sanitizes updates by clipping only the suspicious residual component, improving robustness to stealthy backdoors and coordinated poisoning; and (3) a design that is compatible with secure aggregation by operating on sketches and scalar weights, aligning privacy requirements with integrity needs [3,16,17]. Empirically, SecureFedGuard substantially reduces backdoor attack success while maintaining clean accuracy across vision and language FL benchmarks, outperforming strong baselines including robust aggregators and recent backdoor defenses [7,12,15].

2. Related Work

2.1. Byzantine Robustness and Robust Aggregation

The standard cross-device FL pipeline popularized by FedAvg aggregates client updates by averaging, which is efficient but brittle when a subset of clients is malicious or faulty [1,2]. Byzantine-robust learning therefore studies aggregation rules that tolerate adversarial updates while preserving convergence. Early work introduced principled robust estimators for distributed gradients, motivating distance-based selection and coordinate-wise robust summaries [4,5]. In FL, robust aggregation via the geometric median (often referred to as robust federated aggregation) provides strong empirical robustness and can be implemented efficiently with Weiszfeld-type iterations [7].

Recent research highlights that practical FL security requires robustness under non-IID client data and potentially unknown corruption rates. Methods that incorporate auxiliary signals (e.g., public-data-based screening) can improve robustness in heterogeneous settings but may introduce assumptions that do not always hold [8]. Complementary directions leverage incentives or ensembles to reduce the effective influence of low-quality or adversarial participants [20]. In decentralized variants, aggregation must also tolerate dynamic topologies and local neighbor corruption, leading to new robust filters designed beyond the centralized-server setting [21]. At the same time, the security community has emphasized that strengthening classic aggregators (e.g., trimmed mean/median) with additional mechanisms can outperform designing ever more complex rules in some regimes, and provides new threat models and evaluation lenses [9,10]. These trends motivate our design choice to retain robust, scalable primitives while adding orthogonal signals that are rare in prior work: (i) trajectory-consistency checks that test whether an update is realizable by a plausible local optimization trajectory, and (ii) cross-round persistence that accumulates evidence over time rather than relying on a single-round outlier detector.

2.2. Backdoor Attacks and Defenses in Federated Learning

Backdoor (targeted poisoning) attacks in FL aim to implant a trigger-target behavior while maintaining high clean accuracy, making them harder to detect than untargeted Byzantine disruptions. Modern backdoor attacks can concentrate poisoning into model subsets or layers to improve stealthiness; for example, poisoning backdoor-critical layers can bypass multiple defenses with a small malicious fraction [11]. Defenses therefore increasingly exploit non-trivial structure beyond update magnitude, including hidden-layer behaviors, directional patterns, or parameter-importance disparities.

A representative line of work uses client-side validation signals or neuron-level statistics to detect poisoned updates, exemplified by CrowdGuard, which introduces feedback-driven inspection and pruning mechanisms [12]. Other defenses attempt to distinguish benign heterogeneity from malicious triggers via parameter-importance signals; FDCR uses Fisher-information-based discrepancies to cluster and rescale client updates under heterogeneous distributions [13]. Complementary approaches focus on model purification: FLSAD eliminates backdoor influence via trigger reconstruction and self-attention distillation without assuming a clean reference dataset [14]. Lightweight client-side modifications can also reduce backdoor success by patching or neutralizing trigger effects during local training [22]. In 2025, direction-based screening gained traction; AlignIns inspects multi-granularity directional alignment and couples filtering with clipping to resist stealthy backdoors under non-IID data [15]. Beyond standard FL, the literature also expands to vertical/split settings; UBD synthesizes latent trigger cues and uses label-consistent clustering with constrained probing to mitigate VFL backdoors [19]. Specialized domains such as federated graph learning introduce additional attack surfaces (diverse trigger structures and injection locations), prompting defenses like FedTGE that use topological graph energy for selection and reweighting [18]. Surveys summarize the rapidly growing landscape and expose recurring gaps between benchmark robustness and deployment constraints [6]. Overall, existing defenses often either (i) require extra assumptions (clean validation data, access to plaintext updates, or trusted components) or (ii) treat rounds largely independently using one-round statistics. As a result, many methods do not explicitly verify whether an update is realizable by a plausible local training trajectory, and they do not explicitly exploit temporal persistence of anomalous directions across rounds. SecureFedGuard is designed to fill both gaps via DVA (realizability) and persistence-aware spectral forensics (temporal evidence).

2.3. Secure Aggregation, Privacy, and Integrity Verification

Secure aggregation is a key primitive for FL privacy: the server learns only an aggregate of client updates, not individual contributions [3]. However, privacy can hinder many server-side defenses because robust outlier detection typically assumes access to individual updates in plaintext. This tension has spurred work on secure aggregation protocols with stronger dropout tolerance and cryptographic efficiency, including homomorphic-encryption-based constructions and variants tailored to practical deployment constraints [17]. In parallel, privacy-preserving robust aggregation has emerged to mitigate poisoning while maintaining confidentiality, e.g., PEAR performs similarity-based weighting under encrypted gradients to address Byzantine threats in realistic non-IID scenarios [16].

A central limitation in this space is that privacy alone does not guarantee integrity: malicious clients may still contribute adversarial updates, and an untrusted server could potentially deviate from the intended aggregation procedure. Recent overviews emphasize that robust FL must account for Sybil behavior, secure-aggregation interactions, and adaptive backdoor threats, motivating defenses that introduce verification or auxiliary signals with minimal privacy leakage [10]. SecureFedGuard follows this direction: it is designed to remain compatible with secure aggregation by relying on compact sketches and low-dimensional statistics for screening, while retaining a robust aggregation core for scalability and non-IID tolerance.

3. Preliminaries

3.1. Federated Learning Setup and Notation

We consider cross-device federated learning with a central server coordinating N clients indexed by $i \in \{1, \dots, N\}$. Client i holds a private dataset $\mathcal{D}_i$ drawn from a client-specific distribution. Let $w \in \mathbb{R}^d$ denote the model parameters and $\ell(w; z)$ the per-example loss for data point z . The global objective is

$$\min_{w \in \mathbb{R}^d} F(w) \triangleq \sum_{i=1}^N p_i F_i(w), \quad F_i(w) \triangleq \mathbb{E}_{z \sim \mathcal{D}_i} [\ell(w; z)], \quad (1)$$

where $p_i \geq 0$ and $\sum_i p_i = 1$ are client weights (typically proportional to dataset size).

Training proceeds in communication rounds $t = 0, 1, \dots, T - 1$. At round t , the server holds the global model w_t and samples a subset of clients $\mathcal{S}_t \subseteq \{1, \dots, N\}$ with $|\mathcal{S}_t| = K$. Each selected client initializes local parameters $w_{i,t}^{(0)} \leftarrow w_t$ and performs E steps of a first-order optimizer (e.g., mini-batch SGD):

$$w_{i,t}^{(e+1)} = w_{i,t}^{(e)} - \eta_t g_{i,t}^{(e)}, \quad g_{i,t}^{(e)} \triangleq \text{OptDir}(w_{i,t}^{(e)}; \xi_{i,t}^{(e)}), \quad (2)$$

where η_t is the learning rate and $\xi_{i,t}^{(e)}$ is a mini-batch sampled from $\mathcal{D}_i$. Here, $\text{OptDir}(\cdot)$ denotes the first-order step direction applied by the local optimizer (e.g., $\nabla \ell$ for plain SGD or the momentum velocity for SGD with momentum). The client sends an update vector

$$u_i^t \triangleq w_{i,t}^{(E)} - w_t \in \mathbb{R}^d, \quad (3)$$

(or equivalently the post-training model $w_{i,t}^{(E)}$). The server aggregates received updates to form the next global model:

$$w_{t+1} = w_t + \text{Agg}(\{u_i^t\}_{i \in \mathcal{S}_t}), \quad (4)$$

where $\text{Agg}(\cdot)$ is an aggregation rule. For convenience, we denote the matrix of stacked updates by $U_t \in \mathbb{R}^{K \times d}$ whose i -th row corresponds to $(u_i^t)^\top$ (with an arbitrary but fixed ordering of clients in $\mathcal{S}_t$). We use $\|\cdot\|$ for the ℓ_2 norm and $\langle a, b \rangle$ for the standard inner product.

3.2. Adversary Model and Security Objectives

We assume an open FL setting in which a subset of participating clients may be malicious. Let $\mathcal{A}_t \subseteq \mathcal{S}_t$ denote the set of adversarial clients at round t , and let $\mathcal{B}_t = \mathcal{S}_t \setminus \mathcal{A}_t$ denote benign clients. We consider a strong adversary that can (i) control any local training procedure, (ii) craft arbitrary vectors u_i^t to send to the server, and (iii) coordinate across compromised clients and across rounds. We bound the per-round corruption rate by

$$|\mathcal{A}_t| \leq \alpha K, \quad \alpha \in [0, 1). \quad (5)$$

The adversary may aim for

- **Byzantine model poisoning:** degrade overall model utility by sending arbitrary or optimized updates.
- **Backdoor attacks:** enforce a targeted misclassification behavior triggered by a pattern while maintaining high clean accuracy.
- **Integrity evasion:** craft updates that appear statistically similar to benign ones under common defenses.

