FedTAP: Trust-aware Temporal Aggregation for Robust Federated Learning

NODI Lab., Jilin University

Abstract

Federated learning (FL) enables multiple clients to collaboratively train a shared global model without exposing private data, but its distributed nature makes it vulnerable to adversarial and unreliable updates. Existing robust aggregation methods focus on single-round defenses and often fail against adaptive or stealthy attacks that evolve gradually over time. In this work, we propose **FedTAP** (*Federated Trust-aware Adaptive Prediction and Aggregation*), a temporal modeling framework that enhances robustness by explicitly incorporating the temporal dynamics of global model updates. FedTAP treats the server as an observer that predicts the expected benign evolution of the global model and measures deviations as temporal residuals. These residuals are combined with spatial consistency cues to compute a unified anomaly score, which drives a temporal trust propagation mechanism that accumulates each client's credibility across rounds. The resulting trust values are then used to assign adaptive aggregation weights, allowing reliable clients to maintain influence while gradually suppressing persistent adversaries.

1 Introduction

Federated learning (FL) has become an essential paradigm for distributed machine learning, where a central server coordinates multiple clients to train a shared global model without directly accessing their local data (McMahan et al., 2017). This design protects data privacy and enables large-scale learning across heterogeneous devices and organizations. However, the distributed nature of FL also introduces security and reliability challenges. Some clients may behave maliciously or produce unreliable updates due to adversarial intent, hardware failure, or data corruption. These abnormal updates can significantly degrade the global model, cause convergence instability, or even implant hidden backdoors (Bagdasaryan et al., 2020; Wang et al., 2020). Since the server can only observe model updates and has no visibility into the local data, detecting such behaviors remains a difficult and open problem.

To mitigate these vulnerabilities, many robust aggregation algorithms have been proposed. Early works such as Krum (Blanchard et al., 2017) and the coordinate-wise median or trimmed mean estimators (Yin et al., 2018) reduce the influence of corrupted updates by selecting or reweighting client gradients based on spatial consistency. Later, the geometric-median aggregation (RFA) (Pillutla et al., 2022) improves stability under heterogeneous data distributions, and divergence-based methods such as the γ-mean estimator (Li et al., 2022) achieve robustness against heavy-tailed or adversarial updates. More recent studies employ Huber-loss based aggregation to balance robustness and convergence under non-IID data (Zhao et al., 2024). Although these methods achieve strong performance against Byzantine attacks, they operate at the level of individual communication rounds and do not consider temporal information across rounds. As a result, they remain vulnerable to adaptive and temporally coordinated attacks that make small but consistent changes over time.

Several studies have investigated poisoning and backdoor attacks that exploit this temporal weakness. Attackers can gradually modify model parameters in multiple rounds while keeping each individual update close to the normal distribution (Wang et al., 2020). This strategy allows the malicious clients to remain undetected while progressively biasing the global model. To defend against such attacks, approaches such as FLTrust (Cao et al., 2020) use a small trusted dataset at the server for calibration, and clustering-based detection methods such as MUDGuard (Wang et al., 2024) identify anomalies through spatial analysis of client gradients. However, these defenses treat each training round independently and cannot capture long-term behavioral patterns. Recent methods attempt to use historical information, such as FoolsGold, which detects sybil attacks by analyzing the similarity of client updates over time (Fung et al., 2018), and recursive aggregation methods that adjust client contributions based on previous model deviations (Herath et al., 2023). Yet, these approaches still rely on short-term memory and lack an explicit model for the temporal evolution of trust or the prediction of the global trajectory.

In this work, we propose **FedTAP** (***Fed****erated* ***T****rust-aware* ***A****daptive* ***P****rediction and Aggregation*), a temporal modeling framework for robust federated learning. We view the server as an *observer* that continuously monitors the global model dynamics. FedTAP first builds a lightweight predictor that estimates the expected benign evolution of the global model. Deviations from this forecast define temporal residuals that measure inconsistency between the observed and predicted behaviors. To complement this, a spatial subspace consistency check captures deviations in the geometry of client updates. The two signals are combined into a unified multi-scale anomaly score that drives a *temporal trust propagation* mechanism. This mechanism accumulates each client's credibility over time, allowing the system to recognize persistent anomalies while avoiding overreaction to temporary noise or statistical fluctuations. Finally, the accumulated trust scores are mapped to adaptive weights for the aggregation step, which enables smooth and interpretable adjustment of each client's influence.

