BYZANTINE-ROBUST FEDERATED LEARNING WITH PRIVACY-COMPATIBLE ADAPTIVE CLIENT SELECTION VIA THOMPSON SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Byzantine-robust federated learning aims to protect global model training from malicious updates while retaining useful local updates from benign clients. Reliability assessments based on individual updates can confuse benign variation caused by heterogeneous data with malicious manipulation, risking both false rejection and false acceptance. History-based defenses can inherit this ambiguity when using the same evidence, while reduced participation can restrict corrective feedback for misjudged benign clients. In this paper, we propose BRFL-TS, which shifts reliability assessment to the utility of candidate models aggregated from client subsets and evaluated by clients on local data. We design an adaptive reward-and-penalty rule that updates client trust using clients' membership across utility-ranked groups and determines the number of penalized clients from current trust estimates. These trust updates guide Thompson-weighted subset selection, reducing penalized clients' selection chances while retaining opportunities for reassessment. By avoiding inspection of individual plaintext updates, BRFL-TS supports a common client-selection rule across federated learning systems protecting updates with differential privacy (DP), homomorphic encryption (HE), or secure multi-party computation (MPC). Through probabilistic analysis of the sampling and trust-update process, we prove positive lower bounds on clients' probabilities of receiving renewed reward or penalty feedback within finite training windows, and state-dependent upper bounds on expected waiting times during continued training. Experiments spanning four datasets and eight untargeted poisoning attacks demonstrate robustness across data-heterogeneity and Byzantine-client-ratio settings. Client-level decision evaluations further show that BRFL-TS limits Byzantine false acceptance while maintaining low benign false rejection in its utility-based reward-and-penalty feedback. On Fashion-MNIST without update protection, BRFL-TS improves attack-averaged accuracy by 8.01 percentage points over the strongest baseline.

1 INTRODUCTION

Federated learning (FL) trains a shared model without centralizing client data. Byzantine clients may submit poisoned updates (Bhagoji et al., 2019) or train on backdoored local data and use model replacement to implant a global backdoor (Bagdasaryan et al., 2020). A defense must decide whose contributions to admit while preserving useful participation from heterogeneous benign clients.

False rejection and false acceptance. A defense can reject a benign contribution or accept a harmful one. Under non-IID data, atypical benign updates can be difficult to distinguish from malicious ones and may therefore be suppressed by distance-based selection or reference-based reweighting (Blanchard et al., 2017; Yin et al., 2018; Cao et al., 2021; Yan et al., 2023). Repeated exclusion can alter the effective training population (Cho et al., 2022); relaxing the criterion may instead admit attacks designed to resemble benign updates (Fang et al., 2020; El Mhamdi et al., 2018). We study how aggregate-level performance feedback can inform this joint decision.

Protected updates restrict the evidence available to a defense. Individual updates can reveal training information (Zhu et al., 2019; Nasr et al., 2019). Differential privacy (DP), homomorphic

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.

encryption (HE), and secure multi-party computation (MPC) respectively perturb, encrypt, or mask communicated updates, changing the statistics available to the server. Privacy-compatible defenses therefore need explicit assumptions about the operations and information their protocols expose (So et al., 2021; Chen et al., 2024; Jahani-Nezhad et al., 2025).

Early selection errors can become self-reinforcing. Assessments based on historical information do not necessarily resolve this distinction, and using their outcomes to guide participation can make early errors self-reinforcing. FLDetector compares received client updates with predictions derived from past updates (Zhang et al., 2022), whereas MAB-RFL uses detection results to guide subsequent participation through Thompson Sampling (Wan et al., 2022). FedPBF samples candidate groups solely from accumulated historical membership preferences and updates those preferences according to the winning group (Tang et al., 2023). This mechanism may reinforce previously favored clients and further reduce exploration of benign clients overlooked early. However, when benign variation and malicious behavior produce similar update statistics, historical assessments built from those statistics may remain unreliable. If an early negative assessment lowers a benign client's probability of selection, fewer subsequent updates from that client become available to correct the assessment. We call this feedback loop a *Matthew effect in client participation*. It can suppress a misjudged benign client's participation across multiple rounds, reducing its useful contribution to global training.

Our approach. BRFL-TS evaluates the utility of candidate aggregates instead of inspecting individual update statistics. Normalized Thompson samples generate M client groups. Clients score the corresponding models, and the server uses median scores for group ranking and adaptive client-level reward and penalty assignment. Reward-set reevaluation and rollback determine the retained global model. This provides a common decision rule for different update-protection mechanisms, whose privacy properties still depend on their complete protocols. Our contributions are as follows.

- **Utility-driven adaptive client selection.** We design adaptive credit assignment that translates candidate-model evaluations into client-level rewards and penalties using membership across utility-ranked groups. The resulting feedback guides aggregation and future client selection to suppress malicious updates while preserving opportunities to reassess benign clients.
- **Theoretical guarantees for renewed feedback.** For the weighted subset sampler and trust-update process, we prove positive lower bounds on each client's probability of receiving renewed reward or penalty feedback within finite training windows, together with state-dependent upper bounds on the expected waiting time during continued training. These guarantees establish that prior penalties do not eliminate a client's opportunities for renewed feedback, providing a theoretical basis for continued reassessment.
- **Robustness and client-level decision quality.** Our evaluation spans four datasets and eight untargeted poisoning attacks, covering data heterogeneity, Byzantine-client ratios, and update protection. In targeted client-level comparisons, BRFL-TS achieves the highest test accuracy among the evaluated defenses, together with low benign false rejection and restrained Byzantine false acceptance in its reward-and-penalty feedback.

2 Related Work

2.1 Robust Aggregation and Reliability Assessment

Statistically robust aggregation limits malicious influence using distances, coordinate statistics, or agreement with a trusted reference. Krum selects an update using nearest-neighbour distances (Blanchard et al., 2017) and RFA uses an approximate geometric median (Pillutla et al., 2022), whereas coordinate-wise Median and Trimmed Mean reduce the influence of extreme values through median estimation and trimmed averaging, respectively (Yin et al., 2018). FLTrust weights and normalizes updates against a trusted server update computed on clean data (Cao et al., 2021), while LASA refines screening through sparsification and layer-wise magnitude and direction checks (Xu et al., 2025b). Heterogeneous data can nevertheless make useful benign updates appear anomalous, creating a risk of suppressing their contributions (Yan et al., 2023). Applying statistical rules to protected updates also requires suitable adaptations. Robust-DPFL accounts for DP noise when screening gra-

dients (Qi et al., 2024), AegisFL computes update statistics under HE (Chen et al., 2024), and RFA approximates the geometric median through repeated secure averaging (Pillutla et al., 2022).

Model-evaluation methods instead assess update quality through the performance of the resulting models. Zeno scores individual candidate updates by estimated loss reduction on server-side data and a magnitude penalty (Xie et al., 2019), whereas SLVR uses MPC to evaluate individual client models on other clients' secret-shared validation data and filters updates using the resulting scores (Choi et al., 2025). At the group level, FedGT tests models securely aggregated from overlapping client groups and decodes the binary outcomes to identify malicious clients in cross-silo FL (Xhemrishi et al., 2025). BRFL-TS also evaluates group aggregates, but uses their client-reported utility to assign rewards and penalties that guide aggregation and subsequent client selection. By relying on evaluation scores rather than individual plaintext updates, BRFL-TS can retain the same client-selection rule across federated learning systems that protect updates using DP, HE, or MPC.

2.2 History-Aware Defense and Client Selection

History-based defenses assess client behavior over multiple rounds rather than relying on the current round alone. FoolsGold reduces the weights of clients with highly similar accumulated gradients to counter sybil attacks (Fung et al., 2020), while FLDetector (Zhang et al., 2022) and FLANDERS (Gabrielli et al., 2025) detect anomalies by comparing received updates with predictions based on earlier rounds, with FLANDERS using matrix autoregression. RECESS actively tests clients with constructed gradients, updates trust scores from their responses, and uses these scores to determine aggregation weights (Yan et al., 2023).

Reliability assessments can also determine which clients are selected for later rounds. SCS selects clients according to reputation and recruitment costs through stochastic integer programming (Tan et al., 2024), whereas MAB-RFL updates client-specific Beta distributions from Byzantine-detection results and uses Thompson Sampling to select clients for subsequent rounds (Wan et al., 2022). When past detection results determine future participation, an early misjudgment can reduce a benign client's selection probability, leaving fewer new observations with which to revise that judgment. BRFL-TS instead assigns rewards and penalties by considering which candidate groups include each client and how well their aggregate models perform, then uses this feedback to update trust for future selection. BRFL-TS uses these utility-based trust updates to guide Thompson-weighted subset sampling, reducing penalized clients' selection chances while preserving opportunities to reassess them.

3 Problem Setup

Threat Model. We consider a federated learning system with a central server and K clients indexed by $[K] = \{1, \dots, K\}$. The clients are partitioned into a benign set $\mathcal{G}$ and a Byzantine set $\mathcal{F}$, where $[K] = \mathcal{G} \dot{\cup} \mathcal{F}$ and $|\mathcal{F}| < K/2$. Byzantine clients may collude and arbitrarily manipulate their local training and uploaded updates to degrade global-model utility. They may also fabricate the scores they report for candidate models.

The server follows the protocol but may inspect the information it receives, as in an honest-but-curious threat model. A passive communication eavesdropper may also observe transmitted information. Either may attempt membership inference or gradient inversion using information exposed under the chosen update-protection mechanism. The information available to these privacy adversaries depends on that mechanism.

Client-Level Decision Errors. At round t, let $\mathcal{D}_t^+$ and $\mathcal{D}_t^-$ denote the clients explicitly accepted and rejected by a defense, respectively, and let $\mathcal{D}_t = \mathcal{D}_t^+ \cup \mathcal{D}_t^-$, with $\mathcal{D}_t^+ \cap \mathcal{D}_t^- = \emptyset$. For BRFL-TS, acceptance and rejection correspond to reward and penalty feedback. We define the false rejection rate (FRR) and false acceptance rate (FAR) over client-round decision events.

$$\mathrm{FRR} = \frac{\sum_t |\mathcal{D}_t^- \cap \mathcal{G}|}{\sum_t |\mathcal{D}_t \cap \mathcal{G}|}, \qquad \mathrm{FAR} = \frac{\sum_t |\mathcal{D}_t^+ \cap \mathcal{F}|}{\sum_t |\mathcal{D}_t \cap \mathcal{F}|}. \tag{1}$$

Clients outside $\mathcal{D}_t$ are undecided and are excluded from these rates. A rate is undefined if its denominator is zero.

