FEDHRA: CLIENT-SIDE PURIFICATION AND HIERARCHICAL ROBUST AGGREGATION FOR FEDERATED BACKDOOR DEFENSE IN TIME-SERIES CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning enables industrial sites to train time-series classification models without sharing raw process data, but it exposes training to backdoor poisoning from compromised clients. Existing robust aggregation defenses, designed for generic cross-device settings, become unreliable under the strong non-IID distributions of industrial cross-silo FL, where benign updates naturally deviate from the global average. We propose Federated Backdoor Defense with Client-Side Purification and Hierarchical Robust Aggregation (FedHRA). On the client side, a conditional variational autoencoder learns the class-conditional normal manifold of local time-series signals; prediction flipping, confidence drop, and reconstruction error are fused into a suspiciousness score that purifies risky samples and provides client-level risk evidence. On the server side, FedHRA separates the backbone from the classification head and, using client-reported risk signals and temporal memory, applies stronger sanctions to the decision-sensitive head while softly downweighting the backbone. Experiments on the Tennessee Eastman Process and Spoken Arabic Digits datasets show that FedHRA suppresses terminal attack success rates to near zero under single-shot, continuous, periodic, and late-stage attacks while maintaining usable main-task accuracy.

1 INTRODUCTION

The development of Industry 4.0 is driving intelligent manufacturing from isolated local intelligence toward cross-site collaborative intelligence (Pozzi et al., 2023; Wang et al., 2021; Leng et al., 2025). In high-end manufacturing, cross-factory quality inspection, and large-scale predictive maintenance, deep time-series models have become important tools for extracting task-relevant features from industrial signals (Zhou et al., 2021; Sampath et al., 2023; Roth et al., 2022; Zhang et al., 2021; Vahabi et al., 2025). However, data collected by different factories and production lines are usually isolated because of commercial confidentiality, data governance, and industrial-network constraints (Xiong et al., 2021; Bonawitz et al., 2017; Yan et al., 2024). Federated learning (FL) is a practical collaborative modeling approach that allows industrial clients to train a global model without exchanging raw data (McMahan et al., 2017; Kairouz et al., 2021; Huang et al., 2022; Ogier du Terrail et al., 2022). Although FL mitigates the conflict between data sharing and privacy protection (Liu et al., 2024), it also reshapes the security boundary of model training (Bagdasaryan et al., 2020). Local training is no longer fully observable by the server, and model updates may themselves become attack carriers.

Backdoor attacks are a particularly severe threat to industrial FL because they can implant hidden trigger-response behavior while preserving apparently normal clean accuracy (Gu et al., 2017; Xie et al., 2020). In cross-silo FL, the server usually receives only model updates and cannot directly audit local data or training details (Liu et al., 2022; Kotelevskii et al., 2022). A compromised factory node or edge gateway can therefore poison local samples, modify local objectives, or bias model updates to inject a target trigger into the global time-series classification model (Xie et al., 2020; Nguyen et al., 2022; Rieger et al., 2022). Once the global model is deployed in visual inspection, anomaly detection, or predictive maintenance, such a hidden behavior may cause defect leakage, missed alarms, or unsafe operation decisions (Xue et al., 2022; Li et al., 2021b).

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.

Existing defenses against federated backdoors mainly focus on server-side robust aggregation and malicious-update detection (Pillutla et al., 2022; Cao et al., 2020; Nguyen et al., 2022; Rieger et al., 2022; Lu et al., 2022; Fung et al., 2018). Representative methods include Krum and Multi-Krum (Blanchard et al., 2017), coordinate-wise Median and Trimmed Mean (Yin et al., 2018), similarity-based suppression (Fung et al., 2018), clustering-based filtering (Nguyen et al., 2022), and model-inspection-based detection (Rieger et al., 2022). However, these methods are difficult to apply directly to industrial cross-silo FL. First, industrial data are strongly non-IID because factories differ in equipment types, production batches, working conditions, and class distributions (Vahabi et al., 2025; Leng et al., 2025; Li et al., 2020; Wang et al., 2020; Karimireddy et al., 2020b). Benign updates can naturally deviate from the global average, so distance-based or majority-consistency defenses may mistakenly suppress useful long-tail clients (Blanchard et al., 2017; Yin et al., 2018; Wang et al., 2020; Karimireddy et al., 2020b; Li et al., 2021a). Once benign clients are rejected, the relative influence of malicious clients may even increase (Ye et al., 2025). Second, time-series classification models are usually structurally layered. Backdoor attacks often manipulate the decision-related parameter subspace with small, directional changes; such changes are not necessarily obvious outliers in the full update vector (Arivazhagan et al., 2019; Matsuda et al., 2022; Ma et al., 2022). Third, privacy constraints prevent the server from accessing sample-level evidence, making server-only filtering passive and uncertain under low poisoning ratios.

To address these challenges, we propose **Fed**erated Backdoor Defense with Client-Side Purification and **H**ierarchical **R**obust **A**ggregation (FedHRA). FedHRA uses the computation available at industrial edge clients and combines data-source purification with model-level robust aggregation. On the client side, a conditional variational autoencoder (CVAE) learns the class-conditional normal manifold of local time-series data. Suspicious samples are identified using prediction flipping, confidence drop, and reconstruction error, then purified before local training. The client also uploads a low-dimensional suspicious-sample ratio as client-level risk evidence. On the server side, FedHRA treats the backbone and classification head differently. The classification head is more sensitive to backdoor-induced decision-boundary shifts, so it receives stronger sanctions. The backbone mainly learns reusable representations, so it is softly downweighted to avoid excessive damage to benign heterogeneous knowledge.

The main contributions are summarized as follows:

- We propose a federated backdoor defense framework for time-series classification that extends defense from passive server-side aggregation to the full path of data purification, local training, and global aggregation.
- We design a CVAE-based client-side purification mechanism that converts sample-level reconstruction and prediction instability into client-level risk evidence, weakening poisoned samples before they enter local model updates.
- We propose a hierarchical robust aggregation method that separates representation and classification parameters and, via robust normalization, temporal memory, and layer-wise sanctions, suppresses backdoor updates while retaining useful non-IID contributions.

2 MOTIVATION

In industrial cross-silo FL, statistical heterogeneity makes benign updates resemble malicious ones. As illustrated in Fig. 1(a), a defense boundary that rejects updates far from the global average may discard useful information from clients with distinct operating conditions or class distributions, whereas a more permissive boundary may inadvertently admit malicious updates. Server-side aggregation also lacks direct access to local samples, so it can only react after poisoning has already shaped local updates. These two difficulties point to different remedies: sample-level purification at the data source addresses the observability problem, while update-level defense must distinguish heterogeneous but benign clients from malicious ones.