FedTAP advances robust federated learning by integrating temporal prediction, trust propagation, and adaptive aggregation within a single framework. It has three main advantages. First, temporal prediction captures long-term consistency of the global model and detects slow, stealthy attacks. Second, trust propagation maintains stability and prevents drastic changes caused by transient noise. Third, the soft weighting scheme ensures compatibility with existing aggregation rules without modifying the standard communication process. Experimental results on benchmark datasets such as MNIST, CIFAR-10, and Tiny-ImageNet show that FedTAP significantly improves resilience against both fast and slow poisoning attacks while maintaining high model accuracy and stable convergence.

2 RELATED WORK

Federated learning allows multiple clients to train a shared model collaboratively without exchanging their private data (McMahan et al., 2017). This decentralized paradigm protects data privacy and improves scalability but is highly sensitive to malicious or unreliable clients. Robust aggregation methods aim to mitigate this problem by designing aggregation schemes that can tolerate a certain proportion of corrupted updates. Krum selects the update that is most consistent with others in Euclidean distance, offering formal Byzantine resilience guarantees (Blanchard et al., 2017). Coordinate-wise median and trimmed mean estimators further improve robustness by achieving optimal statistical error bounds under bounded adversarial perturbations (Yin et al., 2018). Pillutla et al. introduce the geometric-median based RFA method, which shows strong empirical robustness across non-IID datasets (Pillutla et al., 2022). Subsequent research explores the γ-mean aggregation rule, which uses minimum γ-divergence estimation to enhance stability under heavy-tailed and adversarial noise (Li et al., 2022). More recent work adopts Huber-loss based aggregation to balance robustness and accuracy under heterogeneous data conditions (Zhao et al., 2024). Although these algorithms perform well for single-round aggregation, they generally overlook the temporal evolution of model parameters across rounds, which is crucial for detecting slow or adaptive attacks.

Beyond aggregation design, many studies investigate poisoning and backdoor attacks in FL. Bagdasaryan et al. demonstrate that a small number of adversarial clients can perform model replacement to embed backdoors while maintaining high accuracy on clean data (Bagdasaryan et al., 2020). Wang et al. show that coordinated multi-round manipulations can produce stealthy and persistent backdoor effects (Wang et al., 2020). Surveys by Nguyen et al. (Nguyen et al., 2024) and Manzoor et al. (Manzoor et al., 2024) summarize diverse poisoning and backdoor strategies as well as their countermeasures. On the defensive side, FLTrust leverages a small trusted dataset at the server to calibrate client updates and mitigate poisoning risk (Cao et al., 2020). Clustering-based anomaly detection methods, such as MUDGuard, identify malicious clients by analyzing the spatial similarity of gradient directions (Wang et al., 2024). These approaches, however, focus on static aggregation within each round and do not explicitly model how malicious behaviors evolve temporally.

Several recent studies begin to integrate temporal or reputational information into the defense process. FoolsGold tracks the similarity of client updates across rounds to detect sybil-based poisoning attacks (Fung et al., 2018). Herath et al. propose recursive Euclidean distance based aggregation that adjusts the contribution of each client using their historical deviation from previous global models (Herath et al., 2023). Although these methods utilize history, they still rely on local pairwise consistency or short-term correlations rather than forecasting the global model trajectory or formalizing long-term trust. Building on these observations, we propose **FedTAP**, which explicitly treats the server as an observer of a dynamic system. It forecasts the expected benign evolution of the global model and accumulates temporal trust based on multi-round consistency.

3 Proposed Design of FedTAP

Federated learning aggregates client updates to evolve a global parameter $g_t \in \mathbb{R}^d$:

$$g_{t+1} \;=\; \mathcal{F}_{\text{base}}\big(g_t, \{\Delta_{t,i}\}_{i\in S_t}\big), \tag{1}$$

where $S_t \subseteq [N]$ is the set of participating clients and $\mathcal{F}_{\text{base}}$ is an arbitrary base aggregator (FedAvg, Trimmed-Mean, Bulyan, *etc.*). We consider *temporally stealthy* adversaries that control a subset $\mathcal{M}$ with $|\mathcal{M}|/N < 0.5$. These adversaries inject small but coordinated changes in each round, which gradually alter the global model while keeping single-round statistics almost unchanged.

We view the server as an *observer* of a dynamical system and monitor its *temporal consistency*. The proposed method has two main parts: (i) a light-weight predictor that estimates the expected benign evolution of the global model, and (ii) a *temporal trust propagation* mechanism that aggregates evidence across rounds. This design allows the server to separate persistent adversaries from normal clients and prevents overreaction to random noise or single-round outliers.