Defense Objective. BRFL-TS aims to limit Byzantine influence while retaining useful benign contributions and sustaining global-model utility as data heterogeneity and the Byzantine-client fraction vary, including when adversaries manipulate updates or candidate-model scores. The defense is designed around three requirements. (1) *Reducing client-level decision errors.* When heterogeneous benign updates and stealthy poisoning make reliability assessment difficult, it should limit both false rejection of benign clients and false acceptance of Byzantine clients. (2) *Preserving reassessment opportunities.* Clients receiving early negative feedback should remain eligible for inclusion in later candidate groups, allowing new utility feedback to revise their reliability estimates. (3) *Remaining compatible with protected updates.* The same selection and feedback logic should remain usable when updates are protected through DP, HE, or MPC, without requiring inspection of individual plaintext updates.

4 METHODOLOGY

4.1 THE BRFL-TS FRAMEWORK

METHOD OVERVIEW

To suppress malicious updates while retaining useful benign contributions, BRFL-TS assesses client reliability through the utility of models aggregated from client subsets. We adopt a combinatorial-bandit perspective to guide client-subset selection across rounds using this utility feedback, treating individual clients as base arms and candidate subsets as combinatorial arms. The utility observed for a combinatorial arm reflects the joint effect of its clients' updates, whereas aggregation and future participation require client-level decisions. BRFL-TS addresses this credit-assignment problem with an adaptive reward-and-penalty mechanism, using current trust to adjust the number of penalties in the lowest-ranked group and membership across utility-ranked groups to determine who receives them. Each client's assigned rewards and penalties update its Beta trust distribution, which summarizes the feedback accumulated for that client. These distributions guide Thompson-weighted group sampling while retaining opportunities to reassess penalized clients. Rewarded clients provide the provisional global update, whose acceptance is determined by comparison with the current model's utility. Figure 1 illustrates the framework. Appendix A.1 summarizes the multiple-play bandit background, and Algorithm 1 in Appendix A.2 gives the complete procedure.

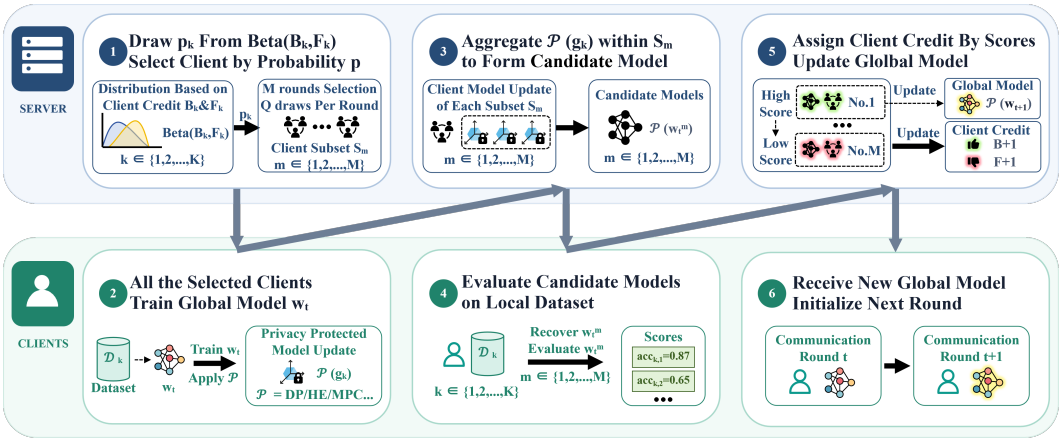


Figure 1: Schematic of BRFL-TS. The Beta sample shown as p_k denotes the unnormalized θ_k; candidate draws use normalized weights. The No. 1 candidate is provisional: reward-set aggregation and the rollback check determine the retained model (Algorithm 1).

THOMPSON-WEIGHTED CLIENT SELECTION

Client selection must exploit accumulated reliability evidence without allowing early negative assessments to irreversibly exclude benign clients. BRFL-TS therefore maintains a trust distribution $\mathrm{Beta}(B_k, F_k)$ for each client k, initialized with $B_k = F_k = 1$, where the two parameters accumulate reward and penalty feedback, respectively. At round t, the server draws and normalizes one

Thompson sample per client,

$$\theta_k \sim \mathrm{Beta}(B_k, F_k), \qquad p_k = \frac{\theta_k}{\sum_{j=1}^{K} \theta_j}. \tag{2}$$

Selecting only the largest samples could sharply reduce the participation of benign clients that perform poorly in early rounds because of data heterogeneity, limited computation, or unstable communication. Instead, BRFL-TS performs M independent weighted samplings without replacement, each time selecting Q clients from the full set of K clients according to $\{p_k\}_{k=1}^{K}$ to form $S_1, \ldots, S_M$. This design favors clients with stronger accumulated evidence while retaining opportunities to reassess clients whose current trust is low or uncertain. Rewards and penalties subsequently shift their trust distributions and hence their future sampling weights, forming a feedback loop between observed utility and participation. Section 4.2 further characterizes these opportunities for reassessment.

Candidate-Model Construction and Utility Evaluation

Let $\mathcal{S} = \bigcup_{m=1}^{M} S_m$ denote the union of selected clients. Each client $k \in \mathcal{S}$ trains from the current global model w_t, obtains a local model w_t^k, and computes the update $g_k = w_t - w_t^k$. Let $\mathcal{P}$ denote the update-protection and aggregation interface of the chosen DP, HE, or MPC mechanism. Each selected client uploads $\mathcal{P}(g_k)$, and the server constructs a candidate model for every group S_m:

$$\mathcal{P}(G_m) = \frac{1}{|S_m|} \sum_{k \in S_m} \mathcal{P}(g_k), \qquad \mathcal{P}(w_t^m) = \mathcal{P}(w_t) - \mathcal{P}(G_m). \tag{3}$$

BRFL-TS treats these client groups as combinatorial arms and uses the learning utility of their candidate models as arm observations. The server transmits the candidate models to all K clients. Each client k obtains $w_{t,k}^m = \mathrm{Recover}_{\mathcal{P},k}(\mathcal{P}(w_t^m))$, evaluates it on its local data, and returns $acc_{k,m}$. Because Byzantine clients may fabricate evaluation scores, the server uses the median score as the utility of each candidate and identifies the highest- and lowest-utility groups:

$$\begin{aligned} med_m &= \mathrm{median}(\{acc_{k,m}\}_{k=1}^{K}), \\ m^* &= \arg\max_m med_m, \qquad m^\circ = \arg\min_m med_m. \end{aligned} \tag{4}$$

This construction shifts reliability evidence from individual protected updates to the learning utility of candidate aggregates. The chosen privacy mechanism carries out the required subset aggregation and candidate-model delivery, whereas the server bases its ranking and credit-assignment decisions only on the returned evaluation scores. Consequently, provided that the underlying protocol supports aggregation over sampled subsets, the selection and credit-assignment rules remain unchanged across protection mechanisms.

Adaptive Credit Assignment and Trust Updates

Candidate utilities provide group-level observations, whereas subsequent sampling and aggregation require client-level decisions. This creates a credit-assignment problem. BRFL-TS treats the clients in S_{m^*} as reward candidates and those in S_{m° as penalty candidates, and then determines the penalty count and recipients separately.

A low group utility reflects the joint effect of its members rather than establishing that every member is unreliable. Moreover, as accumulated feedback reduces the participation of consistently low-utility clients, a fixed penalty quota may increasingly penalize benign members of the lowest-ranked group. BRFL-TS therefore uses the current Thompson samples of that group to adapt the penalty count. For each $k \in S_{m^\circ}$, the server performs three independent Bernoulli trials with success probability $1 - \theta_k$ and computes

$$\kappa = \left\lfloor \frac{1}{3} \sum_{k \in S_{m^\circ}} \sum_{i=1}^{3} \mathbf{1}_{\mathrm{Bernoulli}(1-\theta_k)=1} \right\rfloor. \tag{5}$$

Repeating the trial reduces the sensitivity of the quota to a single random outcome. Before rounding, its conditional expectation is $\sum_{k \in S_{m^\circ}} (1 - \theta_k)$, so stronger trust within the group leads to a smaller expected penalty quota.

After determining how many clients to penalize, BRFL-TS uses their membership across the utility-ranked candidate groups to determine whom to penalize. The reason is that a client in the lowest-utility group may also occur in other combinations: appearing in higher-utility groups provides stronger evidence that the client may have made a beneficial contribution, whereas appearing primarily in lower-utility groups provides weaker evidence of such a contribution. Let $j_{(1)}, \ldots, j_{(M)}$ denote the candidate-group indices sorted by med_m in descending order. For each $k \in S_{m^\circ}$, define

$$s_k[r] = \begin{cases} 1, & k \in S_{j_{(r)}}, \\ 0, & \text{otherwise}, \end{cases} \qquad r \in [M]. \tag{6}$$

The server sorts these membership sequences in ascending lexicographical order and assigns the first κ corresponding clients to the penalty set $\mathcal{J}$. Because earlier positions correspond to higher-utility groups, membership in such a group lowers a client's penalty priority. The remaining members of the highest-utility group form the reward set $\mathcal{R} = S_{m^*} \setminus \mathcal{J}$, and the server updates the trust counters as

$$\begin{aligned} B_k &\leftarrow B_k + 1 \quad \text{for } k \in \mathcal{R}, \\ F_k &\leftarrow F_k + 1 \quad \text{for } k \in \mathcal{J}. \end{aligned} \tag{7}$$

These updates translate comparisons among candidate groups into client-level evidence: clients repeatedly associated with high-utility candidates receive larger future sampling weights, whereas those repeatedly associated with low-utility candidates are gradually suppressed.

The server then aggregates the updates in the reward set to construct a provisional global model,

$$\mathcal{P}(G_{\mathcal{R}}) = \frac{1}{|\mathcal{R}|} \sum_{k \in \mathcal{R}} \mathcal{P}(g_k), \qquad \mathcal{P}(\widehat{w}_{t+1}) = \mathcal{P}(w_t) - \mathcal{P}(G_{\mathcal{R}}). \tag{8}$$

If $\mathcal{R} = S_{m^*}$, the provisional model is the previously evaluated highest-utility candidate and reuses med_{m^*} as its score. If penalty removal changes the group, all clients evaluate the provisional model and the median of their scores gives $\widehat{a}_{t+1}$:

$$\widehat{a}_{t+1} = \begin{cases} med_{m^*}, & \mathcal{R} = S_{m^*}, \\ \text{median}(\{\widehat{acc}_{k,t+1}\}_{k=1}^{K}), & \mathcal{R} \neq S_{m^*}. \end{cases} \tag{9}$$

The candidate ranking establishes only relative utility within the current round and does not ensure that the provisional model improves on the retained global model. Let a_t be the retained evaluation score of w_t. For a threshold $\tau > 0$, the server applies

$$(\mathcal{P}(w_{t+1}), a_{t+1}) = \begin{cases} (\mathcal{P}(w_t), a_t), & \widehat{a}_{t+1} < a_t - \tau, \\ (\mathcal{P}(\widehat{w}_{t+1}), \widehat{a}_{t+1}), & \text{otherwise}. \end{cases} \tag{10}$$

Trust updates and rollback serve different purposes: the former retain relative feedback among the current candidates to guide future participation, while the latter prevents a harmful provisional update from replacing the current model. Together, they allow BRFL-TS to learn client contributions across rounds while limiting their immediate impact on the global model.