FedHRA accordingly deploys client-side purification. Because industrial cross-silo clients have richer computation and storage than lightweight mobile devices, each client can afford to train a local CVAE, reconstruct trigger-corrupted samples before local training, and report a compact suspicious-sample ratio as client-level risk evidence. On the server side, FedHRA separates the backbone from the classification head. The head directly controls class decision boundaries and is more sensitive

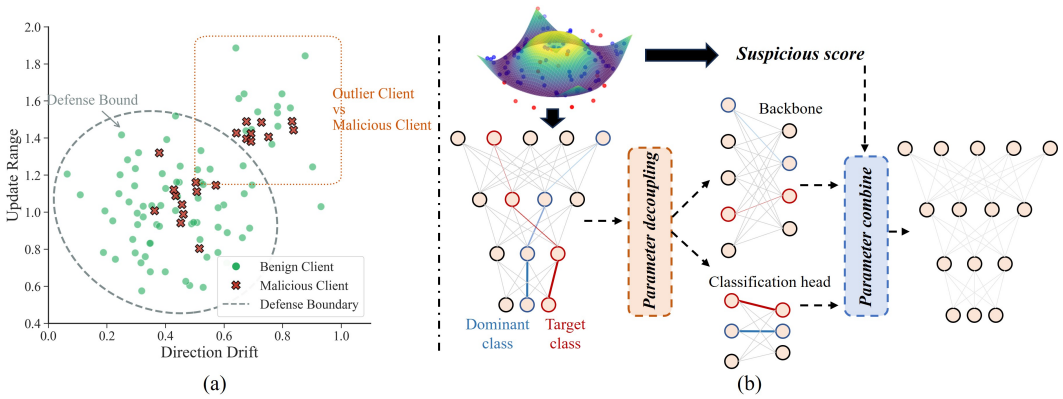


Figure 1: Motivation of this study: (a) challenge of federated backdoor defense under statistical heterogeneity; (b) schematic diagram of the proposed hierarchical federated backdoor defense.

to target-class manipulation, so it receives stronger sanctions; the backbone mainly learns reusable representations, so it is softly downweighted to preserve useful heterogeneous knowledge (Fig. 1(b)). This separation suppresses backdoor influence without discarding the knowledge of long-tail non-IID clients.

3 FedHRA Method

Fig. 2 shows the FedHRA framework. It consists of client-side data purification and server-side hierarchical robust aggregation.

3.1 Problem Definition and Notation

Consider an industrial FL system with K clients, denoted by $\mathcal{U} = \{1, 2, \ldots, K\}$. Training proceeds for R communication rounds. The server maintains a global model $\boldsymbol{\theta}_g^{(t)}$ at round t, and client k returns local parameters $\boldsymbol{\theta}_k^{(t)}$. The model is divided into backbone parameters and classification-head parameters:

$$\boldsymbol{\theta}_k^{(t)} = \{\boldsymbol{\theta}_k^{b,(t)}, \boldsymbol{\theta}_k^{h,(t)}\}. \tag{1}$$

The corresponding parameter index sets are $\mathcal{P}_b$ and $\mathcal{P}_h$.

Client k has local distribution $\mathcal{D}_k$. A clean time-series sample is $(\mathbf{x}, y) \sim \mathcal{D}_k$, where $\mathbf{x} \in \mathbb{R}^{T \times D}$ and $y \in \{0, 1, \ldots, C-1\}$. Under backdoor attacks, an adversary injects a trigger pattern $\mathbf{m}$ into selected samples to form $\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{m}$. The label may remain unchanged in clean-label attacks or be changed to a target label in dirty-label attacks.

3.2 Client-Side Data Purification

The client-side purification pipeline is illustrated in Appendix C. At round T_{vae}, each client pretrains a CVAE on local data and obtains a purification model ϕ_k. The model is then reused in later rounds, and purification starts from $T_{\text{def}} > T_{\text{vae}}$.

3.2.1 CVAE Pretraining

The CVAE consists of an encoder $f_{\phi_k}(\mathbf{x}, \mathbf{y})$ and a decoder $g_{\phi_k}(\mathbf{z}, \mathbf{y})$, where $\mathbf{y}$ is the label condition and $\mathbf{z} = f_{\phi_k}(\mathbf{x}, \mathbf{y})$. The reconstruction is

$$\hat{\mathbf{x}} = g_{\phi_k}(\mathbf{z}, \mathbf{y}). \tag{2}$$

The pretraining objective is

$$\mathcal{L}_{\text{CVAE}}(\phi_k) = \frac{1}{N_k} \sum_{i=1}^{N_k} \left\| \mathbf{x}^{(i)} - g_{\phi_k}\left(f_{\phi_k}(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}), \mathbf{y}^{(i)}\right) \right\|_2^2. \tag{3}$$

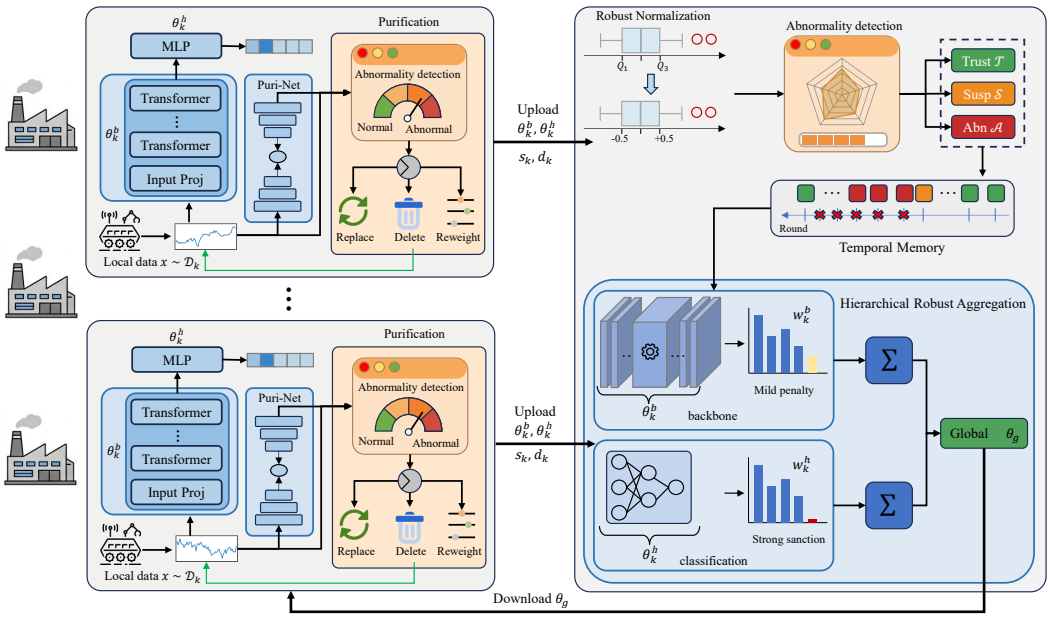


Figure 2: Overall framework of FedHRA.