3.1 Observer-based Prediction and Temporal Residuals

In each communication round t, the server observes the current global model g_t and a batch of client updates $\{\Delta_{t,i}\}_{i\in S_t}$. Under normal conditions, the global parameters change smoothly because both optimization noise and data heterogeneity vary gradually. We use this temporal smoothness to predict where the global model is expected to move next. A client update that moves the model away from this predicted direction is likely to be abnormal or malicious.

Modeling benign dynamics. We model the normal evolution of the global model as

$$g_{t+1} = F(g_t, \{\Delta_{t,i}\}_{i\in S_t}) + \xi_t, \tag{2}$$

where $F(\cdot)$ represents the aggregation mapping and ξ_t accounts for randomness caused by client sampling or mini-batch optimization. Because F is unknown and may be nonlinear, the server learns a data-driven approximation of its dynamics. We use a **linear autoregressive (AR)** predictor:

$$\hat{g}_{t+1} = A_0 g_t + A_1 g_{t-1} + \cdots + A_{K-1} g_{t-K+1}, \tag{3}$$

where each A_k is a small coefficient matrix that captures short-term relationships between previous and future global parameters. The predictor learns how the model tends to evolve during the recent W rounds. To estimate $\{A_k\}$, we minimize the mean squared prediction error:

$$\min_{\{A_k\}} \sum_{u=t-W+1}^{t} \Big\| g_{u+1} - \sum_{k=0}^{K-1} A_k g_{u-k} \Big\|_2^2 + \rho \sum_k \|A_k\|_F^2, \tag{4}$$

where the small ridge term ρ (typically 10^{-4}) improves stability and prevents overfitting. The predictor can be updated each round using **recursive least squares (RLS)** with a forgetting factor of 0.95, which requires only $O(W d_{\text{eff}})$ operations per layer. Empirical results show that $K \in [2, 4]$ and $W \in [10, 30]$ are sufficient to capture the evolution of federated learning dynamics.

Candidate global and residual computation. Once the server obtains the prediction $\hat{g}_{t+1}$, it evaluates each client's update separately. For client i, the server imagines what the global model would become if only this client's update were applied:

$$g_{t+1}^{(i)} = g_t + \eta \Delta_{t,i}, \tag{5}$$

where η is the server step size (or 1 if $\Delta_{t,i}$ already includes the step). The virtual model $g_{t+1}^{(i)}$ represents the trajectory that client i alone would produce. The difference between the predicted benign model $\hat{g}_{t+1}$ and the client-induced model $g_{t+1}^{(i)}$ defines the *temporal residual*:

$$r_{t,i} = \|g_{t+1}^{(i)} - \hat{g}_{t+1}\|_2. \tag{6}$$

A large residual indicates that the client update would move the model in a direction that is inconsistent with recent global behavior. To ensure balance across network layers, $r_{t,i}$ is computed for each layer and then normalized before summation.

Takeaways 3.1. *The predictor can be updated online without storing the full training history. The server maintains a rolling buffer of W past global models and solves equation 4 using standard RLS updates. In practice, this requires only a few matrix–vector multiplications per round. The matrices A_k can also be restricted to block-diagonal form so that each network layer evolves independently, which reduces computation and improves stability.*

3.2 Multi-scale Detection and Temporal Trust

Single-round detection often fails when attackers behave slowly or adaptively. To address this problem, we combine two complementary consistency checks: (i) a *spatial cue* that measures how far a client update lies from the main subspace spanned by recent benign updates, and (ii) a *temporal cue* that tracks how each client's behavior evolves across rounds.

Subspace consistency (spatial cue). During normal training, most client updates lie within a low-dimensional subspace because of the shared model architecture and similar data distributions. Let $\{\bar{\Delta}_u\}_{u=t-W+1}^{t}$ denote the aggregated updates from the most recent W rounds. We maintain an incremental Principal Component Analysis (PCA) to estimate this dominant subspace $\mathcal{S}$. The subspace dimension d_s is selected to retain 90–95% of the total variance. The deviation of client i's update from this subspace is measured as

$$s_{t,i} = \left\|\Delta_{t,i} - \mathbf{P}_{\mathcal{S}}(\Delta_{t,i})\right\|_2, \tag{7}$$

where $\mathbf{P}_{\mathcal{S}}(\cdot)$ is the projection onto $\mathcal{S}$. A large $s_{t,i}$ indicates that the client update moves in a direction that is not well aligned with recent benign updates, which may suggest manipulation. The PCA basis is updated online each round and requires $O(d_s^2 d_{\text{eff}})$ operations.