4.2 Theoretical Analysis

To quantify opportunities to reevaluate previously penalized clients, we analyze BRFL-TS's feedback process under the Thompson-weighted sampling and trust-update rules in Section 4.1. We establish a positive lower bound on the probability of renewed reward or penalty feedback within a finite training window and an upper bound on the expected waiting time during continued training, both determined by the client's current trust state. We consider fixed K, M, and Q, with all clients remaining eligible throughout training. Trust counters start positive and change only through reward or penalty updates, which are retained after rollback.

To make these bounds explicit, consider client i's trust state after round t. Let $\mathcal{H}_t$ denote the training history up to that point, and let $B_i(t)$ and $F_i(t)$ be its reward and penalty counters at that time. During continued execution, let $D_i(t)$ count the additional completed rounds until its next reward or penalty update. We bound this waiting time using a lower bound on the per-round feedback probability. Until the client's next counter update, this lower bound is

$$c_i(t) = \frac{1}{K^M} \prod_{\ell=0}^{M-1} \frac{B_i(t) + \ell}{B_i(t) + F_i(t) + \ell} > 0. \tag{11}$$

Theorem 1 (State-dependent feedback return)**.** *Under the sampling and counter-update rules above, for every client i and integer $H \geq 1$,*

$$\mathbb{P}\big(D_i(t) \leq H \mid \mathcal{H}_t\big) \geq 1 - (1 - c_i(t))^H, \qquad \mathbb{E}[D_i(t) \mid \mathcal{H}_t] \leq \frac{1}{c_i(t)}. \tag{12}$$

The probability bound applies to any H additional training rounds executed under the stated rules, and the expectation refers to continued execution. The full assumptions and proof are given in Appendix B.2. The key to the single-round bound is the event that client i is drawn first in every candidate group. It must then belong to the highest-scoring group and, under the credit-assignment rule, receive a reward or penalty update regardless of the evaluation scores. Conditional on the full vector of Beta samples, the first draw of each group selects client i with probability $\theta_i / \sum_k \theta_k \geq \theta_i / K$, where θ_i is its sample. Conditional independence across candidate draws and the Beta moment identity then yield $c_i(t)$ as a lower bound on its feedback probability for that round.

While client i awaits new feedback, its own counters remain unchanged, so the same single-round lower bound applies throughout this period even as other clients' states evolve. Iterating this conditional bound gives the finite-window probability lower bound. Summing the resulting tail probabilities yields the expected waiting-time upper bound.

This result shows how BRFL-TS combines continued exploration with utility-based trust revision. Thompson-weighted sampling preserves opportunities to reevaluate previously penalized clients, while adaptive credit assignment converts new group-level utility evidence into individual Beta trust updates that guide subsequent selection. For an initially misjudged benign client, later rewards can improve its trust, allowing participation decisions to respond to new utility evidence and supporting BRFL-TS's design for mitigating the Matthew effect and retaining useful benign contributions.

Appendix B.1 gives a complementary conditional analysis that relates winning-group purity and benign-client coverage over finite epochs to reference regret and a per-round winning-group inclusion floor. It also derives a finite-horizon, though potentially loose, inclusion floor for the actual sampler under the stated assumptions.

5 Experiments

We evaluate whether BRFL-TS improves model utility while reducing erroneous rejection of benign clients and erroneous acceptance of Byzantine clients. The experiments also examine its ability to mitigate participation bias and revise early negative assessments of benign clients, together with its robustness under varying attack conditions and compatibility with update protection.

5.1 Experimental Setup

Datasets and models. We evaluate BRFL-TS on MNIST with logistic regression and Fashion-MNIST with LeNet, each using $K = 100$ clients. CIFAR-10 uses AlexNet and Tiny-ImageNet uses ViT-Base-Patch16-224, each with $K = 20$ clients. We consider IID partitions and non-IID partitions generated from a Dirichlet distribution with concentration parameter β.

Attacks. We evaluate eight untargeted attacks spanning data and model poisoning. Label Flipping corrupts local labels, while Sign Flipping reverses update signs. I-FMPA optimizes malicious model parameters (Zhang et al., 2023), Min-Max and Min-Sum construct malicious updates under distance constraints derived from benign updates (Shejwalkar & Houmansadr, 2021), and Fang-v1 and Fang-v2 (Fang et al., 2020) target robust aggregation rules. PoisonedFL (Xie et al., 2025) maintains consistency in malicious updates across rounds.

Baselines. For model utility on MNIST, Fashion-MNIST, and Tiny-ImageNet, we use $\beta = 0.5$ with 20% Byzantine clients and compare BRFL-TS with Multi-Krum (Blanchard et al., 2017), FLTrust (Cao et al., 2021), RFA (Pillutla et al., 2022), Fed-NGA (Zuo et al., 2025), and RandEigen (Lee et al., 2025). On CIFAR-10, the client-level decision comparisons include FLANDERS (Gabrielli et al., 2025), MAB-RFL (Wan et al., 2022), FedDMC (Mu et al., 2024), AlignIns (Xu et al., 2025a), and MSGuard (Yang et al., 2025). FedAvg without attacks provides a reference for model utility.

Update protection. To assess compatibility with update protection, we evaluate model utility while retaining the same client-selection and feedback rules across DP, HE, and MPC settings. The DP setting clips client updates before adding Gaussian noise (Geyer et al., 2017), while HE uses CKKS (Cheon et al., 2017) and MPC uses canceling additive masks (Bonawitz et al., 2017). Dataset-specific clipping and noise settings and full utility results for MNIST, Fashion-MNIST, and Tiny-ImageNet are provided in Appendix C.3.

Evaluation metrics. Average model performance, denoted by AMP, is computed as the mean test accuracy over the eight attacks and reported with the standard deviation of these accuracies. The false rejection rate FRR and false acceptance rate FAR measure erroneous rejection of benign clients and erroneous acceptance of Byzantine clients, respectively. Communication overhead is assessed analytically from model and score payloads in plaintext-equivalent bytes, as detailed in Appendix C.4.

5.2 Model Utility under Byzantine Attacks

We assess model utility on MNIST, Fashion-MNIST, and Tiny-ImageNet, evaluating each dataset under the eight attacks introduced in Section 5.1, which span data and model poisoning. Table 1 reports results without update protection, using non-IID partitions with $\beta = 0.5$ and 20% Byzantine clients. BRFL-TS achieves the highest attack-averaged accuracy on all three datasets, with the largest improvement on Fashion-MNIST.

Table 1: Test accuracy on MNIST, Fashion-MNIST, and Tiny-ImageNet under the eight untargeted attacks introduced in Section 5.1, covering data and model poisoning. Results are reported as the mean and standard deviation across attacks. All three datasets use non-IID partitions with $\beta = 0.5$ and 20% Byzantine clients. Updates are unprotected. FedAvg W.O. is the attack-free reference.

Dataset	Multi-Krum	FLTrust	RFA	Fed-NGA	RandEigen	BRFL-TS	FedAvg W.O.
MNIST	88.36 ± 1.13	89.35 ± 1.22	85.65 ± 2.21	84.17 ± 5.69	79.23 ± 5.09	$\mathbf{91.68 \pm 0.31}$	91.79 ± 0.08
Fashion-MNIST	68.26 ± 24.24	77.21 ± 0.30	69.90 ± 24.30	70.48 ± 24.73	56.26 ± 31.50	$\mathbf{85.22 \pm 1.14}$	87.13 ± 0.16
Tiny-ImageNet	88.70 ± 9.22	92.65 ± 0.99	91.53 ± 2.04	70.35 ± 26.68	60.05 ± 31.55	$\mathbf{92.98 \pm 0.98}$	94.20 ± 0.57

Under HE and MPC, BRFL-TS uses the same selection and feedback rules and maintains accuracies within 1.09 percentage points of its unprotected results on all three datasets. BRFL-TS also demonstrates strong tolerance to DP noise, attaining high accuracy across multiple evaluated noise settings and the highest mean accuracy among the evaluated defenses in nine of twelve dataset–noise combinations. Full results are provided in Appendix C.3.

Evaluation overhead. BRFL-TS obtains the utility feedback used for selection by distributing multiple candidate models for local evaluation, incurring additional communication and computation. For $M = 10$ with negligible score payloads, Appendix C.4 derives an approximate plaintext-equivalent communication upper bound of 6.5 times a full-participation baseline, excluding cryptographic protocol overhead.

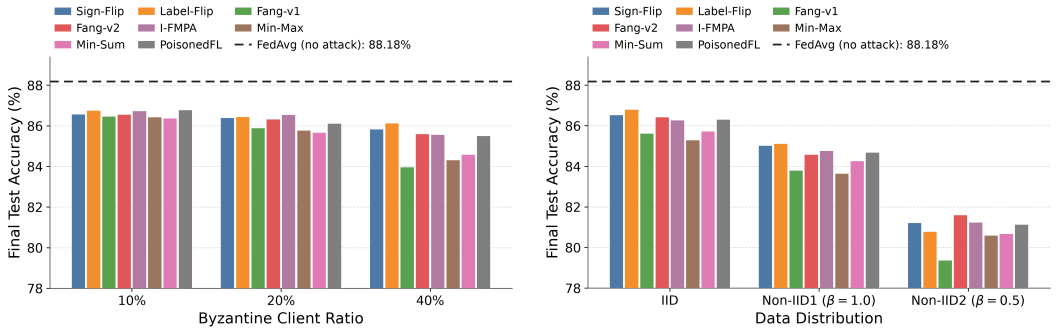
5.3 Client-Level Decisions and Participation

We evaluate whether BRFL-TS reduces both the false rejection of benign clients and the false acceptance of Byzantine clients. The CIFAR-10 evaluation tests Sign Flipping and PoisonedFL under both IID and non-IID partitions, yielding four settings. The non-IID partition uses $\beta = 0.5$. For BRFL-TS and MAB-RFL, FRR and FAR are measured from penalty and reward feedback, respectively.

Table 2 shows that BRFL-TS achieves the highest test accuracy in all four settings. Under Sign Flipping, BRFL-TS has zero FAR and FRRs of 3.23% and 5.66%, whereas MAB-RFL's slightly lower FRRs accompany 100% FAR and 10% accuracy, showing that FRR alone is insufficient. Under PoisonedFL, BRFL-TS keeps both error rates below 9%; in the non-IID case, it improves over MSGuard in accuracy (80.69% versus 78.89%), FRR (8.98% versus 19.85%), and FAR (7.79% versus 15.89%).