The CVAE learns class-conditional normal structures, so trigger-induced deviations tend to produce larger reconstruction or prediction changes.

3.2.2 Suspiciousness Score

For a sample $\mathbf{x}^{(i)}$, FedHRA computes three scores. The first is prediction flipping:

$$s_{\text{flip}}(\mathbf{x}^{(i)}) = \mathbb{I}\left(\arg\max_c p_c(\mathbf{x}^{(i)}) \neq \arg\max_c p_c(\hat{\mathbf{x}}^{(i)})\right), \tag{4}$$

where $\hat{\mathbf{x}}^{(i)}$ is the CVAE reconstruction and $p_c(\cdot)$ is the model prediction probability for class c. The second is maximum confidence drop:

$$s_{\text{drop}}(\mathbf{x}^{(i)}) = \max\left(\max_c \left(p_c(\mathbf{x}^{(i)}) - p_c(\hat{\mathbf{x}}^{(i)})\right), 0\right). \tag{5}$$

The third is reconstruction error:

$$s_{\text{recon}}(\mathbf{x}^{(i)}) = \frac{1}{TD}\sum_{j=1}^{TD}\left(x_j^{(i)} - \hat{x}_j^{(i)}\right)^2. \tag{6}$$

The final suspiciousness score is

$$s(\mathbf{x}) = \lambda_1 s_{\text{drop}}(\mathbf{x}) + \lambda_2 s_{\text{recon}}(\mathbf{x}) + \lambda_3 s_{\text{flip}}(\mathbf{x}). \tag{7}$$

3.2.3 Threshold and Purification

Client k sets a suspicious threshold using a quantile of local scores:

$$\tau_k = \text{Quantile}_q\left(\{s(\mathbf{x}_i)\}\right). \tag{8}$$

The threshold is initialized when purification is first enabled and then kept fixed. Client k forms the suspicious set $\mathcal{Q}_k = \{i : s(\mathbf{x}^{(i)}) > \tau_k\}$. FedHRA supports replacement, deletion, and reweighting for the samples in $\mathcal{Q}_k$. In replacement, suspicious samples are replaced by their CVAE reconstructions. In deletion, they are removed. In reweighting, their training weights are reduced according to their scores. The client reports the suspicious-sample ratio s_k to the server. For the deletion strategy, client-side purification provably reduces the effective poisoning ratio when detection is accurate; the formal statement is in Appendix A.

3.2.4 HEAD DRIFT INDEX

After local training, client k computes the drift of its classification head relative to the previous global head:

$$\delta_{k,i}^{(t)} = \theta_{k,i}^{h,(t)} - \theta_{g,i}^{h,(t-1)}, \qquad \delta_k^{(t)} = [\delta_{k,1}^{(t)}, \delta_{k,2}^{(t)}, \ldots], \tag{9}$$

$$d_k^{(t)} = \|\delta_k^{(t)}\|_2. \tag{10}$$

The drift index captures how strongly the local classification boundary moves in the current round.

3.3 SERVER-SIDE HIERARCHICAL ROBUST AGGREGATION

The server-side hierarchical robust aggregation procedure is illustrated in Appendix C. After clients upload local models and risk signals, the server performs hierarchical robust aggregation.

3.3.1 ROBUST NORMALIZATION

The server receives the suspicious ratio s_k and drift index d_k. To reduce the influence of extreme values, each signal is normalized by the median and interquartile range (IQR):

$$d_k \leftarrow \max\left(0, \frac{d_k - \text{median}(\{d_j\})}{\text{IQR}(\{d_j\}) + \epsilon}\right), \tag{11}$$

$$s_k \leftarrow \max\left(0, \frac{s_k - \text{median}(\{s_j\})}{\text{IQR}(\{s_j\}) + \epsilon}\right). \tag{12}$$

3.3.2 INITIAL RISK SCORING

FedHRA computes separate risk scores for the backbone and head:

$$b_k = \gamma_s^b s_k + \gamma_d^b d_k, \qquad h_k = \gamma_s^h s_k + \gamma_d^h d_k. \tag{13}$$

The backbone uses milder penalties:

$$\gamma_s^b = \eta\gamma_s^h, \qquad \gamma_d^b = \eta\gamma_d^h, \qquad 0 < \eta < 1. \tag{14}$$

For the head score, a median absolute deviation (MAD) normalization is applied: $\tilde{h}_k^{(t)} = \frac{h_k^{(t)} - \text{median}(\{h_j^{(t)}\}_{j=1}^K)}{\text{MAD}(\{h_j^{(t)}\}_{j=1}^K) + \epsilon}$, where $\text{MAD}(\{h_j^{(t)}\}) = \text{median}\big(\{|h_j^{(t)} - \text{median}(\{h_\ell^{(t)}\})|\}\big)$. Clients are initially divided into abnormal, suspicious, and trusted sets:

$$\begin{aligned} \mathcal{A}_{\text{init}}^{(t)} &= \{k \mid \tilde{h}_k^{(t)} \geq T_{\text{blk}}\}, \\ \mathcal{S}_{\text{init}}^{(t)} &= \{k \mid T_{\text{susp}} \leq \tilde{h}_k^{(t)} < T_{\text{blk}}\}, \\ \mathcal{T}_{\text{init}}^{(t)} &= \{k \mid \tilde{h}_k^{(t)} < T_{\text{susp}}\}. \end{aligned} \tag{15}$$

3.3.3 TEMPORAL MEMORY

The server maintains a memory state $\mathbf{M}_k^{(t)} = (r_k^{(t)}, c_k^{(t)})$ for each client, where $r_k^{(t)}$ is the abnormality count and $c_k^{(t)}$ is the cooldown counter. With memory upper bound W, blocking threshold L, and cooldown length K_c, the update is

$$(r_k^{(t)}, c_k^{(t)}) = \begin{cases} (\min(W, r_k^{(t-1)} + 1), K_c), & k \in \mathcal{A}_{\text{init}}^{(t)} \cup \mathcal{S}_{\text{init}}^{(t)}, \\ (r_k^{(t-1)}, c_k^{(t-1)} - 1), & k \in \mathcal{T}_{\text{init}}^{(t)}, \\ & c_k^{(t-1)} > 0, \\ (\max(0, r_k^{(t-1)} - 1), 0), & k \in \mathcal{T}_{\text{init}}^{(t)}, \\ & c_k^{(t-1)} = 0. \end{cases} \tag{16}$$