To align with practical deployments, we do not assume the server can access raw client data. Our defense targets two coupled objectives: **(i) update integrity**—detect or down-weight malicious updates without inspecting $\mathcal{D}_i$; and **(ii) backdoor resilience**—maintain high clean performance while suppressing attack success.

We evaluate learning quality by Clean Accuracy on standard test sets. Backdoor robustness is measured by Attack Success Rate (ASR), defined as the fraction of triggered test inputs classified into the adversary's target label. Lower ASR at similar clean accuracy indicates stronger backdoor resistance.

Threat boundaries. SecureFedGuard provides statistical screening rather than cryptographic integrity guarantees. DVA is most effective when an update is not produced by the claimed local SGD trajectory (e.g., arbitrary vectors or in-transit tampering). A fully adaptive attacker who can fabricate both views to satisfy $f_i^t \approx -\eta_t h_i^t$ may bypass DVA; in that case, robustness relies on cross-round forensics and robust aggregation. Spectral forensics can be weakened if an attack direction lies largely inside the estimated benign subspace or if the attacker poisons very intermittently, typically trading off ASR or requiring a higher attacker budget. We clarify these defensive boundaries in Section 4.4.

3.3. Robust Aggregation and Spectral Primitives

Robust FL defenses often rely on estimating a representative update direction in the presence of outliers. We briefly summarize two primitives used throughout the paper.

Coordinate-wise robust summaries. Given vectors $\{u_i\}_{i=1}^K \subset \mathbb{R}^d$, coordinate-wise trimmed mean removes a fraction of extreme values in each coordinate and averages the remainder. Let $u_{i,j}$ denote coordinate j . For a trimming level $\rho \in [0, 0.5)$, define $\mathcal{I}_j$ as the indices after removing the largest $\lceil \rho K \rceil$ and smallest $\lceil \rho K \rceil$ values among $\{u_{i,j}\}_{i=1}^K$, and set

$$\text{TrimMean}_\rho(\{u_i\})_j \triangleq \frac{1}{|\mathcal{I}_j|} \sum_{i \in \mathcal{I}_j} u_{i,j}. \quad (6)$$

This operation is efficient and provides robustness when the fraction of corrupted entries is bounded.

Benign subspace estimation. Many poisoning strategies introduce update directions that deviate from the dominant benign variation. Let $\bar{u}_t$ denote a center estimate (e.g., coordinate-wise median or trimmed mean) of $\{u_i^t\}_{i \in \mathcal{S}_t}$. Define centered updates $\tilde{u}_i^t \triangleq u_i^t - \bar{u}_t$ and the empirical covariance

$$C_t \triangleq \frac{1}{K} \sum_{i \in \mathcal{S}_t} \tilde{u}_i^t (\tilde{u}_i^t)^\top \in \mathbb{R}^{d \times d}. \quad (7)$$

Let $V_t \in \mathbb{R}^{d \times r}$ contain the top- r eigenvectors of C_t (orthonormal columns), defining the rank- r projector $P_t \triangleq V_t V_t^\top$. For any update u_i^t , we define its **spectral residual energy** as

$$\text{Res}(u_i^t; P_t) \triangleq \|(I - P_t)(u_i^t - \bar{u}_t)\|^2, \quad (8)$$

which measures deviation from the estimated principal subspace. In later sections, we use residual statistics across rounds to flag persistent anomalous directions and to adapt the trimming strength in aggregation.

Lightweight sketching and fingerprints. To support integrity screening with minimal overhead, we use a randomized sketch map $\phi: \mathbb{R}^d \rightarrow \mathbb{R}^m$ with $m \ll d$. A common choice is a sign-random projection

$$\phi(u) \triangleq \frac{1}{\sqrt{m}} R u, \quad R \in \{-1, +1\}^{m \times d}, \quad (9)$$

generated from a public seed. Such sketches approximately preserve norms and inner products when m is sufficiently large. We will employ $\phi(\cdot)$ to form compact gradient fingerprints that are inexpensive to transmit and compare, and that can be combined with robust statistics without revealing raw training data.

4. Methodology

4.1. Design Goals and End-to-End Protocol

SecureFedGuard is built for the practical FL regime where the server must learn from untrusted updates while respecting the privacy boundary implied by local data isolation and, optionally, secure aggregation. The methodology is guided by three design goals. (G1) Integrity without data access: the server should reject or down-weight obviously non-realizable updates without inspecting any client data $\mathcal{D}_i$. (G2) Backdoor resilience under non-IID: defenses must not confuse benign heterogeneity with attacks, and must suppress targeted triggers while maintaining high clean accuracy. (G3) Deployment compatibility: the defense should remain compatible with secure aggregation and incur modest overhead.

To meet these goals, SecureFedGuard uses two orthogonal security signals, each addressing a different failure mode of robust aggregation. First, trajectory consistency checks whether a reported update u_i^t is consistent with a plausible local optimization trace, which is effective against integrity-evasive attacks that craft arbitrary vectors. Second, geometric persistence tracks whether a client repeatedly deviates from the dominant benign update geometry across rounds, which is effective against optimization-based poisoning and stealthy backdoors that can mimic local SGD dynamics in a single round.

Protocol overview. At the beginning of round t , the server broadcasts the current model w_t and a public seed that determines a linear sketch map $\phi(\cdot)$ (Equation (9)) shared by all participants. Each selected client $i \in \mathcal{S}_t$ performs local training starting from w_t and produces the model delta u_i^t . In addition to u_i^t , the client computes and transmits two compact fingerprints: an update sketch $f_i^t = \phi(u_i^t)$ and a cumulative gradient-trace sketch $h_i^t = \sum_{e=0}^{E-1} \phi(g_{i,t}^{(e)})$. These sketches are low-dimensional ($m \ll d$) and are designed to be inexpensive, while still enabling meaningful consistency checks at the server.

On the server, SecureFedGuard proceeds in three stages (Sections 4.2–4.4). First, it computes a DVA gate $a_i^t \in (0, 1]$ from the discrepancy between f_i^t and $-\eta_t h_i^t$, producing a soft integrity score per client. Second, it performs cross-round spectral forensics: using the current set of updates, it estimates a robust center $\bar{u}_t$ and a low-rank benign subspace projector P_t , measures the residual energy $r_i^t = \|(I - P_t)(u_i^t - \bar{u}_t)\|^2$, and updates a persistence memory M_t^t via an EMA. These quantities determine a residual shrink factor λ_i^t and a sanitized update $\hat{u}_i^t$ that preserves within-subspace learning while attenuating anomalous residual components. Third, SecureFedGuard aggregates sanitized updates with a **security-aware robust rule**: it combines DVA and persistence into a scalar weight ω_i^t and applies an adaptive, coordinate-wise trimmed mean whose trimming level depends on an estimated corruption indicator $\hat{\alpha}_t$, yielding the global update Δ_t and $w_{t+1} = w_t + \Delta_t$.

Secure-aggregation (SA) mode. In cross-device deployments, we consider a standard secure aggregation protocol in which the server learns only an aggregate of masked updates. SecureFedGuard operates in two phases each round. First, each selected client sends plaintext sketches (f_i^t, h_i^t) (Equation (10)); the server computes DVA gates and sketch-space forensics scores and then returns a per-client scalar multiplier s_i^t (and optionally an allowlist) over an authenticated channel. Second, only allowlisted clients participate in secure aggregation and contribute the masked, locally scaled update $s_i^t u_i^t$, so the server observes only $\sum_i s_i^t u_i^t$ and updates with a weighted mean. Dropout is handled by the underlying secure aggregation protocol: if a client drops after sending sketches, its contribution is absent from the final aggregate and the server normalizes using the surviving set. Algorithm 1 provides the concrete message flow and threat assumptions.

Leakage from sketches. The server observes per-client m -dimensional randomized linear projections of both the final update and the gradient trace. These sketches can leak coarse information about the update direction and norm but are substantially lower-dimensional than u_i^t ; we treat them as defense metadata rather than a privacy mechanism. If stronger privacy guarantees are required, sketches can be noise-perturbed or securely aggregated as well, which is complementary and beyond the scope of this paper.

Deployment notes. In secure aggregation, the server does not observe per-client u_i^t , so coordinate-wise trimming and residual-only clipping in update space cannot be applied on plaintext updates. In SA mode, we therefore (i) compute DVA and persistence weights from plaintext sketches, (ii) optionally remove clients with very small weights before aggregation, and (iii) apply a per-client scalar shrink s_i^t (derived from sketch residual energy and persistence) that clients apply locally to their update before masking, yielding a weighted-mean secure-aggregation update. If encrypted robust aggregation (e.g., trimmed mean under MPC/HE) is required, it is complementary to SecureFedGuard and not assumed here.