Table 2: Client-level decision errors and final test accuracy on CIFAR-10 (%). FRR and FAR exclude client-rounds without explicit decisions. BRFL-TS feedback is counted before rollback; participation budgets vary across methods.

Method	Sign Flipping IID			Sign Flipping Non-IID			PoisonedFL IID			PoisonedFL Non-IID		
	FRR↓	FAR↓	Acc↑	FRR↓	FAR↓	Acc↑	FRR↓	FAR↓	Acc↑	FRR↓	FAR↓	Acc↑
Counter-feedback events (before rollback)												
MAB-RFL	**3.22**	100.00	10.00	**2.09**	100.00	10.00	86.50	100.00	9.60	82.89	100.00	9.60
BRFL-TS (Ours)	3.23	**0.00**	**85.98**	5.66	**0.00**	**79.29**	**5.43**	**6.90**	**86.42**	**8.98**	**7.79**	**80.69**
Aggregation-decision events												
FLANDERS	16.95	72.73	10.00	15.38	75.00	10.00	20.34	90.91	10.31	20.00	100.00	10.00
FedDMC	11.70	72.90	10.00	12.72	79.44	10.00	13.74	82.24	10.00	13.23	78.50	10.00
AlignIns	18.32	8.41	28.30	18.07	8.41	10.00	17.56	8.41	85.20	18.32	8.41	79.87
MSGuard	18.07	4.67	83.14	11.70	7.48	78.18	18.32	19.63	84.80	19.85	15.89	78.89



(a) Varying Byzantine-client ratio.

(b) Varying data heterogeneity.

Figure 2: Robustness of BRFL-TS on CIFAR-10 under eight Byzantine attacks.

Appendix C.1 examines whether low decision-error rates translate into broad benign feedback. Across the four Sign Flipping and PoisonedFL settings, all 16 benign clients receive BRFL-TS rewards, whereas each Byzantine client receives at most three; FedPBF leaves one or two benign clients without winning-group membership in two IID settings. BRFL-TS also reduces benign-feedback concentration in three settings—for example, the IID Sign Flipping Gini coefficient falls from 0.382 to 0.093—and improves accuracy by 0.51–1.84 percentage points in all four. Paired Fang-v2 snapshots further show that all five benign clients penalized by round 10 later receive rewards and attain higher trust means by round 100.

5.4 Robustness to Heterogeneity and Byzantine Behavior

On CIFAR-10 under all eight attacks, we vary (1) the Byzantine-client ratio from 10% to 40% under IID partitioning, and (2) the data distribution from IID to Non-IID1 ($\beta = 1.0$) and Non-IID2 ($\beta = 0.5$) at a fixed 20% Byzantine ratio. We also evaluate Byzantine clients that falsify candidate-model scores under the IID ratio sweep (Appendix C.2).

Figure 2 shows stable accuracy as the Byzantine ratio increases: at 40%, BRFL-TS obtains 83.96%–86.12% across eight attacks (85.19% on average). Under Non-IID2 ($\beta = 0.5$) with 20% Byzantine clients, it obtains 79.37%–81.60% (80.83% on average). With falsified scores at 40%, the additional loss is at most 0.6 points and candidate-purity detection is 99.7% (Appendix C.2).

6 Conclusion

BRFL-TS converts candidate-model utility into trust feedback for Thompson-weighted selection, suppressing Byzantine contributions while enabling benign-client reassessment. We establish finite-window feedback and waiting-time guarantees. BRFL-TS attains the highest accuracy among the evaluated defenses with low benign false rejection and Byzantine false acceptance. Future work extends this approach to Byzantine-robust multi-agent collaboration.

REFERENCES

Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2938–2948. PMLR, 2020. URL https://proceedings.mlr.press/v108/bagdasaryan20a.html.

Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 634–643. PMLR, 2019. URL https://proceedings.mlr.press/v97/bhagoji19a.html.

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, volume 30, pp. 119–129. Curran Associates, Inc., 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/f4b9ec30ad9f68f89b29639786cb62ef-Abstract.html.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191. ACM, 2017. doi: 10.1145/3133956.3133982. URL https://doi.org/10.1145/3133956.3133982.

Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. FLTrust: Byzantine-robust federated learning via trust bootstrapping. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. The Internet Society, 2021. doi: 10.14722/ndss.2021.24434. URL https://doi.org/10.14722/ndss.2021.24434.

Dong Chen, Hongyuan Qu, and Guangwu Xu. AegisFL: Efficient and flexible privacy-preserving byzantine-robust cross-silo federated learning. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 7207–7219. PMLR, 2024. URL https://proceedings.mlr.press/v235/chen24ag.html.

Jung Hee Cheon, Andrey Kim, Miran Kim, and Yong Soo Song. Homomorphic encryption for arithmetic of approximate numbers. In *Advances in Cryptology – ASIACRYPT 2017*, volume 10624 of *Lecture Notes in Computer Science*, pp. 409–437. Springer, 2017. doi: 10.1007/978-3-319-70694-8_15. URL https://doi.org/10.1007/978-3-319-70694-8_15.

Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Towards understanding biased client selection in federated learning. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 10351–10375. PMLR, 2022. URL https://proceedings.mlr.press/v151/jee-cho22a.html.

Jihye Choi, Sai Rahul Rachuri, Ke Wang, Somesh Jha, and Yizhen Wang. SLVR: Securely leveraging client validation for robust federated learning. *arXiv preprint arXiv:2502.08055*, 2025. URL https://arxiv.org/abs/2502.08055.

El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. The hidden vulnerability of distributed learning in byzantium. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *PMLR*, pp. 3521–3530. PMLR, 2018. URL https://proceedings.mlr.press/v80/mhamdi18a.html.

Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Local model poisoning attacks to byzantine-robust federated learning. In *29th USENIX Security Symposium (USENIX Security 20)*, pp. 1605–1622. USENIX Association, 2020. ISBN 978-1-939133-17-5. URL https://www.usenix.org/conference/usenixsecurity20/presentation/fang.

Clement Fung, Chris J. M. 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. USENIX Association, 2020. ISBN 978-1-939133-18-2. URL https://www.usenix.org/conference/raid2020/presentation/fung.

Edoardo Gabrielli, Dimitri Belli, Zoe Matrullo, Vittorio Miori, and Gabriele Tolomei. Securing federated learning against extreme model poisoning attacks via multidimensional time series anomaly detection on local updates. *IEEE Transactions on Information Forensics and Security*, 20:9610–9624, 2025. doi: 10.1109/TIFS.2025.3608671. URL https://doi.org/10.1109/TIFS.2025.3608671.

Robin C. Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017. URL https://arxiv.org/abs/1712.07557.

Tayyebeh Jahani-Nezhad, Mohammad Ali Maddah-Ali, and Giuseppe Caire. ByzSecAgg: A byzantine-resistant secure aggregation scheme for federated learning based on coded computing and vector commitment. *IEEE Transactions on Information Theory*, 71(8):6410–6424, 2025. doi: 10.1109/TIT.2025.3577019. URL https://doi.org/10.1109/TIT.2025.3577019.

De Zhang Lee, Aashish Kolluri, Prateek Saxena, and Ee-Chien Chang. A practical and secure byzantine robust aggregator. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 2877–2891. Association for Computing Machinery, 2025. doi: 10.1145/3719027.3744851. URL https://doi.org/10.1145/3719027.3744851.

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. doi: 10.1109/TDSC.2024.3372634. URL https://doi.org/10.1109/TDSC.2024.3372634.

Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 739–753. IEEE, 2019. doi: 10.1109/SP.2019.00065. URL https://doi.org/10.1109/SP.2019.00065.

Krishna Pillutla, Sham M. Kakade, and Zaïd Harchaoui. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022. doi: 10.1109/TSP.2022.3153135. URL https://doi.org/10.1109/TSP.2022.3153135.

Tao Qi, Huili Wang, and Yongfeng Huang. Towards the robustness of differentially private federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19911–19919, 2024. doi: 10.1609/aaai.v38i18.29967. URL https://doi.org/10.1609/aaai.v38i18.29967.

Virat Shejwalkar and Amir Houmansadr. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2021. doi: 10.14722/ndss.2021.24498. URL https://doi.org/10.14722/ndss.2021.24498.

Jinhyun So, Basak Güler, and A. Salman Avestimehr. Byzantine-resilient secure federated learning. *IEEE Journal on Selected Areas in Communications*, 39(7):2168–2181, 2021. doi: 10.1109/JSAC.2020.3041404. URL https://doi.org/10.1109/JSAC.2020.3041404.

Xavier Tan, Wei Chong Ng, Wei Yang Bryan Lim, Zehui Xiong, Dusit Niyato, and Han Yu. Reputation-aware federated learning client selection based on stochastic integer programming. *IEEE Transactions on Big Data*, 10(6):953–964, 2024. doi: 10.1109/TBDATA.2022.3191332. URL https://doi.org/10.1109/TBDATA.2022.3191332.

Xingxing Tang, Hanlin Gu, Lixin Fan, and Qiang Yang. Achieving provable byzantine fault-tolerance in a semi-honest federated learning setting. In *Proceedings of the 27th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, volume 13936 of *LNCS*, pp. 415–427. Springer, 2023. doi: 10.1007/978-3-031-33377-4_32. URL https://doi.org/10.1007/978-3-031-33377-4_32.

Wei Wan, Shengshan Hu, Jianrong Lu, Leo Yu Zhang, Hai Jin, and Yuanyuan He. Shielding federated learning: Robust aggregation with adaptive client selection. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 753–760. International Joint Conferences on Artificial Intelligence Organization, 2022. doi: 10.24963/ijcai.2022/106. URL https://www.ijcai.org/proceedings/2022/106.

Marvin Xhemrishi, Johan Östman, Antonia Wachter-Zeh, and Alexandre Graell i Amat. FedGT: Identification of malicious clients in federated learning with secure aggregation. *IEEE Transactions on Information Forensics and Security*, 20:2577–2592, 2025. doi: 10.1109/TIFS.2025.3539964. URL https://doi.org/10.1109/TIFS.2025.3539964.

Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *PMLR*, pp. 6893–6901. PMLR, 2019. URL https://proceedings.mlr.press/v97/xie19b.html.

Yueqi Xie, Minghong Fang, and Neil Zhenqiang Gong. Model poisoning attacks to federated learning via multi-round consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15454–15463, 2025. doi: 10.1109/CVPR52734.2025.01440. URL https://openaccess.thecvf.com/content/CVPR2025/html/Xie_Model_Poisoning_Attacks_to_Federated_Learning_via_Multi-Round_Consistency_CVPR_2025_paper.html.