The final client sets are

$$\begin{aligned} \mathcal{A}_{\text{final}}^{(t)} &= \mathcal{A}_{\text{init}}^{(t)} \cup \{k \mid r_k^{(t)} \geq L\}, \\ \mathcal{S}_{\text{final}}^{(t)} &= \mathcal{S}_{\text{init}}^{(t)} \setminus \mathcal{A}_{\text{final}}^{(t)}, \\ \mathcal{T}_{\text{final}}^{(t)} &= \mathcal{U} \setminus (\mathcal{S}_{\text{final}}^{(t)} \cup \mathcal{A}_{\text{final}}^{(t)}). \end{aligned} \tag{17}$$

3.3.4 HIERARCHICAL WEIGHT MAPPING AND SANCTION

FedHRA maps risk scores to backbone and head weights using negative-softmax weighting:

$$w_k^b = \frac{\exp(-\beta_b b_k)}{\sum_{j=1}^K \exp(-\beta_b b_j)}, \quad w_k^h = \frac{\exp(-\beta_h h_k)}{\sum_{j=1}^K \exp(-\beta_h h_j)}. \tag{18}$$

Backbone weights of suspicious and abnormal clients are softly scaled:

$$\tilde{w}_k^b = \begin{cases} \rho_{\text{susp}} w_k^b, & k \in \mathcal{S}_{\text{final}}^{(t)}, \\ \rho_{\text{blk}} w_k^b, & k \in \mathcal{A}_{\text{final}}^{(t)}, \\ w_k^b, & k \in \mathcal{T}_{\text{final}}^{(t)}. \end{cases} \tag{19}$$

The normalized backbone weight is

$$w_k^b \leftarrow \frac{\tilde{w}_k^b}{\sum_{j=1}^K \tilde{w}_j^b}. \tag{20}$$

For the classification head, FedHRA uses stronger sanctions:

$$\tilde{w}_k^h = \begin{cases} 0, & k \in \mathcal{S}_{\text{final}}^{(t)} \cup \mathcal{A}_{\text{final}}^{(t)}, \\ & \kappa = 0, \\ \min(w_k^h, \kappa), & k \in \mathcal{S}_{\text{final}}^{(t)} \cup \mathcal{A}_{\text{final}}^{(t)}, \\ & \kappa > 0, \\ w_k^h, & k \in \mathcal{T}_{\text{final}}^{(t)}. \end{cases} \tag{21}$$

Then

$$w_k^h \leftarrow \frac{\tilde{w}_k^h}{\sum_{j=1}^K \tilde{w}_j^h}. \tag{22}$$

The global model is updated by

$$\theta_{g,p} \leftarrow \begin{cases} \sum_{k=1}^K w_k^b \theta_{k,p}, & p \in \mathcal{P}_b, \\ \sum_{k=1}^K w_k^h \theta_{k,p}, & p \in \mathcal{P}_h. \end{cases} \tag{23}$$

The complete FedHRA training procedure is summarized in Algorithm 1.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Two datasets are used. The Tennessee Eastman Process (TEP) dataset is a standard chemical process benchmark with multivariate measurements and manipulated variables (Downs & Vogel, 1993). We use 15 predefined fault conditions for multivariate time-series classification. The Spoken Arabic Digits (SAD) dataset contains Arabic digit utterances represented by sequential acoustic features (Alotaibi, 2005). SAD is used as a complementary time-series benchmark to evaluate transferability beyond process data.

We evaluate four attack schedules. A single-shot attack is launched only at round 20. A continuous attack is launched from round 20 to round 100. A periodic attack starts at round 20 and repeats every 10 rounds. A late-stage attack is launched during the last 20% of training rounds, i.e., from round 80 to round 100.

The system trains for 100 rounds with 10 clients. Statistical heterogeneity is generated by Dirichlet partitioning with parameter $\alpha = 0.1$. The malicious-client ratio is $\rho_m = 0.2$, and the poisoned-sample ratio on each malicious client is 0.1. Each client trains for 10 local epochs with batch size 32 and learning rate 10^{-3}. The local model consists of a time-series Transformer encoder and a

Algorithm 1 FedHRA training

Require: Clients $\mathcal{U}$, rounds R, initial model $\boldsymbol{\theta}_g^{(0)}$, purification weights $(\lambda_1, \lambda_2, \lambda_3)$, thresholds $(T_{\text{susp}}, T_{\text{blk}})$, memory (W, L, K_c), temperatures (β_b, β_h), sanctions $(\rho_{\text{susp}}, \rho_{\text{blk}}, \kappa)$.

Ensure: Global model $\boldsymbol{\theta}_g^{(R)}$.

```
Pretrain a local CVAE φ_k for each client k and initialize M_k^(0) = (0, 0).
for t = 1 to R do
    Broadcast θ_g^(t−1).
    for all clients k ∈ U in parallel do
        Reconstruct samples with φ_k and compute s(x).
        Identify suspicious set Q_k and purify local data.
        Train local model on purified data to obtain θ_k^(t).
        Compute suspicious ratio s_k and head drift d_k.
        Upload (θ_k^(t), s_k, d_k).
    end for
    Robustly normalize s_k and d_k, then compute b_k and h_k.
    Obtain initial client sets by equation 15.
    Update temporal memory by equation 16 and obtain final client sets.
    Compute hierarchical weights by equation 18 and apply sanctions.
    Aggregate backbone and head parameters separately.
end for
return θ_g^(R).
```

classification head. CVAE pretraining starts at round 5, and defense is enabled from round 6. The purification strategy is replacement. The client-side weights are $\lambda_1 = \lambda_3 = 1$ and $\lambda_2 = 10$. For server-side scoring, $\gamma_s^h = 2.0$, $\gamma_d^h = 0.4$, $\gamma_s^b = 0.6$, and $\gamma_d^b = 0.12$. The thresholds are $T_{\text{susp}} = 3.5$ and $T_{\text{blk}} = 4.5$. The blocking threshold is $L = 12$, the cooldown length is $K_c = 2$, and the hierarchical softmax temperatures are $\beta_b = 3.5$ and $\beta_h = 8.0$. Two metrics are used. Main Task Accuracy (MTA) measures clean test accuracy. Terminal Attack Success Rate (TASR) measures the attack success rate on triggered samples at the end of federated training.