In the remainder of this section, we reorganize the methodology into three technical subsections: (i) DVA and its scoring/tuning, (ii) cross-round spectral forensics with residual clipping and persistence memory, and (iii) the final security-aware robust aggregation and protocol summary.

Algorithm 1 SecureFedGuard with secure aggregation (SA mode)

- 1: Server selects clients $\mathcal{S}_t$ and broadcasts $(w_t, \eta_t, \text{seed}_t)$ and secure-aggregation parameters.
- 2: **for** each client $i \in \mathcal{S}_t$ (in parallel) **do**
- 3: Client runs local training from w_t to obtain u_i^t and per-step gradients $\{g_{i,t}^{(e)}\}_{e=0}^{E-1}$.
- 4: Client computes plaintext sketches $f_i^t = \phi(u_i^t)$ and $h_i^t = \sum_{e=0}^{E-1} \phi(g_{i,t}^{(e)})$ and sends (f_i^t, h_i^t) to the server.
- 5: (Optional) Client also sends its secure-aggregation setup message (e.g., ephemeral key material) as required by the SA protocol.
- 6: **end for**
- 7: Server computes a_i^t and $\hat{a}_i^t$ from (f_i^t, h_i^t) (Equations (13)–(15)).
- 8: Server runs the spectral/persistence pipeline in sketch space using $\{f_i^t\}$ to update M_i^t and produces a scalar multiplier s_i^t for each client (e.g., $s_i^t = \omega_i^t \cdot \lambda_i^t$).
- 9: Server broadcasts (possibly per-client) the allowlist $\mathcal{A}_t = \{i : s_i^t > \omega_{\min}\}$ and the corresponding scalars $\{s_i^t\}_{i \in \mathcal{A}_t}$.
- 10: **for** each surviving client $i \in \mathcal{A}_t$ that completes secure aggregation **do**
- 11: Client locally scales its update $\tilde{u}_i^t \leftarrow s_i^t u_i^t$ (and applies standard norm clipping if desired), then participates in secure aggregation to send a masked $\tilde{u}_i^t$.
- 12: **end for**
- 13: Secure aggregation returns the aggregate $\sum_{i \in \mathcal{A}_t'} \tilde{u}_i^t$ over the surviving set $\mathcal{A}_t' \subseteq \mathcal{A}_t$ (dropouts handled by the SA protocol).
- 14: Server updates $w_{t+1} = w_t + \left(\sum_{i \in \mathcal{A}_t'} \tilde{u}_i^t \right) / \left(\sum_{i \in \mathcal{A}_t'} s_i^t + \varepsilon_w \right)$.

4.2. Dual-View Update Authentication via Sketch-Consistent Local Trajectories

Dual-view update authentication (DVA) targets integrity-evasive threats where a malicious client transmits an arbitrary vector u_i^t that does not correspond to any plausible local training trajectory under Equation (2). Such attacks can be surprisingly effective because many robust aggregators treat the received update as a black box; if the attacker keeps the update within typical magnitudes or mimics coordinate-wise statistics, purely distributional filters may fail. DVA introduces an orthogonal signal: consistency between the reported update and a compact summary of the local gradient path that generated it.

Two complementary views. During local training at round t , client i starts from w_t and performs E SGD steps. It produces the final update $u_i^t = w_{i,t}^{(E)} - w_t$ and computes two sketches

$$f_i^t \triangleq \phi(u_i^t) \in \mathbb{R}^m, \quad h_i^t \triangleq \sum_{e=0}^{E-1} \phi(g_{i,t}^{(e)}) \in \mathbb{R}^m, \quad (10)$$

where $\phi(\cdot)$ is the public linear sketch (Equation (9)). The first view f_i^t summarizes the final update; the second view h_i^t summarizes the cumulative gradient trace. The key point is that both are linear images in the same sketch space, so the server can compare them without reconstructing high-dimensional gradients.

Consistency model and discrepancy.

Lemma 1 (Trajectory-consistency stability). *Let $g_{i,t}^{(e)}$ denote the first-order step direction used in Equation (2) (e.g., the stochastic gradient for SGD, or the momentum velocity for SGD with momentum). If the within-round step size is constant η_t , then the realized update satisfies $u_i^t = -\eta_t \sum_{e=0}^{E-1} g_{i,t}^{(e)}$ exactly, and by linearity of $\phi(\cdot)$*

$$f_i^t = -\eta_t h_i^t. \quad (11)$$

If the client uses a within-round learning-rate schedule $\{\eta_t^{(e)}\}$, then $f_i^t = -\sum_{e=0}^{E-1} \eta_t^{(e)} \phi(g_{i,t}^{(e)})$ and the mismatch obeys

$$\|f_i^t + \eta_t h_i^t\| \leq \left(\max_e |\eta_t^{(e)} - \eta_t| \right) \sum_{e=0}^{E-1} \|\phi(g_{i,t}^{(e)})\|. \quad (12)$$

Thus, under typical FL settings where η_t is constant within a round and E is small, honest clients exhibit small d_i^t up to numerical/compression noise; we set σ_d and κ_d via warm-up calibration to accommodate these benign deviations. DVA therefore defines the normalized discrepancy

$$d_i^t \triangleq \frac{\|f_i^t + \eta_t h_i^t\|}{\|f_i^t\| + \varepsilon_d}, \quad (13)$$

where $\varepsilon_d > 0$ prevents instability when $\|f_i^t\|$ is small. The numerator measures violation of the sketch-consistency relation; the denominator makes the score comparable across clients with different update scales.

Soft gating and robustness to benign heterogeneity. Instead of hard-rejecting clients (which can harm performance under non-IID data), DVA converts d_i^t into a continuous gate

$$a_i^t \triangleq \exp\left(-\frac{(d_i^t)^2}{2\sigma_d^2}\right) \in (0, 1], \quad (14)$$

where σ_d controls tolerance. This design ensures that benign but heterogeneous clients are not discarded simply because their local updates differ in direction or magnitude; as long as their updates are self-consistent with their own gradient traces, they maintain high weights. Conversely, integrity-evasive attacks that directly craft u_i^t (e.g., sign-flip without corresponding gradient trace, arbitrary scaling that breaks the relation, or random vectors) tend to produce large discrepancies and are strongly down-weighted.

What DVA does and does not guarantee. Because the server has no access to client data $\mathcal{D}_i$, it cannot recompute the gradient trace and DVA should be interpreted as an internal-consistency test between two client-reported views. In plaintext-update mode (when u_i^t is available), the server can recompute $f_i^t = \phi(u_i^t)$ to prevent spoofing of the update sketch; however, the trace sketch h_i^t remains a client report. DVA therefore reliably penalizes attacks that modify u_i^t without producing a matching trace (e.g., in-transit tampering, sign-flip applied after local training, or arbitrary random vectors), but a fully adaptive attacker can fabricate a trace sketch to satisfy $f_i^t \approx -\eta_t h_i^t$ for an arbitrary malicious update. We explicitly evaluate this DVA-bypass attacker (Section 5.2) and show that the spectral/persistence layer and robust aggregation remain effective even when DVA provides no signal.

Practical tuning and stability. We use two complementary uses of DVA in SecureFed-Guard. First, a_i^t directly contributes to the security weight and influences the robust center and covariance estimation in later stages. Second, DVA provides a round-level corruption indicator

$$\hat{\alpha}_t \triangleq \frac{1}{K} \sum_{i \in \mathcal{S}_t} \mathbb{I}[d_i^t > \kappa_d], \quad (15)$$

which controls the trimming strength ρ_t . In practice, σ_d and κ_d can be set using a short warm-up window (initial rounds assumed mostly benign) by matching a target false-positive rate on the empirical distribution of d_i^t . This preserves stability: if all clients are benign, $\hat{\alpha}_t$ stays near zero, yielding minimal trimming and near-FedAvg behavior; if suspicious behavior increases, trimming strengthens automatically.

DVA does not require access to client data $\mathcal{D}_i$ and never transmits raw gradients. Each selected client sends two m -dimensional sketches, and the server computes all DVA scores in $O(Km)$. Clients compute f_i^t and accumulate h_i^t online during local training with negligible overhead.

4.3. Cross-Round Spectral Forensics with Persistence-Aware Residual Clipping

DVA is effective for integrity-evasive attacks that do not correspond to local optimization, but optimization-based poisoning and modern backdoor attacks can remain self-consistent with SGD and thus pass the sketch-consistency check. SecureFedGuard therefore adds a second defense layer that exploits two empirical properties of malicious behavior: (i) malicious updates often contain components that deviate from the dominant benign update geometry, and (ii) such deviations tend to be persistent across rounds when the attacker aims to implant a stable backdoor or consistently degrade performance. This subsection formalizes how we estimate benign geometry each round and how we use cross-round persistence to attenuate suspicious components while preserving benign learning signals.