Jiahao Xu, Zikai Zhang, and Rui Hu. Detecting backdoor attacks in federated learning via direction alignment inspection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20654–20664, 2025a. URL https://openaccess.thecvf.com/content/CVPR2025/html/Xu_Detecting_Backdoor_Attacks_in_Federated_Learning_via_Direction_Alignment_Inspection_CVPR_2025_paper.html.

Jiahao Xu, Zikai Zhang, and Rui Hu. Achieving byzantine-resilient federated learning via layer-adaptive sparsified model aggregation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1508–1517, 2025b. URL https://openaccess.thecvf.com/content/WACV2025/html/Xu_Achieving_Byzantine-Resilient_Federated_Learning_via_Layer-Adaptive_Sparsified_Model_Aggregation_WACV_2025_paper.html.

Haonan Yan, Wenjing Zhang, Qian Chen, Xiaoguang Li, Wenhai Sun, Hui Li, and Xiaodong Lin. RECESS vaccine for federated learning: Proactive defense against model poisoning attacks. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8702–8713, 2023. doi: 10.52202/075280-0380. URL https://papers.nips.cc/paper_files/paper/2023/hash/1b80fe066fdbceb3a2960117bac33917-Abstract-Conference.html.

Li Yang, Yinbin Miao, Ziteng Liu, Zhiquan Liu, Xinghua Li, Da Kuang, Hongwei Li, and Robert H. Deng. Enhanced model poisoning attack and multi-strategy defense in federated learning. *IEEE Transactions on Information Forensics and Security*, 20:3877–3892, 2025. doi: 10.1109/TIFS.2025.3555193. URL https://doi.org/10.1109/TIFS.2025.3555193.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5650–5659. PMLR, 2018. URL https://proceedings.mlr.press/v80/yin18a.html.

Hangtao Zhang, Zeming Yao, Leo Yu Zhang, Shengshan Hu, Chao Chen, Alan Liew, and Zhetao Li. Denial-of-service or fine-grained control: Towards flexible model poisoning attacks on federated learning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4567–4575, 2023. doi: 10.24963/ijcai.2023/508. URL https://www.ijcai.org/proceedings/2023/508.

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. Association for Computing Machinery, 2022. doi: 10.1145/3534678.3539231. URL https://doi.org/10.1145/3534678.3539231.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, volume 32, pp. 14774–14784. Curran Associates, Inc., 2019. URL https://papers.nips.cc/paper/2019/hash/60a6c4002cc7b29142def8871531281a-Abstract.html.

Shiyuan Zuo, Xingrun Yan, Rongfei Fan, Li Shen, Puning Zhao, Jie Xu, and Han Hu. Efficient federated learning against byzantine attacks and data heterogeneity via aggregating normalized gradients. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=Iyn0D3cFvt.

AI USE STATEMENT

Generative AI tools were used to aid and polish writing; support the retrieval and discovery of related work; assist research ideation and execution; and aid the development and checking of proofs for mathematical claims. The authors independently verified the cited literature, mathematical arguments, experimental results, and final text, and take full responsibility for all content and conclusions.

REPRODUCIBILITY STATEMENT

Algorithm 1 specifies the procedure; Appendices B.1, B.2, and C.4 give proofs and accounting assumptions. Experimental summaries specify the evaluated configurations; code is not included.

A Bandit Background and Full Procedure

A.1 Multiple-play Bandits and Thompson Sampling

Consider K arms indexed by $[K] = \{1, \dots, K\}$. Arm k generates independent Bernoulli rewards with unknown mean $\mu_k \in (0, 1)$. At round t, the learner selects a subset $\mathcal{I}(t) \subset [K]$ of fixed cardinality and observes the reward $r_k(t) \in \{0, 1\}$ of each selected arm. Maximizing cumulative reward requires learning to select arms with high means while continuing to explore arms whose means remain uncertain. In federated learning, the server is the learner and clients are the base arms; BRFL-TS additionally evaluates sampled client subsets as combinatorial arms.

For Bernoulli feedback, Thompson Sampling maintains a conjugate posterior $\mathrm{Beta}(B_k(t), F_k(t))$ for arm k, where the counters are one plus the respective positive and negative feedback counts. A fresh posterior draw $\theta_k \mid \mathcal{H}_t \sim \mathrm{Beta}(B_k(t), F_k(t))$ has conditional mean and variance

$$\mathbb{E}[\theta_k \mid \mathcal{H}_t] = \frac{B_k(t)}{B_k(t) + F_k(t)}, \qquad \mathrm{Var}[\theta_k \mid \mathcal{H}_t] = \frac{B_k(t) F_k(t)}{(B_k(t) + F_k(t))^2 (B_k(t) + F_k(t) + 1)}. \quad (13)$$

Few observations leave a Beta distribution diffuse, while many consistent observations concentrate it. Infrequent recent selection does not erase previously accumulated confidence. BRFL-TS uses group-derived feedback to update algorithmic Beta trust distributions; their Bayesian calibration for client identity is not assumed.

A.2 Complete BRFL-TS Pseudocode

Algorithm 1: BRFL-TS (single training round)

Input: K, M, Q, rollback threshold τ; trust counters $\{B_k, F_k\}_{k=1}^K$; global model $\mathcal{P}(w_t)$ and retained evaluation score a_t; privacy primitive $\mathcal{P}$
Output: $\mathcal{P}(w_{t+1})$, a_{t+1}, and updated $\{B_k, F_k\}_{k=1}^K$

Stage 1: Thompson Sampling-based client selection
for each client $k \in [K]$ **do**
  $\widetilde{p}_k \sim \mathrm{Beta}(B_k, F_k)$
end for
$p_k \leftarrow \widetilde{p}_k / \sum_{j=1}^K \widetilde{p}_j$ for every k
for $m = 1$ to M **do**
  $S_m \leftarrow \textsc{SampleWOR}([K], \{p_k\}_{k=1}^K, Q)$
end for
$\mathcal{S} \leftarrow \bigcup_{m=1}^M S_m$
Stage 2: Protected local update
Server broadcasts $\mathcal{P}(w_t)$ to clients in $\mathcal{S}$
for each client $k \in \mathcal{S}$ **in parallel do**
  $w_t^k \leftarrow \textsc{LocalTrain}(w_t, \mathcal{D}_k)$
  $g_k \leftarrow w_t - w_t^k$ and upload $\mathcal{P}(g_k)$
end for
Stage 3: Privacy-compatible candidate evaluation
for $m = 1$ to M **do**
  $\mathcal{P}(G_m) \leftarrow \frac{1}{|S_m|} \sum_{k \in S_m} \mathcal{P}(g_k)$
  $\mathcal{P}(w_t^m) \leftarrow \mathcal{P}(w_t) - \mathcal{P}(G_m)$
end for
Server broadcasts $\{\mathcal{P}(w_t^m)\}_{m=1}^M$ to all clients
for each client $k \in [K]$ **in parallel do**
  for $m = 1$ to M **do**
    $w_{t,k}^m \leftarrow \textsc{Recover}_{\mathcal{P},k}(\mathcal{P}(w_t^m))$
    $acc_{k,m} \leftarrow \textsc{Evaluate}(w_{t,k}^m, \mathcal{D}_k)$
  end for
  Submit $\{acc_{k,m}\}_{m=1}^M$
end for
$med_m \leftarrow \mathrm{Median}(\{acc_{k,m}\}_{k=1}^K)$ for every m
$m^* \leftarrow \arg\max_m med_m$ and $m^\circ \leftarrow \arg\min_m med_m$

Algorithm 1 continued

Stage 4: Adaptive reward and penalty
for each client $k \in S_{m^\circ}$ **do**
  for $r = 1$ to 3 **do**
    $z_{k,r} \sim \text{Bernoulli}(1 - \widetilde{p}_k)$
  end for
end for
$\kappa \leftarrow \left\lfloor \frac{1}{3} \sum_{k \in S_{m^\circ}} \sum_{r=1}^{3} z_{k,r} \right\rfloor$
Sort candidate indices as $j_{(1)}, \ldots, j_{(M)}$ such that $med_{j_{(1)}} \geq \cdots \geq med_{j_{(M)}}$
for each client $k \in S_{m^\circ}$ **do**
  $s_k \leftarrow \left(\mathbf{1}\{k \in S_{j_{(1)}}\}, \ldots, \mathbf{1}\{k \in S_{j_{(M)}}\}\right)$
end for
Sort the clients in S_{m° by s_k in ascending lexicographical order, using a fixed rule to break ties
$\mathcal{J} \leftarrow$ the first κ clients in this order
$\mathcal{R} \leftarrow S_{m^*} \setminus \mathcal{J}$
for $k \in \mathcal{R}$ **do**
  $B_k \leftarrow B_k + 1$
end for
for $k \in \mathcal{J}$ **do**
  $F_k \leftarrow F_k + 1$
end for
Stage 5: Reward-set aggregation and rollback
if $\mathcal{R} = \emptyset$ **then**
  return $\mathcal{P}(w_t)$, a_t, and the updated counters
end if
$\mathcal{P}(G_{\mathcal{R}}) \leftarrow \frac{1}{|\mathcal{R}|} \sum_{k \in \mathcal{R}} \mathcal{P}(g_k)$
$\mathcal{P}(\widehat{w}_{t+1}) \leftarrow \mathcal{P}(w_t) - \mathcal{P}(G_{\mathcal{R}})$
if $\mathcal{R} = S_{m^*}$ **then**
  $\widehat{a}_{t+1} \leftarrow med_{m^*}$
else
  Server broadcasts $\mathcal{P}(\widehat{w}_{t+1})$ to all clients
  for each client $k \in [K]$ **in parallel do**
    $\widehat{w}_{t+1,k} \leftarrow \text{RECOVER}_{\mathcal{P},k}(\mathcal{P}(\widehat{w}_{t+1}))$
    $\widehat{acc}_{k,t+1} \leftarrow \text{EVALUATE}(\widehat{w}_{t+1,k}, \mathcal{D}_k)$ and submit it
  end for
  $\widehat{a}_{t+1} \leftarrow \text{Median}(\{\widehat{acc}_{k,t+1}\}_{k=1}^{K})$
end if
if $\widehat{a}_{t+1} < a_t - \tau$ **then**
  $\mathcal{P}(w_{t+1}) \leftarrow \mathcal{P}(w_t)$ and $a_{t+1} \leftarrow a_t$
else
  $\mathcal{P}(w_{t+1}) \leftarrow \mathcal{P}(\widehat{w}_{t+1})$
  $a_{t+1} \leftarrow \widehat{a}_{t+1}$
end if

B Additional Proofs

B.1 Purity and the joint finite-epoch bound

This appendix gives a conditional relation between winning-group purity, benign-client coverage, and reference regret. We first define the reference-quality model and the joint event, then establish the purity relation and the finite-epoch bound.