4.2 Comparison with Baselines

We compare FedHRA with No defense, Krum (Blanchard et al., 2017), Multi-Krum (Blanchard et al., 2017), coordinate-wise Median (Yin et al., 2018), Trimmed Mean (Yin et al., 2018), RFA (Pillutla et al., 2022), FoolsGold (Fung et al., 2018), FLAME (Nguyen et al., 2022), DeepSight (Rieger et al., 2022), FLDetector (Zhang et al., 2022), GAS (Liu et al., 2023), and Bucketing (Karimireddy et al., 2020a). For methods requiring the malicious-client-ratio prior, the true ratio $\rho_m = 0.2$ is used. Other hyperparameters are fixed across datasets and attack schedules.

Table 1 shows that continuous and late-stage attacks are the most threatening. Without defense, TASR reaches 96.75% and 81.50% on TEP, and 100.00% and 98.33% on SAD. FedHRA keeps TASR close to zero across both datasets and all attack schedules. On TEP, FedHRA reduces continuous-attack TASR from 96.75% to 0.42% and late-stage TASR from 81.50% to 0.17%. On SAD, it reduces continuous-attack TASR from 100.00% to 0.00% and keeps late-stage TASR at 2.33%. Several baselines maintain high MTA but fail to suppress continuous or late-stage attacks. For example, Multi-Krum, Trimmed Mean, FLAME, DeepSight, and FLDetector often keep TASR near 100% under continuous attacks. Krum can suppress TASR in some cases but severely damages MTA, especially on TEP. FedHRA therefore does not maximize MTA in every scenario: on TEP under single-shot attacks, its MTA is 82.05%, about nine points below No defense (90.87%), the price of stable backdoor suppression.

4.3 Ablation Study

Table 2 verifies the complementarity of the two modules. Removing client-side defense causes severe security failures under continuous and late-stage attacks: TASR rises to 97.83% and 92.00% on TEP and reaches 100.00% in both settings on SAD. This indicates that edge-side purification is es-

Table 1: Comparison of defense methods on TEP and SAD under four attack schedules, reporting main-task accuracy (MTA) and terminal attack success rate (TASR) in percent.

Dataset	Defense	Single-shot		Continuous		Periodic		Late-stage	
		MTA	TASR	MTA	TASR	MTA	TASR	MTA	TASR
TEP	No defense	90.87	0.42	89.41	96.75	89.13	3.83	91.21	81.50
	FoolsGold	87.84	0.83	82.05	26.75	89.56	69.67	82.67	60.08
	Krum	53.39	0.00	39.33	0.00	35.14	0.00	32.32	0.00
	Multi-Krum	90.69	1.58	90.46	97.50	89.16	2.33	89.69	92.42
	Median	90.39	0.33	89.04	96.08	89.65	0.42	89.96	78.42
	RFA	89.76	1.08	90.17	94.42	89.27	1.67	90.10	64.92
	Trimmed Mean	90.25	0.08	89.48	99.67	90.70	0.17	91.01	88.67
	FLAME	90.59	0.33	90.03	83.58	88.66	1.25	91.64	88.25
	DeepSight	89.70	0.25	88.40	92.33	91.73	8.08	90.35	98.75
	FLDetector	90.72	0.33	92.98	97.25	88.62	0.00	88.17	85.58
	GAS	89.55	0.25	89.07	89.83	89.22	3.58	89.22	71.25
	Bucketing	85.65	0.25	83.76	99.42	85.36	39.83	83.23	92.83
	FedHRA	82.05	0.42	87.53	0.42	85.56	0.08	87.73	0.17
SAD	No defense	97.64	0.33	98.59	100.00	97.68	22.67	96.09	98.33
	FoolsGold	96.04	1.00	97.00	99.33	96.91	79.67	86.90	98.00
	Krum	57.75	21.67	57.75	21.67	57.80	21.00	57.75	19.33
	Multi-Krum	97.86	0.33	98.45	100.00	99.05	0.33	99.05	1.00
	Median	98.45	0.00	98.64	85.33	98.23	0.00	98.64	0.00
	RFA	98.27	0.00	98.27	16.67	98.27	0.33	98.32	0.33
	Trimmed Mean	98.64	0.00	98.73	100.00	98.64	0.00	98.04	1.33
	FLAME	98.18	0.00	98.41	100.00	98.27	1.00	98.50	0.33
	DeepSight	98.82	0.00	98.68	100.00	98.50	0.00	97.32	94.67
	FLDetector	96.59	1.00	97.64	100.00	97.18	78.33	98.18	99.67
	GAS	98.45	0.00	98.64	85.33	98.23	0.00	98.64	0.00
	Bucketing	98.54	0.33	97.95	100.00	98.36	72.67	98.36	100.00
	FedHRA	97.86	0.00	97.95	0.00	97.09	1.00	97.27	2.33

Table 2: Ablation study on TEP and SAD.

Dataset	Setting	Single-shot		Continuous		Periodic		Late-stage	
		MTA	TASR	MTA	TASR	MTA	TASR	MTA	TASR
TEP	FedHRA	82.05	0.42	87.53	0.42	85.56	0.08	87.73	0.17
	w/o hierarchical aggregation	73.69	0.33	73.27	0.00	82.26	0.33	81.64	0.08
	w/o client defense	89.08	0.58	89.07	97.83	87.45	56.25	91.92	92.00
SAD	FedHRA	97.86	0.00	97.95	0.00	97.09	1.00	97.27	2.33
	w/o hierarchical aggregation	90.00	0.00	88.22	0.00	80.90	7.33	49.34	48.67
	w/o client defense	98.59	0.00	99.09	100.00	97.77	0.33	98.82	100.00

sential for suppressing the source of backdoor gradients. Removing hierarchical aggregation keeps TASR low in some TEP settings but causes large MTA degradation, and on SAD it also causes both accuracy and security degradation under late-stage attacks. This indicates that client-side purification alone cannot robustly handle non-IID updates and residual backdoor effects. Server-side hierarchical aggregation is needed to stabilize global fusion and suppress decision-boundary manipulation.