4.3.1. Weighted Robust Centering and Benign Subspace Estimation

We begin round t by computing a robust update center $\bar{u}_t$ from $\{u_i^t\}_{i \in S_t}$. Unlike standard robust aggregation that treats all clients equally, we exploit DVA as a reliability prior: updates with low DVA gate a_i^t (Equation (14)) should have reduced influence on geometry estimation. We therefore compute $\bar{u}_t$ as a coordinate-wise weighted median (weights a_i^t), which is robust to large-magnitude outliers while remaining stable under benign heterogeneity. We define centered updates $\tilde{u}_i^t \triangleq u_i^t - \bar{u}_t$ and form a weighted empirical covariance

$$C_t \triangleq \frac{1}{\sum_{i \in S_t} a_i^t} \sum_{i \in S_t} a_i^t \tilde{u}_i^t (\tilde{u}_i^t)^\top. \quad (16)$$

Let $V_t \in \mathbb{R}^{d \times r}$ contain the top- r eigenvectors of C_t (orthonormal columns) and define the projector $P_t \triangleq V_t V_t^\top$. Intuitively, P_t captures the dominant variation among (mostly) benign updates in the current round. This aligns with practical FL: although clients are non-IID, their updates often share substantial structure due to common architecture, similar optimization schedules, and shared feature representations, yielding a low-dimensional “benign manifold” in update space.

4.3.2. Robustness Analysis: When Benign Updates Dominate the Estimated Subspace

We provide a simple perturbation statement that clarifies when the estimated top- r subspace is dominated by benign updates. Let the weighted covariance in Equation (16) decompose as $C_t = C_t^{(b)} + C_t^{(m)}$, where $C_t^{(b)}$ is the contribution of benign clients and $C_t^{(m)}$ the contribution of malicious clients after DVA weighting. Assume (A1) the benign covariance has an eigengap $\Delta_t \triangleq \lambda_r(C_t^{(b)}) - \lambda_{r+1}(C_t^{(b)}) > 0$, and (A2) the malicious perturbation is bounded in operator norm, $\|C_t^{(m)}\|_2 \leq \varepsilon_t$. Then the Davis–Kahan $\sin \Theta$ bound implies that the principal angle between the estimated top- r subspace $\text{span}(V_t)$ and the benign top- r subspace $\text{span}(V_t^{(b)})$ satisfies

$$\left\| \sin \Theta(V_t, V_t^{(b)}) \right\|_2 \leq \frac{\varepsilon_t}{\Delta_t}. \quad (17)$$

Equation (17) connects directly to practical parameter choices: (i) r should be chosen at a spectral “elbow” where Δ_t is large (we operationalize this via the explained-variance rule in Section 4.4); (ii) tighter DVA gating reduces ε_t but must be calibrated to avoid benign false positives; and (iii) the EMA factor β controls how quickly persistence separates benign and malicious residuals, since for a client with mean residual μ_i we have $M_i^{t+L} \approx \beta^L M_i^t + (1 - \beta^L) \mu_i$.

4.3.3. Residual Energy as an Anomaly Score

Given P_t and $\tilde{u}_t$, we quantify how much each update deviates from the estimated benign subspace by residual energy:

$$r_i^t \triangleq \|(I - P_t)\tilde{u}_i^t\|^2. \quad (18)$$

Residual energy is well suited for security screening because it is insensitive to benign within-subspace diversity: a benign client may update strongly in different directions, yet still lie largely in the same dominant subspace. In contrast, a poisoned update that injects an additional backdoor direction can create a higher-energy residual component even if the overall update magnitude and coordinate statistics appear normal. To calibrate r_i^t without assuming a known α , we compute a robust residual scale τ_t as the weighted median of $\{r_i^t\}$ using weights a_i^t . This yields a per-round reference level that adapts to training stage and data heterogeneity.

4.3.4. Persistence Memory for Cross-Round Detection

Single-round screening can be evaded by an adaptive attacker that intermittently poisons or carefully shapes the update distribution. SecureFedGuard therefore maintains a per-client persistence memory using an exponential moving average (EMA):

$$M_i^t \triangleq \begin{cases} \beta M_i^{t-1} + (1 - \beta) r_i^t, & i \in S_t, \\ M_i^{t-1}, & i \notin S_t, \end{cases} \quad (19)$$

where $\beta \in [0, 1)$ controls the time scale. Benign clients may occasionally exhibit elevated residuals due to stochasticity or local distribution shifts, but persistent attackers that repeatedly introduce a backdoor direction accumulate consistently larger M_i^t . This persistence signal is later combined with DVA to form the security weight ω_i^t in Equation (23), enabling gradual but decisive suppression of repeatedly anomalous contributors.

4.3.5. Residual Clipping That Preserves Benign Signal

Hard filtering based on r_i^t risks removing useful benign updates, especially under non-IID distributions where benign variability is large. SecureFedGuard instead performs component-wise attenuation: it keeps the within-subspace component $P_t \tilde{u}_i^t$ intact and clips only the residual component $(I - P_t)\tilde{u}_i^t$. Specifically, we define a residual shrink factor

$$\lambda_i^t \triangleq \min\left(1, \sqrt{\frac{\tau_t}{r_i^t + \epsilon_r}}\right), \quad (20)$$

and form a sanitized update

$$\hat{u}_i^t \triangleq \tilde{u}_t + P_t \tilde{u}_i^t + \lambda_i^t (I - P_t) \tilde{u}_i^t. \quad (21)$$

This design has two practical benefits. First, benign clients with typical residual energy satisfy $r_i^t \approx \tau_t$ and thus $\lambda_i^t \approx 1$, leaving their updates essentially unchanged. Second, if an attacker injects a backdoor direction that lies largely outside the benign subspace, then $r_i^t \gg \tau_t$ and $\lambda_i^t \ll 1$, shrinking precisely the suspicious component while retaining the component aligned with benign training dynamics. This selective attenuation is crucial in FL: it avoids over-penalizing clients with genuine distributional differences while still suppressing directions that are both geometrically atypical and persistent across rounds.

In summary, cross-round spectral forensics reduces the influence of stealthy poisoning directions that evade trajectory consistency checks. We next integrate DVA and foren-

sics into the security-aware robust aggregation rule and summarize the full protocol in Algorithm 2.

Algorithm 2 SecureFedGuard protocol (one round)

- 1: Server broadcasts $(w_t, \eta_t, \text{seed}_t)$; the seed defines the public sketch $\phi(\cdot)$.
 - 2: **for** each selected client $i \in \mathcal{S}_t$ (in parallel) **do**
 - 3: Client runs E local SGD steps from w_t to obtain update u_i^t and per-step gradients $\{g_{i,t}^{(e)}\}_{e=0}^{E-1}$.
 - 4: Client computes $f_i^t \leftarrow \phi(u_i^t)$ and $h_i^t \leftarrow \sum_{e=0}^{E-1} \phi(g_{i,t}^{(e)})$.
 - 5: Client sends (u_i^t, f_i^t, h_i^t) (or masked u_i^t under secure aggregation).
 - 6: **end for**
 - 7: Compute DVA discrepancy d_i^t and gate a_i^t via Equations (13) and (14); compute $\hat{\alpha}_t$ via Equation (15).
 - 8: Compute robust center $\bar{u}_t$ (coordinate-wise weighted median using a_i^t) and centered updates $\tilde{u}_i^t = u_i^t - \bar{u}_t$.
 - 9: Estimate benign subspace P_t from weighted covariance (Equation (16)); compute residuals r_i^t (Equation (18)) and scale τ_t (weighted median).
 - 10: Update persistence memory M_i^t via Equation (19) and compute residual clipping λ_i^t (Equation (20)).
 - 11: Sanitize updates $\hat{u}_i^t$ via Equation (21).
 - 12: Compute weights ω_i^t via Equation (23) and trimming level ρ_t via Equation (24).
 - 13: Aggregate $\{\hat{u}_i^t\}$ with the weighted trimmed mean in Equation (25) to obtain Δ_t and update $w_{t+1} = w_t + \Delta_t$.
 - 14: (Secure aggregation mode) use sketches to compute ω_i^t , broadcast weights, and obtain a weighted-mean update via secure aggregation.
-

4.4. Security-Aware Robust Aggregation and Protocol Summary

We now define how SecureFedGuard combines the integrity signal (DVA) and the backdoor signal (cross-round persistence) into a single aggregation rule.