Because the scale of residuals can vary across rounds, we apply robust normalization using the median and the median absolute deviation (MAD), which are resistant to outliers:

$$\tilde{r}_{t,i} = \frac{r_{t,i} - \mathrm{Med}(r_t)}{1.4826\,\mathrm{MAD}(r_t)}, \quad \tilde{s}_{t,i} = \frac{s_{t,i} - \mathrm{Med}(s_t)}{1.4826\,\mathrm{MAD}(s_t)}. \tag{8}$$

Here $r_{t,i}$ is the temporal residual defined in Eq. equation 6. After normalization, typical benign clients have values near 0–1, while clear outliers often exceed 3–4. We combine both cues into a single *anomaly score*:

$$z_{t,i} = \alpha\tilde{r}_{t,i} + \beta\tilde{s}_{t,i}, \quad \text{where } \alpha, \beta \geq 0, \text{ and } \alpha{=}\beta{=}1 \text{ by default.} \tag{9}$$

This score balances spatial and temporal information for reliable detection. We convert this anomaly score into an *instantaneous credibility* $c_{t,i} \in (0,1)$ through a smooth logistic mapping:

$$c_{t,i} = \frac{1}{1 + \exp((z_{t,i} - \tau_t)/\kappa_t)}, \tag{10}$$

where τ_t is an adaptive threshold and κ_t controls the sharpness of the transition. A value of $c_{t,i}$ close to 1 means the update appears normal, while smaller values indicate that the update is less reliable. We set $\kappa_t = 1.4826\,\mathrm{MAD}(\{z_{t,i}\})$ so that the mapping automatically adapts to the score dispersion in each round.

Temporal trust propagation (temporal cue). Instantaneous credibility alone can be unreliable because a benign client may appear abnormal in a single noisy round. To improve long-term robustness, we maintain a latent *trust variable* $\theta_{t,i} \in [0,1]$ for each client. This variable integrates the client's credibility over time:

$$\theta_{t+1,i} = \beta_\theta\,\theta_{t,i} + (1 - \beta_\theta)\,c_{t,i}, \tag{11}$$

where $\beta_\theta \in [0.8, 0.99]$ determines the memory length (default β_θ=0.9). If a client remains reliable ($c_{t,i} \approx 1$), its trust gradually returns to 1. If it stays suspicious for several rounds, $\theta_{t,i}$ decays exponentially and marks the client as low trust. This process smooths short-term noise and captures persistent misbehavior. We also apply a mild hysteresis rule: when $\theta_{t,i}$ is below 0.3, small one-time improvements do not immediately restore full trust.

Client behaviors and gradient magnitudes may change over time. Therefore, the decision threshold τ_t should adjust smoothly. We update it using robust exponential smoothing:

$$\tau_{t+1} = \gamma_\tau \, \tau_t + (1 - \gamma_\tau) \, \mathrm{Med}(\{z_{t,i}\}) + c_\tau \, \mathrm{MAD}(\{z_{t,i}\}), \tag{12}$$

where $\gamma_\tau \in [0.8, 0.95]$ defines the memory length and $c_\tau \in [2, 4]$ controls sensitivity. This update allows the threshold to follow gradual distribution shifts while ignoring short-term noise.

Takeaways 3.2. *Each round provides an instantaneous credibility $c_{t,i}$ based on current statistics, and a trust variable $\theta_{t,i}$ that aggregates credibility over time. Benign clients maintain high trust, while adversarial clients accumulate low trust and are gradually downweighted in the aggregation. Together, this multi-scale mechanism identifies both short-term anomalies and slow, stealthy attacks.*

3.3 Trust-aware Adaptive Aggregation and Updating

Once the server obtains the latent trust scores $\{\theta_{t,i}\}$, it must incorporate them into the aggregation process in a smooth and interpretable way. Instead of using hard rejection or strict thresholds, we translate trust into *soft aggregation weights*. This approach allows reliable clients to have stronger influence, while clients that behave abnormally are gradually suppressed. The design also provides temporal memory, so that the model does not overreact to noise in a single round.

Trust-guided aggregation. We convert each client's trust value $\theta_{t,i} \in [0, 1]$ into a positive aggregation weight:

$$w_{t,i} = \exp\big(-\lambda(1 - \theta_{t,i})\big), \tag{13}$$

where $\lambda > 0$ controls how strongly low-trust clients are downweighted (typical $\lambda \in [4, 6]$). When $\theta_{t,i} = 1$, the client is fully trusted ($w_{t,i} = 1$); when $\theta_{t,i} = 0.5$, the weight decreases to $e^{-\lambda/2}$. Since $\theta_{t,i}$ evolves smoothly according to Eq. equation 11, this mapping makes the aggregation stable across rounds. A client with a temporary anomaly will not be severely penalized, but persistent abnormal behavior will lead to exponential suppression.