4.4 Sensitivity Analysis

Fig. 3 reports the sensitivity analysis on TEP, where (a)–(d) cover the poisoned sample ratio and (e)–(h) cover the malicious client ratio. For the poisoned sample ratio, TASR stays low across the four attack schedules and does not increase monotonically with poisoning strength, indicating that FedHRA is not overly sensitive to single-client poisoning intensity; MTA fluctuates mildly, espe-

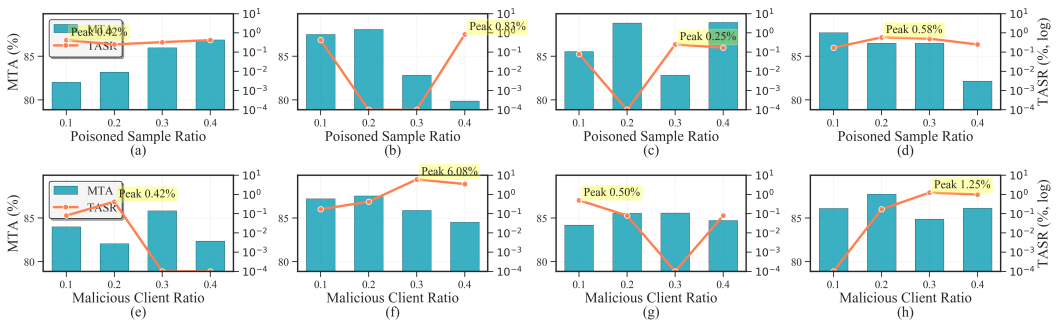


Figure 3: Sensitivity analysis on TEP. Panels (a)–(d) vary the poisoned sample ratio; panels (e)–(h) vary the malicious client ratio.

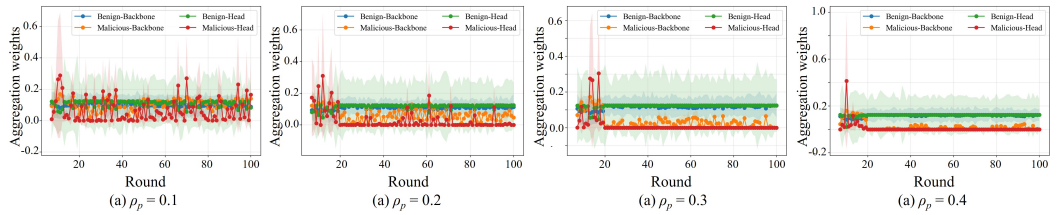


Figure 4: Aggregation weight dynamics under different poisoned sample ratios.

cially under continuous attacks, reflecting the security-utility tradeoff under stronger local pollution. For the malicious client ratio, FedHRA keeps TASR low under single-shot and periodic attacks even as the ratio increases from 0.1 to 0.4. Under continuous and late-stage attacks, TASR increases at higher malicious client ratios but remains far below the undefended case; MTA remains usable, although stronger suppression under larger malicious scale introduces some accuracy cost.

4.5 Aggregation Weight Dynamics

Fig. 4 records the average aggregation weights of benign and malicious clients under continuous attacks with $\rho_p \in \{0.1, 0.2, 0.3, 0.4\}$. Malicious client weights in the classification head are continuously suppressed and remain close to zero for most rounds. In the backbone, malicious weights are reduced but not completely eliminated, while benign weights remain relatively stable. This matches the intended behavior of FedHRA: the decision-sensitive head is strongly protected, while the representation-learning backbone retains limited heterogeneous information when it is not overwhelmingly risky.

5 Conclusion

FedHRA combines client-side purification with server-side hierarchical robust aggregation for federated time-series classification. On the client side, FedHRA uses a CVAE-based purification module to identify and purify suspicious samples through prediction flipping, confidence drop, and reconstruction error, thereby reducing the effective poisoning ratio before local training. On the server side, FedHRA combines client-reported risk evidence and head-parameter drift, uses robust normalization and temporal memory to identify suspicious clients, and performs hierarchical aggregation with strong sanctions on the classification head and mild sanctions on the backbone. The theoretical analysis (Appendix A) explains why this hierarchy is appropriate and shows that FedHRA adds limited communication overhead. Experiments on TEP and SAD show that FedHRA consistently suppresses terminal attack success under multiple attack schedules while maintaining usable main-task accuracy. Ablation, sensitivity, and weight-dynamics analyses confirm that edge-side purification and cloud-side hierarchical aggregation are both necessary for robust defense under statistical heterogeneity.

AI USE STATEMENT

The authors used a generative AI assistant during manuscript preparation for language polishing. The problem formulation, proposed method, theoretical analysis, experimental design, and all reported results were produced by the authors. The authors take full responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Yousef Ajami Alotaibi. Investigating spoken Arabic digits in speech recognition setting. *Information Sciences*, 173(1-3):115–139, 2005.

Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv Preprint arXiv:1912.00818*, 2019.

Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 2938–2948. PMLR, 2020.

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, volume 30, pp. 118–128, 2017.

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, 2017.

Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. FLTrust: Byzantine-robust federated learning via trust bootstrapping. *arXiv Preprint arXiv:2012.13995*, 2020.

J.J. Downs and E.F. Vogel. A plant-wide industrial process control problem. *Computers & Chemical Engineering*, 17(3):245–255, 1993. ISSN 0098-1354.

Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. Mitigating sybils in federated learning poisoning. *arXiv Preprint arXiv:1808.04866*, 2018.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv Preprint arXiv:1708.06733*, 2017.

Chao Huang, Jianwei Huang, and Xin Liu. Cross-silo federated learning: Challenges and opportunities. *arXiv Preprint arXiv:2206.12949*, 2022.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2):1–210, 2021.

Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Byzantine-robust learning on heterogeneous datasets via bucketing. *arXiv Preprint arXiv:2006.09365*, 2020a.

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

Nikita Kotelevskii, Maxime Vono, Alain Durmus, and Eric Moulines. FedPop: A bayesian approach for personalised federated learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 8687–8701, 2022.

Jiewu Leng, Rongjie Li, Junxing Xie, Xueliang Zhou, Xiang Li, Qiang Liu, Xin Chen, Weiming Shen, and Lihui Wang. Federated learning-empowered smart manufacturing and product lifecycle management: A review. *Advanced Engineering Informatics*, 65:103179, 2025.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. FedBN: Federated learning on non-IID features via local batch normalization. *arXiv Preprint arXiv:2102.07623*, 2021a.

Yuanchun Li, Jiayi Hua, Haoyu Wang, Chunyang Chen, and Yunxin Liu. DeepPayload: Black-box backdoor attack on deep learning models through neural payload injection. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pp. 263–274. IEEE, 2021b.

Haibo Liu, Jianfeng Lu, Xiong Wang, Chen Wang, Riheng Jia, and Minglu Li. FedUp: Bridging fairness and efficiency in cross-silo federated learning. *IEEE Transactions on Services Computing*, 17(6):3672–3684, 2024.

Ken Liu, Shengyuan Hu, Steven Z Wu, and Virginia Smith. On privacy and personalization in cross-silo federated learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 5925–5940, 2022.

Yuchen Liu, Chen Chen, Lingjuan Lyu, Fangzhao Wu, Sai Wu, and Gang Chen. Byzantine-robust learning on heterogeneous data via gradient splitting. In *International Conference on Machine Learning*, pp. 21404–21425. PMLR, 2023.