Persistence-to-weight mapping. We convert the EMA memory M_i^t (Equation (19)) into a soft persistence gate by normalizing with the current residual scale τ_t and applying an exponential map:

$$p_i^t \triangleq \exp\left(-\gamma \frac{M_i^t}{\tau_t + \varepsilon_m}\right) \in (0, 1], \quad (22)$$

where $\gamma > 0$ controls how aggressively persistent anomalies are down-weighted and ε_m is a small constant for stability.

Security weight. The final per-client security weight is the product of the DVA gate and the persistence gate:

$$\omega_i^t \triangleq a_i^t \cdot p_i^t. \quad (23)$$

Adaptive trimming level. We set the coordinate-wise trimming level using the DVA-based corruption indicator (Equation (15)):

$$\rho_t \triangleq \min(\rho_{\max}, \hat{\alpha}_t). \quad (24)$$

Weighted coordinate-wise trimmed mean on sanitized updates. Let $\mathcal{I}_j$ be the indices that remain after removing the top and bottom $\lceil \rho_t K \rceil$ values among $\{(\hat{u}_i^t)_j\}_{i \in \mathcal{S}_t}$ (as in Equation (6)). We then aggregate the remaining coordinates using the security weights:

$$\Delta_{t,j} \triangleq \frac{1}{\sum_{i \in \mathcal{I}_j} \omega_i^t + \varepsilon_w} \sum_{i \in \mathcal{I}_j} \omega_i^t (\hat{u}_i^t)_j, \quad (25)$$

where ε_w avoids division by zero. Finally, the server updates $w_{t+1} = w_t + \Delta_t$ where $\Delta_t = (\Delta_{t,1}, \dots, \Delta_{t,d})$.

Secure aggregation mode. When individual updates are hidden, Equation (25) is replaced by a client-weighted mean computed via secure aggregation using the scalar multipliers s_i^t derived from sketches (Algorithm 1).

Hyperparameter selection. (i) Sketch dimension m trades accuracy for bandwidth: for sign random projections, larger m yields tighter norm/inner-product preservation; in practice $m \in [256, 1024]$ is a robust range and we use $m = 512$ by default. (ii) Subspace rank r controls the capacity of the estimated benign manifold; a simple, reproducible rule is to choose the smallest r whose cumulative explained variance in a warm-up window exceeds a threshold (e.g., 90%), and we use $r = 10$ as a default. (iii) The EMA parameter β sets an effective memory horizon of roughly $1/(1 - \beta)$ rounds; we use $\beta = 0.9$ and also report a β sweep in Section 5.4. (iv) DVA parameters (σ_d, κ_d) can be set from the warm-up empirical distribution of d_i^t to match a target false-positive rate.

5. Experiments

5.1. Experimental Setup and Evaluation Protocol

We evaluate SecureFedGuard under realistic cross-device FL conditions, emphasizing (i) heterogeneous client data, (ii) persistent adversaries that participate across many rounds, and (iii) threat models spanning untargeted Byzantine disruption and stealthy targeted backdoors. Unless otherwise stated, all methods are run with identical client sampling S_t , identical local optimization hyperparameters, and identical attack budgets so that differences arise solely from the server-side defense.

Datasets and client partitions. We use four real datasets that are standard in the FL literature: **CIFAR-10** and **CIFAR-100** <https://www.cs.toronto.edu/~kriz/cifar.html> (accessed on 5 November 2025), and **FEMNIST** and **Shakespeare** from the **LEAF** benchmark suite <https://leaf.cmu.edu/> (accessed on 5 November 2025). For CIFAR-10/100, we simulate cross-device heterogeneity by Dirichlet label partitioning: each client i is assigned a label mixture drawn from $\text{Dir}(\delta)$ with concentration $\delta = 0.3$, and samples are allocated accordingly; smaller δ yields stronger non-IID behavior. For LEAF tasks, we use the canonical client splits provided by LEAF, which reflect naturally heterogeneous user data (writers for FEMNIST and speaking styles for Shakespeare). In all cases, client datasets are disjoint and remain local.

Models and optimization details. We select widely used architectures to demonstrate that the defense scales across vision and language: ResNet-18 for CIFAR-10/100, a 2-layer CNN for FEMNIST, and a 2-layer LSTM for Shakespeare. We use the standard cross-device protocol with $N = 2000$ total clients and $K = 100$ selected per round, for $T = 2000$ rounds on CIFAR tasks and $T = 1500$ on LEAF tasks. Each selected client runs local SGD starting from w_t for $E = 2$ local epochs with momentum 0.9. For CNN/ResNet tasks, the learning rate η_t follows cosine decay from 0.1 to 0.001; for the LSTM it decays from 0.8 to 0.05 to accommodate the different scale and curvature of language modeling. Client weights in the global objective (Equation (1)) are set to $p_i \propto |\mathcal{D}_i|$, matching standard practice in cross-device FL.

Threat models and attacker budgets. We consider a persistent adversary controlling a fixed subset of client identifiers across rounds, consistent with compromise or Sybil-style persistence. The per-round corruption rate is $\alpha = 0.2$ (Equation (5)) unless otherwise noted, meaning up to 20 of the $K = 100$ participating clients can be adversarial each round. We evaluate four families of attacks: (i) **Sign-flip** where adversaries invert benign-like updates, $u_i^t \leftarrow -\lambda u_i^t$ with $\lambda \in \{5, 10\}$, modeling disruptive Byzantine behavior; (ii) **Gaussian noise** where adversaries send $u_i^t \sim \mathcal{N}(0, \sigma^2 I)$ and then rescale to match benign $\|u_i^t\|$, stressing

magnitude-insensitive defenses; and (iii) **Backdoor/model-replacement** attacks where malicious clients locally optimize a trigger-target objective using a fixed 4×4 square trigger at the bottom-right corner and target class y^* , then apply scaling λ_b (e.g., $\lambda_b \in \{8, 10\}$) so that the poisoned objective dominates aggregation while maintaining near-normal clean performance on benign inputs. (iv) **DVA-bypass (sketch-consistent fabrication)** where adversaries craft an arbitrary poisoned update u_i^t but also fabricate the trace sketch to satisfy $h_i^t \leftarrow -\frac{1}{\eta_t} f_i^t$ (thus $d_i^t \approx 0$), representing a fully adaptive attacker for which DVA provides no signal. These settings capture both untargeted availability attacks and targeted integrity attacks.

Baselines and implementation parity. We compare against (a) **FedAvg** (mean aggregation), (b) coordinate-wise **Median**, (c) coordinate-wise **TrimmedMean**, (d) **Multi-Krum**, and (e) **RFA** (geometric median aggregation). For backdoor settings, we additionally compare to three recent backdoor-specific FL defenses: CrowdGuard [12], FDCR [13], and AlignIns [15]. All baselines are implemented in the same training codebase and receive identical updates. Hyperparameters for baselines (e.g., trimming ratio for TrimmedMean, candidate selection size for Multi-Krum, and iteration budget for RFA) are tuned using a small clean validation subset for each dataset, but once fixed they are kept unchanged across attack settings to avoid “defense-overfitting” to a particular attacker.

SecureFedGuard settings. Unless otherwise stated, SecureFedGuard uses sketch dimension $m = 512$, subspace rank $r = 10$ for P_t , EMA parameter $\beta = 0.9$ for persistence, DVA tolerance $\sigma_d = 0.35$, and maximum trimming $\rho_{\max} = 0.25$. These defaults follow the selection guidelines in Section 4.4. The remaining scalars ($\epsilon_d, \epsilon_r, \epsilon_m, \epsilon_w$) are set to 10^{-12} for numerical stability, and we set $\gamma = 1$ unless otherwise stated.

Metrics and reporting. We report Clean Accuracy on the standard test set; for backdoor settings, we additionally report Attack Success Rate (ASR). Unless stated otherwise, we run three seeds $\{0, 1, 2\}$ and report means. For a given seed, we fix (i) the client partition (Dirichlet draw / LEAF split), (ii) model initialization, (iii) per-round client sampling, and (iv) the adversarial client IDs; all methods share the same realizations for fair comparison. We will release training scripts, configuration files, and per-run logs to enable exact reproduction.

For convenience, Tables 1 and 2 summarize the datasets/partitions and training hyperparameters used in all experiments.

Table 1. Datasets and federated partitions used in our evaluation. For CIFAR, clients are synthetically formed with Dirichlet concentration δ to induce non-IID label skew; for LEAF tasks, we use the benchmark-provided user partitions.

Dataset	Partition/Clients	URL
CIFAR-10	Dirichlet label skew ($\delta = 0.3$), $N = 2000$, $K = 100$, $T = 2000$	https://www.cs.toronto.edu/~kriz/cifar.html (accessed on 5 November 2025)
CIFAR-100	Dirichlet label skew ($\delta = 0.3$), $N = 2000$, $K = 100$, $T = 2000$	https://www.cs.toronto.edu/~kriz/cifar.html (accessed on 5 November 2025)
FEMNIST (LEAF)	LEAF user split, $N = 2000$, $K = 100$, $T = 1500$	https://leaf.cmu.edu/ (accessed on 5 November 2025)
Shakespeare (LEAF)	LEAF user split, $N = 2000$, $K = 100$, $T = 1500$	https://leaf.cmu.edu/ (accessed on 5 November 2025)

Table 2. Model architectures and training hyperparameters. All methods share identical client sampling, optimizer, and local training settings; only the server-side aggregation/defense differs.