Let $W_n = S_{n,m^*}$ denote the size-Q winning candidate group at round n, before penalty removal and rollback, and let $\mathcal{H}_n$ contain the history through that round. Let C_n denote the clients whose new updates enter the retained global model after reward-set aggregation and rollback at round n, with $C_n = \emptyset$ if no new model is retained. Then $C_n \subseteq W_n$.

Reference-quality model. Fix $\mathcal{F} \neq \emptyset$, $1 \leq Q \leq |\mathcal{G}|$, and reference qualities $\mu_k \in [0,1]$ satisfying

$$\min_{i \in \mathcal{G}} \mu_i - \max_{j \in \mathcal{F}} \mu_j \geq \Delta > 0. \tag{14}$$

These are analytical reference values, not the observed candidate scores or means of the algorithm's counter feedback. Let $\mathcal{G}_Q^\star$ be a highest-quality benign Q-subset, and define

$$r_n = \sum_{i \in \mathcal{G}_Q^\star} \mu_i - \sum_{k \in W_n} \mu_k, \qquad R(L) = \sum_{n=1}^{L} r_n. \tag{15}$$

Joint event. For integers $1 \leq H \leq N$, set $E = \lfloor N/H \rfloor$ and $N' = EH$. Partition the first N' rounds into complete epochs $\mathcal{T}_e = \{(e-1)H + 1, \ldots, eH\}$; any remaining rounds are excluded from this average. Define

$$A_e^{\text{win}} := \{W_n \subseteq \mathcal{G} \text{ for every } n \in \mathcal{T}_e\} \cap \{\mathcal{G} \subseteq \bigcup_{n \in \mathcal{T}_e} W_n\}. \tag{16}$$

Theorem 2 (Conditional joint finite-epoch bound)**.** *Under the reference-quality model above, suppose that for every $i \in \mathcal{G}$ and $1 \leq n \leq N'$, $\mathbb{P}(i \in W_n \mid \mathcal{H}_{n-1}) \geq \rho$ almost surely, for some $\rho \in (0,1]$. Then*

$$\frac{1}{E} \sum_{e=1}^{E} \mathbb{P}(A_e^{\text{win}}) \geq 1 - \frac{\mathbb{E}[R(N')]}{\Delta E} - |\mathcal{G}|(1-\rho)^H. \tag{17}$$

The two error terms account for contaminated winning groups and missed benign clients. The bound does not assume independence across rounds. Winning-group purity transfers to accepted aggregation because $C_n \subseteq W_n$; coverage additionally requires that contributions survive penalty assignment and rollback. The following deterministic relation is the purity step in Theorem 2.

Lemma 1 (Deterministic purity–regret relation)**.** *The pathwise regret and average purity satisfy*

$$r_n \geq \Delta |W_n \cap \mathcal{F}|, \qquad \frac{1}{N} \sum_{n=1}^{N} \mathbb{P}(W_n \subseteq \mathcal{G}) \geq 1 - \frac{\mathbb{E}[R(N)]}{\Delta N}. \tag{18}$$

Replace each Byzantine member of W_n by a distinct benign client outside W_n. Such replacements exist because $Q \leq |\mathcal{G}|$, and each increases the total reference quality by at least Δ. The resulting benign Q-subset has quality no larger than that of $\mathcal{G}_Q^\star$, which proves the first inequality. The bound $\mathbf{1}\{W_n \cap \mathcal{F} \neq \varnothing\} \leq r_n/\Delta$, followed by expectations and summation, proves the second.

Proof of Theorem 2. Fix a benign client i and an epoch e. Conditional on the preceding history at each round, its probability of missing the next winning group is at most $1 - \rho$. Iterating conditional expectation over the H rounds gives

$$\mathbb{P}\left(i \notin \bigcup_{n \in \mathcal{T}_e} W_n \right) \leq (1-\rho)^H. \tag{19}$$

A union bound over benign clients and contaminated rounds yields

$$\mathbb{P}((A_e^{\text{win}})^c) \leq \sum_{n \in \mathcal{T}_e} \mathbb{P}(W_n \cap \mathcal{F} \neq \varnothing) + |\mathcal{G}|(1-\rho)^H. \tag{20}$$

Sum over all E complete epochs and apply Lemma 1, giving

$$\frac{1}{E} \sum_{e=1}^{E} \mathbb{P}((A_e^{\text{win}})^c) \leq \frac{\mathbb{E}[R(N')]}{\Delta E} + |\mathcal{G}|(1-\rho)^H. \tag{21}$$

Taking complements proves Eq. (17).

B.2 SAMPLING ASSUMPTIONS AND FEEDBACK RETURN

Fix positive integers K, M, Q with $Q \leq K$, and keep all K clients eligible for sampling. Each completed round uses fresh samples from the algorithmic Beta trust distributions and, conditional on these samples, independent random draws for the M candidates. Sampling is without replacement within a candidate; repeated candidate subsets are allowed. Counters start positive and change only through the stated reward–penalty increments, with no additional reset or decay. Every client in the winning candidate receives a reward or a penalty counter update. These updates are retained when the reward set is empty or rollback rejects the model.

At round n, let $W_n = S_{n,m^*}$ denote the highest-scoring candidate group before penalty removal and rollback, and let $\mathcal{J}_n$ be the penalty set. The reward set is $W_n \setminus \mathcal{J}_n$. Let $U_{i,n}$ indicate whether client i receives a reward or penalty counter update in that round. Then

$$U_{i,n} = \mathbf{1}\{i \in W_n \cup \mathcal{J}_n\}, \tag{22}$$

and these counter updates are retained after rollback.

Consider an indefinitely continued execution and let $\mathcal{H}_t$ denote its history after round t. Write $B_i(t), F_i(t)$ for the counters after round t, and define

$$D_i(t) = \inf\{h \geq 1 : U_{i,t+h} = 1\}, \qquad \inf \varnothing = +\infty. \tag{23}$$

Finite-window statements apply to any H additional rounds actually executed; the waiting-time expectation refers to the continued process. With $c_i(t)$ defined in Eq. (11), the tail form underlying Theorem 1, for every integer $H \geq 0$, is

$$\mathbb{P}\big(D_i(t) > H \mid \mathcal{H}_t\big) \leq (1 - c_i(t))^H. \tag{24}$$

Consequently, the conditional probability that every benign client receives feedback within the next H rounds is at least

$$1 - \sum_{i \in \mathcal{G}} (1 - c_i(t))^H. \tag{25}$$

This result excludes permanent feedback starvation if the protocol can continue indefinitely. The return bound applies to every client, including Byzantine clients. For a client whose feedback history is dominated by penalties, $F_i(t)$ grows relative to $B_i(t)$, decreasing both its Beta mean $B_i(t)/(B_i(t) + F_i(t))$ and the state-dependent constant $c_i(t)$. Because the normalized sampler uses Beta draws as selection weights, such penalty-dominated histories progressively reduce the sampling pressure allocated to persistently unreliable clients, including Byzantine ones, while $c_i(t) > 0$ preserves an opportunity for future reassessment. BRFL-TS therefore suppresses exploration without imposing irreversible elimination.

Proof of Theorem 1. A client in the winning candidate belongs either to its reward set $W_n \setminus \mathcal{J}_n$ or to the penalty set $\mathcal{J}_n$. Thus it receives a counter update in either case. This statement concerns the counters, before any subsequent rollback decision.

Fix a round n and condition on its preceding history and current Beta samples. Because each sample lies in $(0, 1)$ almost surely,

$$p_i(n) = \frac{\theta_i(n)}{\sum_{k=1}^K \theta_k(n)} \geq \frac{\theta_i(n)}{K}. \tag{26}$$

Let $E_{i,n}$ be the event that client i occupies the first position of all M candidates. The first draw of each candidate has probabilities $\{p_k(n)\}_{k=1}^K$. Conditional independence of the candidate draws therefore gives

$$\mathbb{P}(E_{i,n} \mid \mathcal{H}_{n-1}, \boldsymbol{\theta}(n)) = p_i(n)^M \geq \frac{\theta_i(n)^M}{K^M}. \tag{27}$$

Sampling without replacement within each candidate does not affect this first-position event. On $E_{i,n}$, every candidate contains i; hence the winning candidate contains i under any evaluation or tie-breaking rule, and $U_{i,n} = 1$.

For a Beta random variable with parameters (B, F),

$$\mathbb{E}[\theta^M] = \frac{\mathrm{B}(B + M, F)}{\mathrm{B}(B, F)} = \prod_{\ell=0}^{M-1} \frac{B + \ell}{B + F + \ell}, \tag{28}$$

where B denotes the beta function. Integrating (27) over the current samples yields

$$\mathbb{P}(U_{i,n} = 1 \mid \mathcal{H}_{n-1}) \geq \frac{1}{K^M} \prod_{\ell=0}^{M-1} \frac{B_i(n-1)+\ell}{B_i(n-1)+F_i(n-1)+\ell}. \tag{29}$$

This bound does not impose any distributional assumption on model updates, evaluation scores, or the reward–penalty assignments beyond the stated counter-update rule.

Now condition on the history $\mathcal{H}_t$ after a fixed round t. On the event $\{D_i(t) > h\}$, client i has received no counter update in rounds $t+1, \ldots, t+h$, so

$$(B_i(t+h), F_i(t+h)) = (B_i(t), F_i(t)). \tag{30}$$

Consequently, its next-round feedback probability on this event is still at least $c_i(t)$. The tower property gives

$$\begin{aligned} &\mathbb{P}(D_i(t) > h+1 \mid \mathcal{H}_t) \\ &= \mathbb{E}[\mathbf{1}\{D_i(t) > h\}\mathbb{P}(U_{i,t+h+1} = 0 \mid \mathcal{H}_{t+h}) \mid \mathcal{H}_t] \\ &\leq (1 - c_i(t))\,\mathbb{P}(D_i(t) > h \mid \mathcal{H}_t). \end{aligned} \tag{31}$$

Induction, starting from $\mathbb{P}(D_i(t) > 0 \mid \mathcal{H}_t) = 1$, proves the tail bound in (24). Summing the conditional tail probabilities gives

$$\mathbb{E}[D_i(t) \mid \mathcal{H}_t] = \sum_{H=0}^{\infty} \mathbb{P}(D_i(t) > H \mid \mathcal{H}_t) \leq \sum_{H=0}^{\infty} (1 - c_i(t))^H = \frac{1}{c_i(t)}. \tag{32}$$

Finally, a conditional union bound over $i \in \mathcal{G}$ proves (25). No independence across clients' waiting times or across training rounds is used.