Shiwei Lu, Ruihu Li, Wenbin Liu, and Xuan Chen. Defense against backdoor attack in federated learning. *Computers & Security*, 121:102819, 2022.

Xiaosong Ma, Jie Zhang, Song Guo, and Wenchao Xu. Layer-wised model aggregation for personalized federated learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10092–10101, 2022.

Koji Matsuda, Yuya Sasaki, Chuan Xiao, and Makoto Onizuka. An empirical study of personalized federated learning. *arXiv Preprint arXiv:2206.13190*, 2022.

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

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

Jean Ogier du Terrail, Samy-Safwan Ayed, Edwige Cyffers, Felix Grimberg, Chaoyang He, Regis Loeb, Paul Mangold, Tanguy Marchand, Othmane Marfoq, Erum Mushtaq, et al. Flamby: Datasets and benchmarks for cross-silo federated learning in realistic healthcare settings. In *Advances in Neural Information Processing Systems*, volume 35, pp. 5315–5334, 2022.

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

Rossella Pozzi, Tommaso Rossi, and Raffaele Secchi. Industry 4.0 technologies: Critical success factors for implementation and improvements in manufacturing companies. *Production Planning & Control*, 34(2):139–158, 2023.

Phillip Rieger, Thien Duc Nguyen, Markus Miettinen, and Ahmad-Reza Sadeghi. DeepSight: Mitigating backdoor attacks in federated learning through deep model inspection. *arXiv Preprint arXiv:2201.00763*, 2022.

Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14318–14328, 2022.

Vignesh Sampath, Iñaki Maurtua, Juan José Aguilar Martín, Andoni Rivera, Jorge Molina, and Aitor Gutierrez. Attention-guided multitask learning for surface defect identification. *IEEE Transactions on Industrial Informatics*, 19(9):9713–9721, 2023.

Maryam Vahabi, Hossein Fotouhi, et al. Federated learning at the edge in industrial internet of things: A review. *Sustainable Computing: Informatics and Systems*, 46:101087, 2025.

Baicun Wang, Fei Tao, Xudong Fang, Chao Liu, Yufei Liu, and Theodor Freiheit. Smart manufacturing and intelligent manufacturing: A comparative review. *Engineering*, 7(6):738–757, 2021.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7611–7623, 2020.

Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. DBA: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations*, 2020.

Zuobin Xiong, Zhipeng Cai, Daniel Takabi, and Wei Li. Privacy threat and defense for federated learning with non-IID data in AIoT. *IEEE Transactions on Industrial Informatics*, 18(2):1310–1321, 2021.

Mingfu Xue, Can He, Yinghao Wu, Shichang Sun, Yushu Zhang, Jian Wang, and Weiqiang Liu. PTB: Robust physical backdoor attacks against deep neural networks in real world. *Computers & Security*, 118:102726, 2022.

Peishen Yan, Jun Li, Hao Wang, Tao Song, Yang Hua, Lu Peng, Haihui Zhou, and Haibing Guan. FedCod: An efficient communication protocol for cross-silo federated learning with coding. *arXiv Preprint arXiv:2501.00216*, 2024.

Pei Ye, Yuqing Li, Kun He, Qiao Li, Tianjie Qin, Xiong Wang, Kaige Yang, Chujun Zhang, and Jing Chen. Breaking the illusion: A critical study of backdoor defense in federated learning with non-IID data. *IEEE Transactions on Information Forensics and Security*, 20:13264–13279, 2025.

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

Jiwei Zhang, Md Zakirul Alam Bhuiyan, Xu Yang, Amit Kumar Singh, D Frank Hsu, and Entao Luo. Trustworthy target tracking with collaborative deep reinforcement learning in EdgeAI-aided IoT. *IEEE Transactions on Industrial Informatics*, 18(2):1301–1309, 2021.

Zaixi Zhang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. FLDetector: Defending federated learning against model poisoning attacks via detecting malicious clients. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2545–2555, 2022.

Xiaokang Zhou, Xuesong Xu, Wei Liang, Zhi Zeng, Shohei Shimizu, Laurence T Yang, and Qun Jin. Intelligent small object detection for digital twin in smart manufacturing with industrial cyber-physical systems. *IEEE Transactions on Industrial Informatics*, 18(2):1377–1386, 2021.

A Theoretical Analysis

A.1 Effective Poisoning Ratio after Deletion

Consider the deletion strategy for analysis. Suppose the local training data are drawn from the mixture

$$(\mathbf{x}, y) \sim (1 - \pi_k)\mathcal{D}_k + \pi_k \mathcal{D}_k^{\mathrm{bd}}, \tag{24}$$

where π_k is the backdoor-sample ratio. Define the suspicious event

$$A = \{s(\mathbf{x}) > \tau_k\}. \tag{25}$$

Let

$$\mathrm{TPR}_k = \mathbb{P}\big(A \mid (\mathbf{x}, y) \sim \mathcal{D}_k^{\mathrm{bd}}\big), \tag{26}$$

$$\mathrm{FPR}_k = \mathbb{P}\big(A \mid (\mathbf{x}, y) \sim \mathcal{D}_k\big). \tag{27}$$

Proposition 1 (Effective Poisoning Ratio after Deletion)**.** *If client-side purification deletes all samples satisfying A, the effective backdoor ratio in the retained samples is*

$$\pi_k^{\mathrm{del}} = \frac{\pi_k(1-\mathrm{TPR}_k)}{\substack{(1-\pi_k)(1-\mathrm{FPR}_k)\\ +\pi_k(1-\mathrm{TPR}_k)}}. \tag{28}$$

The probability of a sample being flagged is

$$\mathbb{P}(A) = (1-\pi_k)\mathrm{FPR}_k + \pi_k \mathrm{TPR}_k. \tag{29}$$

The proof is provided in Appendix B.

This result shows that when TPR_k is high and FPR_k is low, client-side purification reduces the effective poisoning ratio before aggregation.

A.2 Rationale of Hierarchical Weighting

To explain why the classification head and backbone should be treated differently, consider the last classification layer as a linear classifier. Let the backbone feature be

$$\mathbf{z}(\mathbf{x}) = f_b(\mathbf{x}; \boldsymbol{\theta}^b), \tag{30}$$

and the logit for class c be

$$a_c(\mathbf{x}) = \langle \mathbf{w}_c, \mathbf{z}(\mathbf{x}) \rangle + b_c. \tag{31}$$

For true class y and target wrong class ℓ, define the margin

$$m(\mathbf{x}) = a_\ell(\mathbf{x}) - a_y(\mathbf{x}). \tag{32}$$

A larger margin makes the sample more likely to be misclassified into the target class.