The adaptive weights are then integrated into the base aggregator:

$$g_{t+1} = \mathcal{F}_{\text{base}}\Big(g_t, \{w_{t,i} \cdot \Delta_{t,i}\}_{i \in S_t}\Big), \tag{14}$$

where $\mathcal{F}_{\text{base}}$ can represent FedAvg, Trimmed-Mean, Bulyan, or other robust methods. This structure is plug-and-play: the proposed trust mechanism acts as a *reweighting layer* that can be added to any existing aggregation framework. For long-term stability, a client is temporarily isolated if its trust remains below a minimum level $\theta_{\min}$ for L consecutive rounds. We set $\theta_{\min} = 0.3$ and $L = 3$ by default. Such clients are re-evaluated after a cooldown period to avoid permanent exclusion.

Stability and analysis. To analyze the mechanism, assume benign residuals follow a sub-Gaussian distribution with variance proxy σ^2, and that an adversary causes a mean shift $\delta > 0$ in the anomaly score $z_{t,i}$ for L consecutive rounds. Then the expected credibility decreases by $\Omega(\delta/\kappa_t)$. By repeatedly applying Eq. equation 11, the trust variable evolves as

$$\theta_{t+L,i} \le \beta_\theta^L \theta_{t,i} + (1 - \beta_\theta^L)\Big(1 - \Omega(\tfrac{\delta}{\kappa_t})\Big), \tag{15}$$

which shows that persistent anomalies cause an exponential decay in trust. Under the standard Lipschitz continuity of $\mathcal{F}_{\text{base}}$, the expected model deviation satisfies

$$\mathbb{E}[\|g_{t+1} - g^*\|] \le \rho \, \mathbb{E}[\|g_t - g^*\|] + \epsilon, \tag{16}$$

where $\rho < 1$ because adversarial updates receive very small weights as $\theta_{t,i}$ decreases. This guarantees the stability and convergence of the global model even when bounded adversarial perturbations are present.

Takeaways 3.3. *This stage transforms trust into quantitative influence. Clients with consistent behavior keep high trust and near-unit weights, while long-term or stealthy attackers experience exponential trust decay, which limits their effect on the global model. This time-weighted aggregation extends standard robust aggregators with a temporal perspective, combining statistical robustness with adaptive control.*

REFERENCES

Eugene Bagdasaryan, Alexander Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 2938–2948, 2020.

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Byzantine-tolerant machine learning. *arXiv preprint arXiv:1703.02757*, 2017.

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)*, 2020.

Clement Fung, Chih-Jen Yoon, and Ivan Beschastnikh. Mitigating sybils in federated learning poisoning. In *arXiv preprint arXiv:1808.04866*, 2018.

Charuka Herath, Yogachandran Rahulamathavan, and Xiaolan Liu. Recursive euclidean distance based robust aggregation technique for federated learning. In *Proceedings of 2023 IEEE IAS Global Conference on Emerging Technologies (GlobConET)*, pp. 1–6, 2023.

Cen-Jhih Li, Pin-Han Huang, Yi-Ting Ma, Hung Hung, and Su-Yun Huang. Robust aggregation for federated learning by minimum γ-divergence estimation. *Entropy*, pp. 686, 2022.

Habib Ullah Manzoor, Attia Shabbir, Ao Chen, David Flynn, and Ahmed Zoha. A survey of security strategies in federated learning: Defending models, data, and privacy. *Future Internet*, 16:374, 2024.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1273–1282, 2017.

Thuy Dung Nguyen, Tuan Nguyen, Phi Le Nguyen, Hieu H. Pham, Khoa D. Doan, and Kok-Seng Wong. Backdoor attacks and defenses in federated learning: A survey, challenges and future directions. *Engineering Applications of Artificial Intelligence*, 127:107166, 2024.

Krishna Pillutla, Sham M. Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022.

Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pp. 16070–16084, 2020.

Rui Wang, Xingkai Wang, Huanhuan Chen, Jérémie Decouchant, Stjepan Picek, Nikolaos Laoutaris, and Kaitai Liang. Mudguard: Taming malicious majorities in federated learning using privacy-preserving byzantine-robust clustering. *Proceedings of the ACM on Measurement and Analysis of Computing Systems (POMACS)*, pp. 1–41, 2024.

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 (ICML)*, volume 80, pp. 5650–5659, 2018.

Puning Zhao, Fei Yu, and Zhiguo Wan. A huber loss minimization approach to byzantine robust federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 21806–21814, 2024.