The positivity of $c_i(t)$ implies that the next feedback time is finite almost surely, conditional on any finite history, if the protocol continues to run.

A finite-horizon floor for the actual sampler. Under the sampling and counter-update assumptions of Theorem 1, with $B_i(0) = F_i(0) = 1$, each client receives at most one counter increment per round because the reward and penalty sets are disjoint. Before round $n \leq N$, $B_i(n-1) \geq 1$ and $B_i(n-1) + F_i(n-1) \leq n+1 \leq N+1$. The event that all M candidates draw i first forces $i \in W_n$, regardless of candidate ranking. Conditional independence of candidate draws and the Beta moment identity yield

$$\begin{aligned} \mathbb{P}(i \in W_n \mid \mathcal{H}_{n-1}) &\geq K^{-M}\mathbb{E}[\theta_i(n)^M \mid \mathcal{H}_{n-1}] \\ &\geq \rho_N := \frac{1}{K^M} \prod_{\ell=0}^{M-1} \frac{1+\ell}{N+1+\ell} > 0. \end{aligned} \tag{33}$$

Consequently, under the reference-quality assumptions, Theorem 2 applies with $\rho = \rho_N$ on every fixed finite horizon. For fixed K and M, $\rho_N = \Theta(N^{-M})$, so the resulting joint bound may be loose.

C Additional Evaluation and Communication

C.1 Client Participation and Trust Revision

Low overall decision-error rates can still conceal concentration of positive feedback among a small subset of benign clients. To examine this participation-related Matthew effect, we compare BRFL-TS with FedPBF (Tang et al., 2023), which uses historical experience for selection and accumulates that experience only for members of the winning group. This feedback rule can reinforce preferences for previously favored clients. Figure 3 compares accumulated BRFL-TS rewards and FedPBF winning-group memberships after 100 rounds, using $M = 10$ candidate groups of $Q = 3$ clients from 16 benign and four Byzantine clients. Under both Sign Flipping and PoisonedFL with IID and non-IID data, all 16 benign clients receive rewards from BRFL-TS, whereas each Byzantine client receives at most three. FedPBF leaves one benign client with no winning-group membership under

IID Sign Flipping and two under IID PoisonedFL. BRFL-TS therefore extends positive feedback across the benign population while keeping rewards to Byzantine clients sparse.

Positive feedback is also less concentrated among benign clients under both attacks with IID data and under non-IID Sign Flipping. For IID Sign Flipping, the benign-client Gini coefficient decreases from 0.382 with FedPBF to 0.093 with BRFL-TS. Under non-IID PoisonedFL, the coefficients are nearly equal at 0.363 and 0.362, but BRFL-TS still improves accuracy from 79.75% to 80.69%. Table 3 reports accuracy gains of 0.51 to 1.84 percentage points across the two attacks and two data partitions. Thus, broader access to positive feedback for benign clients does not come at the expense of model utility in these tests. Together with the sparse Byzantine rewards, these results support the design objective of retaining benign clients without indiscriminately favoring all clients.

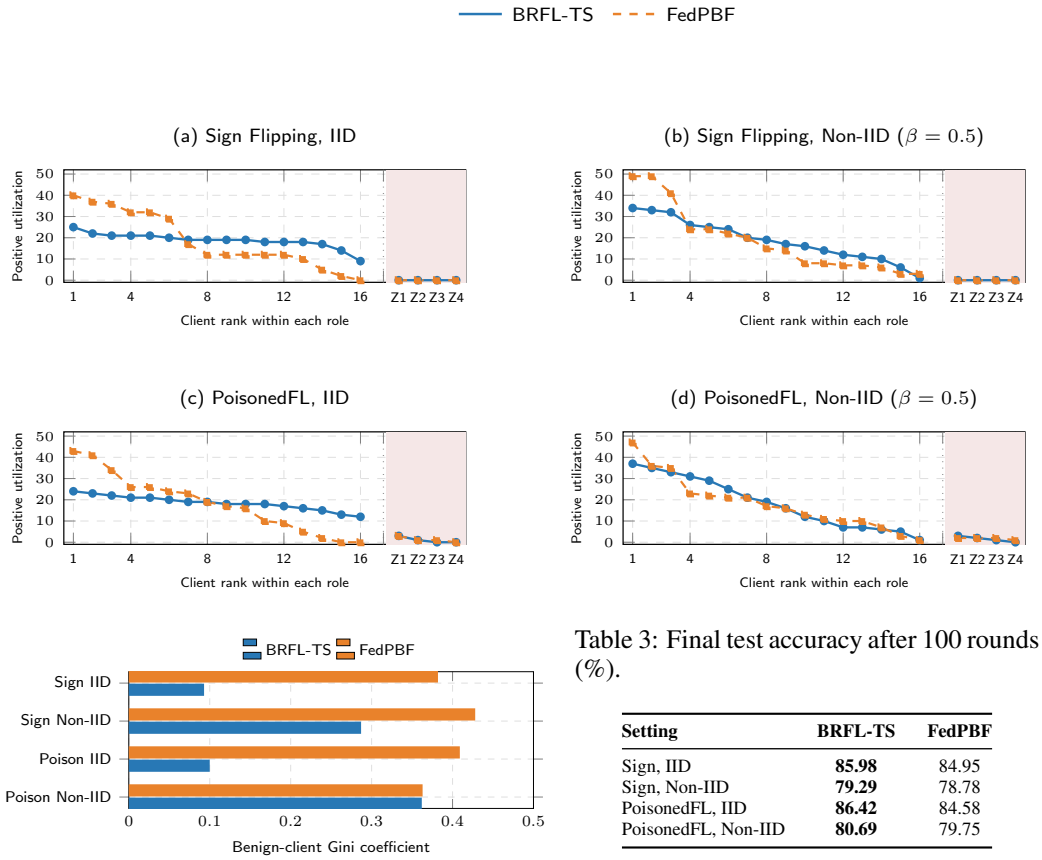


Figure 3: Client-level counts after 100 training rounds under Sign Flipping and PoisonedFL with IID and non-IID data. BRFL-TS counts pre-rollback reward events as $B_k - 1$, with the initial prior excluded, whereas FedPBF counts winning-group memberships. Panels (a)–(d) rank 16 benign and four Byzantine clients separately for each method and setting, so equal ranks need not identify the same client across methods. Panel (e) reports the Gini coefficient of benign-client counts. Final test accuracy is reported in Table 3.

Table 3: Final test accuracy after 100 rounds (%).

Setting	BRFL-TS	FedPBF
Sign, IID	**85.98**	84.95
Sign, Non-IID	**79.29**	78.78
PoisonedFL, IID	**86.42**	84.58
PoisonedFL, Non-IID	**80.69**	79.75

The paired Fang-v2 snapshots in Figure 4 show that all five benign clients penalized by round 10 receive further rewards and attain higher trust means by round 100. Among them, IID client 11 and non-IID client 6 initially have more penalties than rewards. Their Beta trust means rise from 0.333 to 0.889 and from 0.400 to 0.921, respectively. In both cases, new rewards accumulate while the penalty parameters remain unchanged. These cases show how BRFL-TS uses subsequent utility feedback to revise early negative assessments without erasing past penalties.

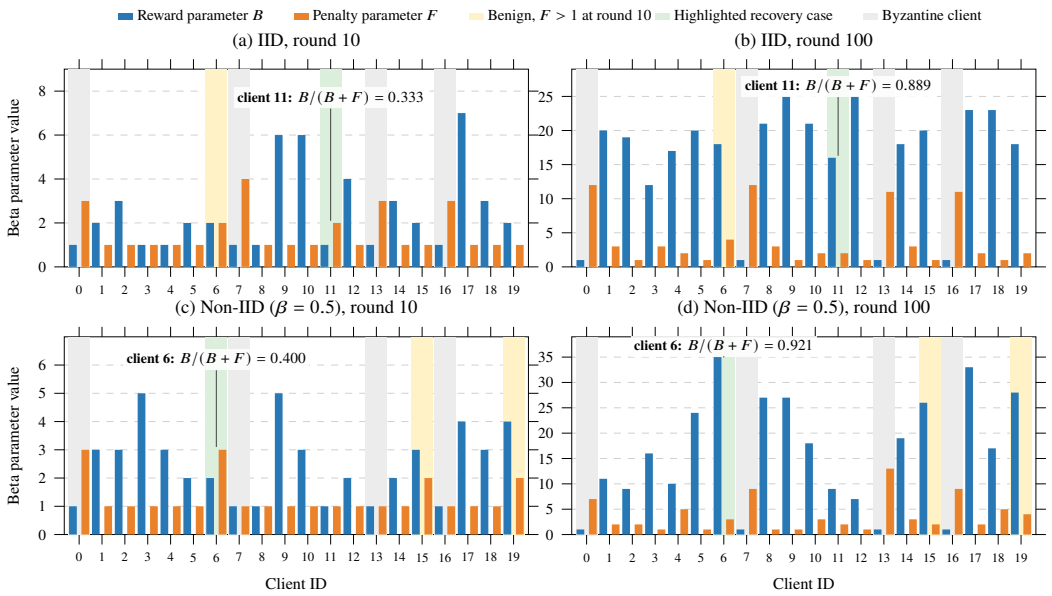


Figure 4: Paired client-level Beta trust parameters under Fang-v2 on CIFAR-10 with AlexNet and 20% Byzantine clients at rounds 10 and 100. Non-IID data use $\beta = 0.5$. The plotted B and F values include the $\mathrm{Beta}(1, 1)$ prior, and each row preserves client IDs across rounds. Gray bands mark Byzantine clients, yellow bands mark benign clients with $F > 1$ at round 10, and green bands highlight IID client 11 and non-IID client 6. The annotated $B/(B + F)$ values are internal Beta trust means.

C.2 Robustness under Byzantine Score Misreporting

Byzantine clients can attack both their model updates and their candidate-model evaluations. We therefore test whether the coordinate-wise median used in Stage 3 remains effective when Byzantine clients deliberately falsify their complete score vectors.

Setup. We use CIFAR-10 with AlexNet, $K = 20$ clients, $T = 100$ rounds, $M = 10$ candidate models, and $Q = 3$ clients per candidate. Data are IID, the model-update attack is Sign Flipping, and no additional update-protection mechanism is applied. Only Byzantine clients modify evaluation scores; benign clients report their measured local accuracies. We vary the Byzantine ratio over 10%, 20%, and 40% and consider three reporting modes: (i) honest scores; (ii) model-performance flipping, which swaps the j-th highest and j-th lowest candidate scores for $j = 1, \ldots, \lfloor M/2 \rfloor$; and (iii) independent scores drawn uniformly from $[0, 1]$.