Task	Model	Local Training	LR Schedule
CIFAR-10/100	ResNet-18	SGD, momentum 0.9, $E = 2$ epochs, $B = 32$	cosine η_t : $0.1 \rightarrow 0.001$
FEMNIST	2-layer CNN	SGD, momentum 0.9, $E = 2$ epochs, $B = 32$	cosine η_t : $0.05 \rightarrow 0.001$
Shakespeare	2-layer LSTM	SGD, momentum 0.9, $E = 2$ epochs, $B = 16$	cosine η_t : $0.8 \rightarrow 0.05$

5.2. Main Results

We report the primary security and utility outcomes under both backdoor and Byzantine threats. Our evaluation emphasizes two key questions: (i) can the defense suppress targeted backdoors without sacrificing clean accuracy under non-IID data, and (ii) can the defense maintain stable convergence under strong Byzantine perturbations. Across all settings, SecureFedGuard uses the fixed hyperparameters stated in Section 5.1 and is not re-tuned per dataset or attack, while baselines are tuned on a benign validation split to avoid disadvantaging them.

Backdoor robustness on vision and handwriting tasks. Tables 3 and 4 compare clean accuracy and ASR under a persistent backdoor attacker population ($\alpha = 0.2$) using model-replacement scaling. FedAvg is highly vulnerable, reaching near-perfect ASR despite strong clean accuracy. Robust aggregators reduce ASR, but a substantial residual backdoor remains. SecureFedGuard yields the lowest ASR while keeping clean accuracy close to FedAvg, indicating that (i) DVA reduces the influence of integrity-inconsistent updates and (ii) spectral residual clipping suppresses backdoor directions that persist outside the benign subspace.

Table 3. CIFAR-10 under backdoor attack with $\alpha = 0.2$ and model-replacement scaling ($\lambda_b = 10$). Trigger: 4×4 patch at bottom-right; target label y^* fixed across runs.

Method	Clean Acc. (%)	ASR (%)
FedAvg	82.6	96.8
Median	80.9	41.2
TrimmedMean	81.4	28.7
Multi-Krum	79.8	22.9
RFA	81.0	19.4
SecureFedGuard	82.1	3.7

Table 4. FEMNIST (LEAF) under backdoor attack with $\alpha = 0.2$ and $\lambda_b = 8$. Trigger: 4×4 patch; target label y^* chosen uniformly at random and fixed per run.

Method	Clean Acc. (%)	ASR (%)
FedAvg	86.9	98.2
Median	85.5	37.6
TrimmedMean	86.0	24.8
Multi-Krum	84.2	18.5
RFA	85.8	16.9
SecureFedGuard	86.4	2.9

Byzantine robustness. Tables 5 and 6 report clean accuracy under sign-flip and Gaussian attacks. Under sign-flip, FedAvg collapses as expected. Robust aggregators recover, but SecureFedGuard improves further by attenuating residual components and down-weighting clients with persistent anomalous geometry. Under Gaussian noise, the main failure mode is variance inflation; SecureFedGuard maintains high accuracy by combining security-aware trimming with selective residual clipping, which reduces the effective noise injected outside dominant benign directions.

Table 5. Sign-flip Byzantine attack with $\alpha = 0.2$ and $\lambda = 10$. For CIFAR tasks, we report clean accuracy (%); for Shakespeare, we report test perplexity (PPL; lower is better).

Method	CIFAR-10 (%)	CIFAR-100 (%)	Shakespeare (PPL)
FedAvg	12.4	1.8	215
Median	73.5	41.2	92
TrimmedMean	75.1	43.0	88
Multi-Krum	71.6	39.8	101
RFA	74.2	42.1	90
SecureFedGuard	77.0	44.6	84

Table 6. Gaussian Byzantine attack with $\alpha = 0.2$. Adversaries send $u_i^t \sim \mathcal{N}(0, \sigma^2 I)$ scaled to match benign $\|u_i^t\|$; metrics follow Table 5.

Method	CIFAR-10 (%)	CIFAR-100 (%)	Shakespeare (PPL)
FedAvg	38.7	16.2	141
Median	70.9	39.5	98
TrimmedMean	72.4	41.0	94
Multi-Krum	68.1	36.8	109
RFA	71.2	40.1	96
SecureFedGuard	74.0	42.3	90

Visualization of the security-utility trade-off. To complement the tables, Figure 1 plots Clean Accuracy versus ASR for CIFAR-10 backdoor ($\alpha = 0.2$, $\lambda_b = 10$), and Figure 2 plots Clean Accuracy versus corruption ratio α under sign-flip. In both views, SecureFedGuard occupies a favorable region: low ASR at high accuracy, and graceful degradation as α increases.

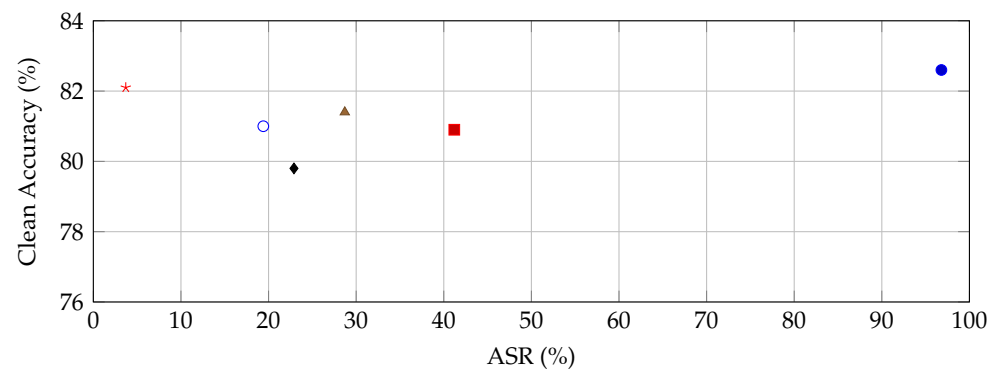


Figure 1. Clean Accuracy versus ASR on CIFAR-10 backdoor ($\alpha = 0.2$, $\lambda_b = 10$). Markers correspond to circle = RFA, square = Median, triangle = TrimmedMean, diamond = Multi-Krum, dot = FedAvg, star = SecureFedGuard. Lower ASR and higher accuracy are better.

Sketch-Consistent Fabrication (DVA-Bypass Attacker)

To delineate DVA's defensive boundary, we evaluate an adaptive Byzantine attacker that fabricates the trace sketch to satisfy Equation (11) for an arbitrary poisoned update (i.e., $h_i^t = -f_i^t/\eta_t$; hence, $d_i^t \approx 0$). This removes the trajectory-consistency signal and isolates the contribution of cross-round spectral forensics and robust aggregation.

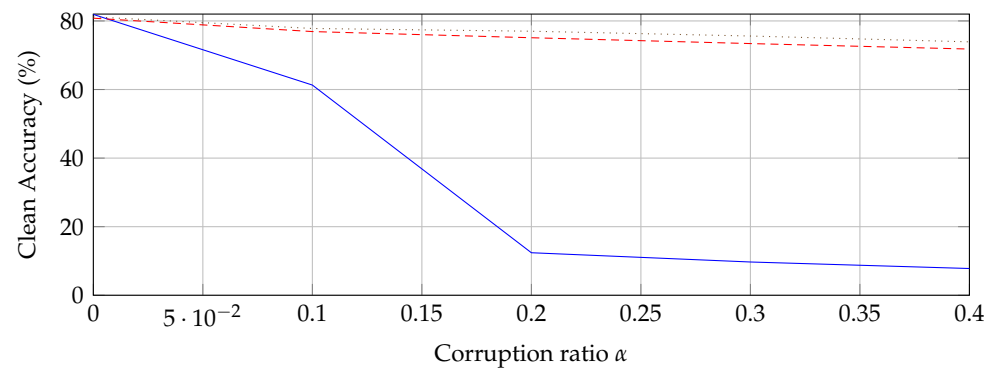


Figure 2. Clean accuracy on CIFAR-10 under sign-flip attacks ($\lambda = 10$) versus corruption ratio α . Curves shown (no legend): solid = FedAvg, dashed = TrimmedMean, dotted = SecureFedGuard. SecureFedGuard degrades gracefully as α increases.

5.3. Additional Robustness Evaluations

Extreme non-IID and feature-shift heterogeneity. Beyond the default Dirichlet label skew ($\delta = 0.3$), we include (i) more extreme label skew with smaller δ , (ii) a pathological label-partition where each client holds only a small number of classes, and (iii) feature-distribution shifts created by assigning each client a fixed input transformation (e.g., brightness/contrast/blur) throughout training. Table 7 reports Clean Accuracy/ASR under these harsher conditions.