We report final test accuracy and average candidate-purity detection accuracy, defined as the fraction of rounds in which every client in the selected optimal candidate group is benign. Tables 4–5 and Figures 5–6 reproduce the results.

Table 4: Final test accuracy under Byzantine score misreporting on CIFAR-10 (IID, Sign Flipping).

Reporting mode	10% Byzantine	20% Byzantine	40% Byzantine
No misreporting	0.864	0.860	0.856
Model-performance flipping	0.860	0.859	0.852
Random score	0.864	0.859	0.850

At 40% Byzantine clients, model-performance flipping lowers final accuracy from 85.6% to 85.2%, and random scores lower it to 85.0%; the maximum additional reduction is therefore 0.6 percentage points. Detection accuracy remains 100% at 10% and 20% Byzantine clients and 99.7% at 40%

Table 5: Average candidate-purity detection accuracy under Byzantine score misreporting on CIFAR-10 (IID, Sign Flipping).

Reporting mode	10% Byzantine	20% Byzantine	40% Byzantine
No misreporting	1.000	1.000	1.000
Model-performance flipping	1.000	1.000	0.997
Random score	1.000	1.000	0.997

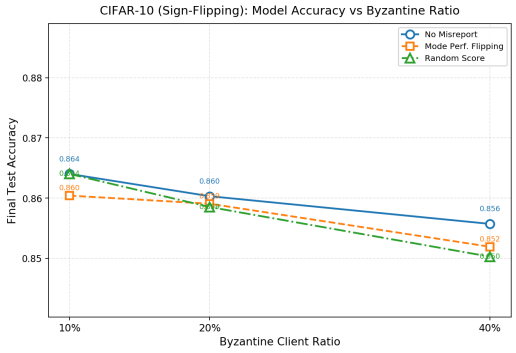


Figure 5: Final test accuracy versus Byzantine-client ratio under the three score-reporting modes.

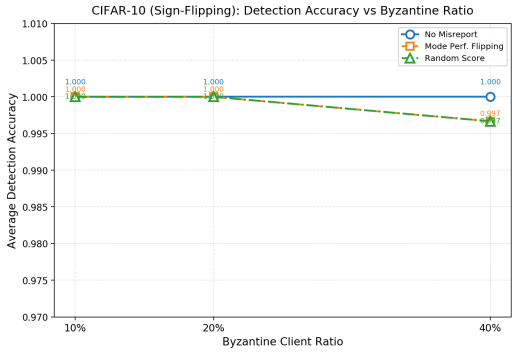


Figure 6: Average candidate-purity detection accuracy versus Byzantine-client ratio.

under both falsification modes. Thus, with an honest scoring majority, median score aggregation substantially limits the added effect of Byzantine score reports in these runs.

C.3 Training Details and Utility under Update Protection

Table 6 lists the model, local training epochs, and optimizer settings used for each dataset. For the DP experiments, client updates are clipped with $C = 5$ on MNIST and Fashion-MNIST and with $C = 2$ on Tiny-ImageNet before Gaussian perturbation. The dataset-specific Gaussian noise multipliers are listed in the corresponding utility-table captions.

Table 6: Dataset-specific model and local-training hyperparameters.

Dataset	Model	Local epochs	Optimizer
MNIST	Logistic regression	1	SGD; momentum 0.9
Fashion-MNIST	LeNet	1	SGD; momentum 0.9; weight decay 0
CIFAR-10	AlexNet	3	SGD; momentum 0.9; weight decay 0.001
Tiny-ImageNet	ViT-B/16 (224×224) `google/vit-base-patch16-224`	1	AdamW; weight decay 0.01

The following tables give the attack-averaged utility results for all update-protection settings summarized in the main text. Entries reproduce the reported mean$\pm$std across the eight attacks; a dash denotes an unavailable result. FedAvg W.O. is an attack-free reference. These are utility comparisons, not comparisons at matched end-to-end privacy budgets.

Table 7: MNIST utility under update protection. G1–G4 use σ=3.2×10^{-2}, σ=6.4×10^{-2}, σ=1.28×10^{-1}, σ=2.56×10^{-1}, respectively.

Method	Unprotected	G1	G2	G3	G4	HE	MPC
Multi-Krum	88.36 ± 1.13	85.27 ± 4.76	79.43 ± 14.68	68.38 ± 28.90	62.22 ± 27.91	–	–
FLTrust	89.35 ± 1.22	87.22 ± 3.80	80.34 ± 5.14	$\mathbf{81.95 \pm 6.39}$	$\mathbf{73.16 \pm 12.77}$	–	–
RFA	85.65 ± 2.21	78.00 ± 9.89	60.41 ± 34.16	55.83 ± 33.95	51.93 ± 32.50	–	–
Fed-NGA	84.17 ± 5.69	75.19 ± 14.42	56.64 ± 33.90	49.55 ± 37.17	44.86 ± 34.72	–	–
RandEigen	79.23 ± 5.09	78.13 ± 9.00	66.73 ± 23.91	56.95 ± 25.77	45.88 ± 25.40	–	–
BRFL-TS	$\mathbf{91.68 \pm 0.31}$	$\mathbf{88.40 \pm 0.22}$	$\mathbf{84.00 \pm 0.87}$	75.85 ± 2.24	62.50 ± 2.63	$\mathbf{90.59 \pm 2.69}$	$\mathbf{90.77 \pm 2.19}$
FedAvg W.O.	91.79 ± 0.08	91.58 ± 0.15	90.41 ± 0.35	85.76 ± 0.35	80.19 ± 1.03	91.79 ± 0.08	91.79 ± 0.08

Table 8: FMNIST utility under update protection. G1–G4 use σ=1×10^{-3}, σ=4.64×10^{-3}, σ=2.15×10^{-2}, σ=1×10^{-1}, respectively.

Method	Unprotected	G1	G2	G3	G4	HE	MPC
Multi-Krum	68.26 ± 24.24	67.32 ± 23.65	54.11 ± 23.83	26.27 ± 23.44	10.00 ± 0.00	–	–
FLTrust	77.21 ± 0.30	73.73 ± 0.51	55.19 ± 9.90	24.93 ± 22.57	$\mathbf{25.22 \pm 14.67}$	–	–
RFA	69.90 ± 24.30	62.76 ± 23.61	52.38 ± 27.18	31.59 ± 24.15	10.00 ± 0.00	–	–
Fed-NGA	70.48 ± 24.73	67.04 ± 23.93	53.25 ± 27.28	18.63 ± 15.51	10.00 ± 0.00	–	–
RandEigen	56.26 ± 31.50	59.61 ± 23.70	43.73 ± 24.48	13.04 ± 5.83	10.00 ± 0.00	–	–
BRFL-TS	$\mathbf{85.22 \pm 1.14}$	$\mathbf{85.16 \pm 0.90}$	$\mathbf{85.13 \pm 0.54}$	$\mathbf{73.38 \pm 2.23}$	16.45 ± 7.79	$\mathbf{84.93 \pm 1.30}$	$\mathbf{85.01 \pm 1.15}$
FedAvg W.O.	87.13 ± 0.16	86.87 ± 0.54	84.61 ± 0.62	68.75 ± 4.17	10.00 ± 0.01	87.10 ± 0.31	87.07 ± 0.35

Table 9: Tiny-ImageNet utility under update protection. G1–G4 use σ=5×10^{-3}, σ=7.05×10^{-3}, σ=1×10^{-2}, σ=1.415×10^{-2}, respectively.

Method	Unprotected	G1	G2	G3	G4	HE	MPC
Multi-Krum	88.70 ± 9.22	49.92 ± 37.25	42.20 ± 33.73	26.90 ± 19.57	13.43 ± 4.69	–	–
FLTrust	92.65 ± 0.99	50.80 ± 22.87	44.55 ± 24.74	40.00 ± 25.10	36.93 ± 26.99	–	–
RFA	91.53 ± 2.04	49.62 ± 37.00	44.33 ± 36.40	31.67 ± 25.78	17.83 ± 8.61	–	–
Fed-NGA	70.35 ± 26.68	16.83 ± 12.74	11.50 ± 3.70	10.05 ± 0.28	10.00 ± 0.00	–	–
RandEigen	60.05 ± 31.55	35.30 ± 31.98	28.12 ± 20.28	15.35 ± 11.13	12.73 ± 5.22	–	–
BRFL-TS	$\mathbf{92.98 \pm 0.98}$	$\mathbf{89.45 \pm 0.82}$	$\mathbf{82.97 \pm 1.21}$	$\mathbf{66.47 \pm 4.64}$	$\mathbf{51.12 \pm 8.57}$	$\mathbf{92.48 \pm 0.99}$	$\mathbf{92.60 \pm 0.46}$
FedAvg W.O.	94.20 ± 0.57	85.80 ± 1.70	73.20 ± 3.11	44.70 ± 5.52	27.20 ± 3.39	94.40 ± 0.28	94.40 ± 0.28

C.4 Communication Overhead

We count point-to-point model and score payloads in plaintext-equivalent bytes, rather than measured network traffic. Let $B = db$ be the bytes in a model or update and s the bytes in a scalar score. A baseline with n_t participating clients sends and receives $2n_tB$ bytes per round. For BRFL-TS, let $U_t = |\bigcup_m S_{m,t}| \le \min(K, MQ)$, and let $I_t = \mathbf{1}\{\emptyset \neq \mathcal{R}_t \neq S_{m^*,t}\}$ indicate reward-set reevaluation.

$$C_t^{\text{BRFL-TS}} = (2U_t + K(M + I_t))B + K(M + I_t)s. \tag{34}$$

The terms account for selected-client training, evaluation of M candidates by all clients, and the conditional evaluation of the revised reward-set model. An empty reward set returns the previous model without reevaluation. With $\bar{I} = T^{-1}\sum_t I_t$, the ratio to a full-participation baseline is

$$\Gamma = \frac{1}{TK}\sum_t U_t + \frac{M + \bar{I}}{2}\left(1 + \frac{s}{B}\right) \le 1 + \frac{M+1}{2}\left(1 + \frac{s}{B}\right). \tag{35}$$

For $M = 10$ and negligible score payload, this envelope is approximately $6.5\times$. BRFL-TS uses $2T + \sum_t I_t$ sequential request-response phases under this accounting, compared with T for a baseline that only uploads and downloads once per round. The exact payload depends on the realized union sizes, reevaluations, and baseline participation counts; the expression is an analytical envelope rather than a measured network cost.

For fixed M, the plaintext-equivalent payload scales as $O(TKd)$. This is not an upper bound on encrypted or masked protocol traffic: ciphertext expansion, key distribution, masking, secure reconstruction, and possible additional secure aggregation for the reward set must be counted separately. Candidate evaluation also requires KM local model evaluations per round, plus KI_t conditional reevaluations. These computation and cryptographic costs are not measured by the payload formula.