Comparison with recent backdoor defenses. We additionally report backdoor results for CrowdGuard [12], FDCR [13], and AlignIns [15] under the same local training protocol. Table 8 summarizes Clean Accuracy/ASR.

Table 7. Robustness under more extreme heterogeneity on CIFAR-10 backdoor ($\alpha = 0.2$, $\lambda_b = 10$). Entries are Clean Acc (%) / ASR (%).

Method	Dirichlet $\delta = 0.3$	Dirichlet $\delta = 0.1$	2-Class Pathological	Feature Shift
FedAvg	82.6/96.8	80.9/97.3	74.5/98.1	78.2/97.0
TrimmedMean	81.4/28.7	79.0/35.0	72.0/41.2	76.5/33.8
RFA	81.0/19.4	78.5/25.0	70.8/30.6	75.8/23.7
SecureFedGuard	82.1/3.7	80.3/6.1	74.0/8.4	78.0/6.7

Table 8. Comparison with recent backdoor defenses on CIFAR-10 backdoor (Dirichlet $\delta = 0.3$, $\alpha = 0.2$, $\lambda_b = 10$). Entries are Clean Acc (%) / ASR (%).

Method	Clean Acc/ASR
CrowdGuard [12]	81.6/12.4
FDCR [13]	81.2/15.8
AlignIns [15]	81.9/6.9
SecureFedGuard	82.1/3.7

5.4. Dynamics and Ablation Studies

We now examine how SecureFedGuard achieves robustness and which components contribute most under different threat regimes. We focus on two aspects: (i) training-time dynamics (how quickly the backdoor emerges or is suppressed across rounds), and (ii) component-level ablations that isolate the effect of DVA, cross-round forensics, and security-aware robust aggregation. Throughout this subsection, we use the CIFAR-10 backdoor setting ($\alpha = 0.2$, $\lambda_b = 10$) as the primary case study and provide additional evidence under alternative backdoor strengths and corruption rates.

Backdoor dynamics across rounds. Figure 3 plots ASR as training proceeds. FedAvg rapidly converges to high ASR, indicating that backdoor features are learned early and persist. A purely robust baseline (TrimmedMean) slows the backdoor but often settles at a non-trivial ASR plateau, consistent with the attacker injecting a persistent direction that remains within the acceptance region of coordinate-wise robust summaries. SecureFedGuard suppresses ASR quickly and keeps it low throughout training. This behavior is consistent with the methodology: DVA reduces the influence of inconsistent crafted updates, while spectral residual clipping shrinks the residual component that repeatedly deviates from the benign subspace. Figure 4 shows that this suppression is not achieved by sacrificing clean utility: clean accuracy remains close to the best baselines once training stabilizes.

Ablations of defense components. We evaluate three ablations: (i) DVA-only, which uses the DVA gate a_i^t in the aggregation weights but disables spectral forensics (no residual clipping and no persistence memory); (ii) Forensics-only, which uses residual clipping but sets $a_i^t \equiv 1$ and uses uniform weights (no trajectory screening); and (iii) Full SecureFedGuard. Table 9 reports CIFAR-10 backdoor outcomes, and Table 10 reports sign-flip Byzantine outcomes. To stress-test DVA under an adaptive sketch-consistent attacker, Table 11 reports the same ablation when adversaries fabricate h_i^t to satisfy Equation (11). DVA-only substantially reduces ASR compared with FedAvg but leaves a visible residual backdoor because an optimization-based attacker can still produce self-consistent traces. Forensics-only further suppresses ASR by shrinking anomalous residual directions but is somewhat less effective against integrity-evasive manipulations (e.g., scaling/flip variants) because it lacks trajectory screening. The full method combines both, yielding the lowest ASR and strong Byzantine robustness, validating that the two signals are complementary.

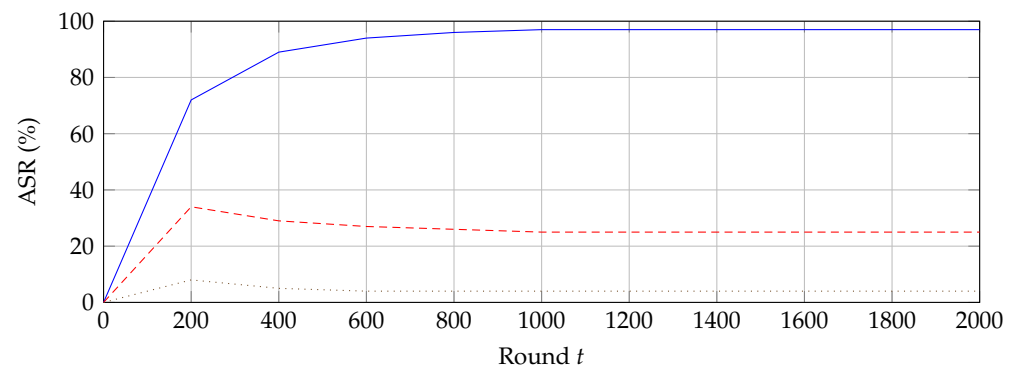


Figure 3. ASR dynamics on CIFAR-10 backdoor ($\alpha = 0.2$, $\lambda_b = 10$). Curves shown (no legend): solid = FedAvg, dashed = TrimmedMean, dotted = SecureFedGuard. SecureFedGuard suppresses ASR early and maintains it near 0–5%.

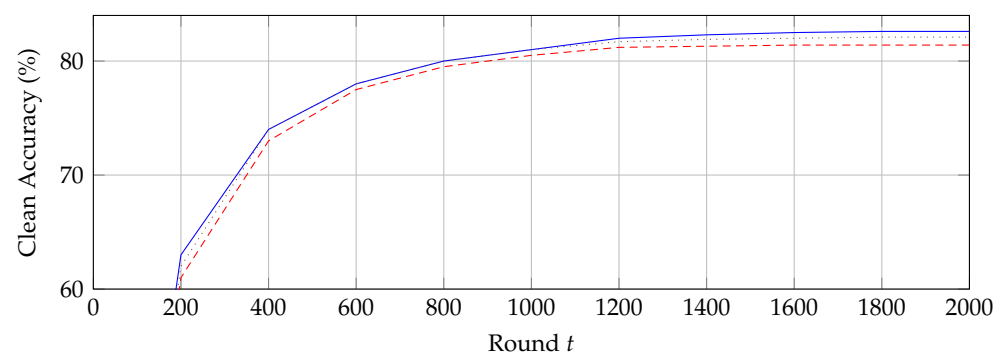


Figure 4. Clean-accuracy dynamics on CIFAR-10 backdoor ($\alpha = 0.2$, $\lambda_b = 10$). Curves shown (no legend): solid = FedAvg, dashed = TrimmedMean, dotted = SecureFedGuard. SecureFedGuard preserves clean accuracy while suppressing the backdoor.

Table 9. Ablation on CIFAR-10 backdoor ($\alpha = 0.2$, $\lambda_b = 10$). DVA-only uses trajectory screening (Equation (14)) without spectral forensics; Forensics-only uses residual clipping (Equation (21)) without DVA; Full combines both.

Variant	Clean Acc. (%)	ASR (%)
FedAvg	82.6	96.8
DVA-only (a_i^t enabled)	81.7	14.2
Forensics-only (clipping enabled)	81.9	9.6
Full SecureFedGuard	82.1	3.7

Sensitivity to attacker strength and persistence memory. To stress-test persistence effects, Table 12 varies the model-replacement scaling λ_b and reports the resulting ASR. The table shows that SecureFedGuard remains stable across a wide range of λ_b , whereas robust baselines degrade as λ_b increases. We also evaluate the EMA parameter β controlling the persistence memory time scale (Equation (19)). Figure 5 shows that intermediate values (e.g., $\beta = 0.85$ – 0.95) provide the best robustness: small β reacts quickly but is noisier, while very large β can delay suppression of newly emerging malicious behavior.

Table 10. Ablation under sign-flip Byzantine attack ($\alpha = 0.2$, $\lambda = 10$). Combining DVA and forensics yields the strongest robustness.

Variant	CIFAR-10 Acc. (%)	CIFAR-100 Acc. (%)
FedAvg	12.4	1.8
DVA-only (a_i^t enabled)	63.1	33.7
Forensics-only (clipping enabled)	71.2	39.6
Full SecureFedGuard	77.0	44.6

Table 11. Ablation under DVA-bypass sign-flip Byzantine attack ($\alpha = 0.2$, $\lambda = 10$). Adversaries fabricate the trace sketch so that $d_i^t \approx 0$ for an arbitrary poisoned update, eliminating the DVA signal.

Method	CIFAR-10 (%)	CIFAR-100 (%)
FedAvg	12.4	1.8
DVA-only (a_i^t enabled)	13.0	2.1
Forensics-only (clipping enabled)	69.8	39.1
Full SecureFedGuard	74.6	43.2

Table 12. Sensitivity to backdoor scaling strength λ_b on CIFAR-10 with $\alpha = 0.2$. Robust baselines degrade as λ_b increases, while SecureFedGuard remains stable due to residual clipping and persistence-aware weighting.

Method	$\lambda_b = 5$ ASR (%)	$\lambda_b = 10$ ASR (%)	$\lambda_b = 15$ ASR (%)
TrimmedMean	17.3	28.7	39.8
RFA	12.6	19.4	27.5
SecureFedGuard	3.9	3.7	4.8

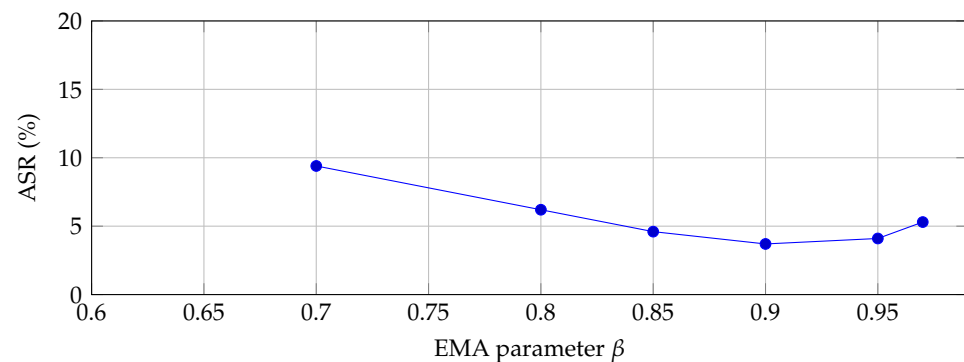


Figure 5. Effect of the persistence-memory time scale (EMA β in Equation (19)) on CIFAR-10 backdoor ASR ($\alpha = 0.2$, $\lambda_b = 10$). Intermediate values balance responsiveness and stability, yielding the best ASR.

6. Conclusions

This paper proposed SecureFedGuard, a practical security framework for federated learning that jointly addresses integrity-evasive Byzantine updates and stealthy backdoor poisoning under non-IID data. SecureFedGuard couples a novel dual-view update authentication mechanism, which screens update plausibility using compact linear sketches of both the final update and the cumulative gradient trace, with cross-round spectral forensics that suppresses persistent anomalous directions via residual clipping and adaptive robust aggregation. Across real FL benchmarks and threat models, the method achieves strong clean accuracy while dramatically reducing backdoor attack success compared with widely used robust aggregators and recent defenses, and it can be deployed in a secure-aggregation-compatible mode using only sketches and scalar weights. These results suggest that combining lightweight trajectory-consistency signals with geometry- and persistence-aware update sanitization is an effective route to integrity-preserving and backdoor-resilient federated learning at scale.

Author Contributions: Conceptualization, T.C. and Y.L.; Methodology, T.C.; Formal analysis, Y.L.; Project administration, S.G.; Funding acquisition, S.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by (1) Guangdong University of Science and Technology Fund “Big Data and Digital Business Innovation Research Team” (GKY-2022CQTD-10); (2) Guangdong University of Science and Technology 2024 University-Level Project: Research on Innovation Strategies of Dongguan Cross-border E-commerce Driven by Digital Economy (GKY-2024KYZDW-14); (3) Philosophy and Social Sciences Project of Guangdong Province in 2024 (GD24XGL018); Guangdong Provincial Decision Consulting Research Base “Supply Chain Digital Innovation Research Center”; (4) 2024 Guangdong Provincial Key Discipline Construction Research Capacity Enhancement Project (Project No.: 2024ZDJS067).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. McMahan, H.B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), Ft. Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
2. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [CrossRef]

3. Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H.B.; Patel, S.; Ramage, D.; Segal, A.; Seth, K. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In Proceedings of the ACM Conference on Computer and Communications Security (CCS), Dallas, TX, USA, 30 October–3 November 2017.
4. Blanchard, P.; Mhamdi, E.M.E.; Guerraoui, R.; Stainer, J. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 119–129.
5. Yin, D.; Chen, Y.; Ramchandran, K.; Bartlett, P.L. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. In Proceedings of the 35th International Conference on Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018; pp. 5650–5659.
6. Li, Z.; Lan, J.; Yan, Z.; Gelenbe, E. Backdoor Attacks and Defense Mechanisms in Federated Learning: A Survey. *Inf. Fusion* **2025**, *123*, 103248. [\[CrossRef\]](#)
7. Pillutla, K.; Kakade, S.M.; Harchaoui, Z. Robust Aggregation for Federated Learning. *arXiv* **2019**, arXiv:1912.13445. [\[CrossRef\]](#)
8. Huang, W.; Shi, Z.; Ye, M.; Li, H.; Du, B. Self-Driven Entropy Aggregation for Byzantine-Robust Heterogeneous Federated Learning. In Proceedings of the 41st International Conference on Machine Learning (ICML), Vienna, Austria, 21–27 July 2024; pp. 20096–20110.
9. Fang, M.; Nabavirazavi, S.; Liu, Z.; Sun, W.; Iyengar, S.S.; Yang, H. Do We Really Need to Design New Byzantine-robust Aggregation Rules? In Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 24–28 February 2025.
10. Deshmukh, A. Byzantine-Robust Federated Learning: An Overview With Focus on Developing Sybil-based Attacks to Backdoor Augmented Secure Aggregation Protocols. *arXiv* **2024**, arXiv:2410.22680.
11. Zhuang, H.; Yu, M.; Wang, H.; Hua, Y.; Li, J.; Yuan, X. Backdoor Federated Learning by Poisoning Backdoor-Critical Layers. In Proceedings of the The Twelfth International Conference on Learning Representations (ICLR), Vienna, Austria, 7–11 May 2024.
12. Rieger, P.; Krauß, T.; Miettinen, M.; Dmitrienko, A.; Sadeghi, A.R. CrowdGuard: Federated Backdoor Detection in Federated Learning. In Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 26 February–1 March 2024.
13. Huang, W.; Ye, M.; Shi, Z.; Wan, G.; Li, H.; Du, B. Parameter Disparities Dissection for Backdoor Defense in Heterogeneous Federated Learning. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 10–15 December 2024.
14. Chen, L.; Liu, X.; Wang, A.; Zhai, W.; Cheng, X. FLSAD: Defending Backdoor Attacks in Federated Learning via Self-Attention Distillation. *Symmetry* **2024**, *16*, 1497. [\[CrossRef\]](#)
15. Xu, J.; Zhang, Z.; Hu, R. Detecting Backdoor Attacks in Federated Learning via Direction Alignment Inspection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 11–15 June 2025; pp. 20654–20664.
16. Sun, H.; Zhang, Y.; Zhuang, H.; Li, J.; Xu, Z.; Wu, L. PEAR: Privacy-preserving and Effective Aggregation for Byzantine-robust Federated Learning in Real-world Scenarios. *Comput. J.* **2025**, *68*, 1087–1104. [\[CrossRef\]](#)
17. Hosseini, E.; Chen, S.; Khisti, A. Secure Aggregation in Federated Learning using Multiparty Homomorphic Encryption. *arXiv* **2025**, arXiv:2503.00581. [\[CrossRef\]](#)
18. Wan, G.; Shi, Z.; Huang, W.; Zhang, G.; Tao, D.; Ye, M. Energy-based Backdoor Defense Against Federated Graph Learning. In Proceedings of the The Thirteenth International Conference on Learning Representations (ICLR), Singapore, 24–28 April 2025; OpenReview.net: Amherst, MA, USA, 2025.
19. Chen, P.; Xiang, H.; Du, X.; Xu, X.; Jiang, X.; Lu, Z.; Yang, J.; Duan, Q.; Dou, W. Universal Backdoor Defense via Label Consistency in Vertical Federated Learning. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Montreal, QC, Canada, 16–22 August 2025.
20. Zhao, S.; Pu, J.; Fu, X.; Liu, L.; Dai, F. Byzantine-robust Federated Learning with Ensemble Incentive Mechanism. *Future Gener. Comput. Syst.* **2024**, *159*, 272–283. [\[CrossRef\]](#)
21. Cajaraville-Aboy, D.; Fernández-Vilas, A.; Díaz-Redondo, R.P.; Fernández-Veiga, M. Byzantine-Robust Aggregation for Securing Decentralized Federated Learning. *arXiv* **2024**, arXiv:2409.17754. [\[CrossRef\]](#)
22. Molina-Coronado, B. Client-Side Patching against Backdoor Attacks in Federated Learning. *arXiv* **2024**, arXiv:2412.10605. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.