Proposition 2 (Effect of Head and Backbone on Margin)**.** *Let local perturbations of the head and backbone be $\Delta\boldsymbol{\theta}^h$ and $\Delta\boldsymbol{\theta}^b$. If the backbone is fixed, changing the head gives*

$$\Delta m(\mathbf{x}) = \langle \Delta\mathbf{w}_\ell - \Delta\mathbf{w}_y, \mathbf{z}(\mathbf{x}) \rangle + \Delta b_\ell - \Delta b_y. \tag{33}$$

If the head is fixed, changing the backbone gives

$$\Delta m(\mathbf{x}) = \langle \mathbf{w}_\ell - \mathbf{w}_y, \Delta\mathbf{z}(\mathbf{x}) \rangle. \tag{34}$$

If f_b is differentiable around the current parameters and $J_b(\mathbf{x})$ is the Jacobian of $\mathbf{z}(\mathbf{x})$ with respect to $\boldsymbol{\theta}^b$, then

$$|\Delta m(\mathbf{x})| \le \|\mathbf{w}_\ell - \mathbf{w}_y\|_2 \|J_b(\mathbf{x})\| \|\Delta\boldsymbol{\theta}^b\|_2. \tag{35}$$

This proposition indicates that head parameters directly control class margins, whereas backbone perturbations affect margins through feature mapping. Thus, FedHRA applies stronger suppression to the head and softer suppression to the backbone.

A.3 Weight Suppression Properties

Let $\mathcal{U}^{(t)}$ be the participating clients at round t, with benign clients $\mathcal{B}^{(t)}$ and malicious clients $\mathcal{M}^{(t)}$.

Theorem 1 (Malicious Head-Weight Upper Bound)**.** *Assume that every malicious client has a head risk score at least $\Delta_h^{(t)} > 0$ larger than every benign client, i.e.,*

$$\begin{aligned} h_m^{(t)} &\ge h_u^{(t)} + \Delta_h^{(t)}, \\ m \in \mathcal{M}^{(t)}, &\quad u \in \mathcal{B}^{(t)}. \end{aligned} \tag{36}$$

Then the total malicious softmax weight in the head satisfies

$$\Omega_{h,\mathrm{sm}}^{(t)} \le \frac{|\mathcal{M}^{(t)}| \exp(-\beta_h \Delta_h^{(t)})}{\substack{(|\mathcal{U}^{(t)}|-|\mathcal{M}^{(t)}|)\\ +|\mathcal{M}^{(t)}| \exp(-\beta_h \Delta_h^{(t)})}}. \tag{37}$$

Thus, increasing β_h exponentially reduces malicious head contribution when a score gap exists.

Proposition 3 (Upper Bound after Head Capping). *Let $\mathcal{R}^{(t)} = \mathcal{U}^{(t)} \setminus \mathcal{T}^{(t)}_{\text{final}}$ be the nontrusted set. If nontrusted head weights are capped by κ, their total normalized head weight satisfies*

$$\Omega^{(t)}_{h,\text{cap}} \le \frac{|\mathcal{R}^{(t)}|\kappa}{\sum_{u\in\mathcal{T}^{(t)}_{\text{final}}} \widehat{w}^{h,(t)}_u + |\mathcal{R}^{(t)}|\kappa}. \tag{38}$$

When $\kappa = 0$, $\Omega^{(t)}_{h,\text{cap}} = 0$.

For the backbone, hard removal of nontrusted clients can enlarge benign reweighting bias under false positives.

Proposition 4 (Backbone Aggregation Bias). *Let $\Delta^{(t)}_{k,b} = \boldsymbol{\theta}^{b,(t)}_k - \boldsymbol{\theta}^{b,(t-1)}_g$ and assume $\|\Delta^{(t)}_{k,b}\|_2 \le G_b$. Let*

$$\widehat{g}^{(t)}_b = \sum_{k\in\mathcal{U}^{(t)}} w^{b,(t)}_k \Delta^{(t)}_{k,b} \tag{39}$$

be the FedHRA backbone aggregation direction, and

$$g^{(t)}_b = \sum_{u\in\mathcal{B}^{(t)}} p^{(t)}_u \Delta^{(t)}_{u,b} \tag{40}$$

be the ideal benign aggregation direction. Then

$$\begin{aligned}\|\widehat{g}^{(t)}_b - g^{(t)}_b\|_2 \le G_b \sum_{m\in\mathcal{M}^{(t)}} w^{b,(t)}_m \\ + G_b \sum_{u\in\mathcal{B}^{(t)}} |w^{b,(t)}_u - p^{(t)}_u|.\end{aligned} \tag{41}$$

This proposition explains why FedHRA softly downweights the backbone: reducing malicious residual weight is useful, but overly aggressive rejection may increase benign reweighting bias.

B Proofs and Complexity Derivations

B.1 Proof of the Effective Poisoning Ratio after Deletion

Let $Z = 1$ denote a backdoor sample and $Z = 0$ denote a benign sample. Then $\mathbb{P}(Z = 1) = \pi_k$ and $\mathbb{P}(Z = 0) = 1 - \pi_k$. After deletion, the retained event is A^c, so

$$\begin{aligned}\pi^{\text{del}}_k &= \mathbb{P}(Z = 1 \mid A^c) \\ &= \frac{\mathbb{P}(A^c \mid Z = 1)\mathbb{P}(Z = 1)}{\mathbb{P}(A^c)}.\end{aligned} \tag{42}$$

Using $\mathbb{P}(A^c \mid Z = 1) = 1 - \text{TPR}_k$ and $\mathbb{P}(A^c \mid Z = 0) = 1 - \text{FPR}_k$ gives the first result. The second follows directly from the law of total probability.

B.2 Proof of the Margin Proposition

Restate the margin as

$$m(\mathbf{x}) = \langle \mathbf{w}_\ell - \mathbf{w}_y, \mathbf{z}(\mathbf{x})\rangle + b_\ell - b_y. \tag{43}$$

Equations equation 33 and equation 34 follow by expanding this identity under head-only and backbone-only perturbations, respectively. For the bound, Cauchy's inequality and the first-order approximation $\|\Delta\mathbf{z}(\mathbf{x})\|_2 \le \|J_b(\mathbf{x})\|\|\Delta\boldsymbol{\theta}^b\|_2$ together give

$$|\Delta m(\mathbf{x})| \le \|\mathbf{w}_\ell - \mathbf{w}_y\|_2 \|J_b(\mathbf{x})\|\|\Delta\boldsymbol{\theta}^b\|_2. \tag{44}$$

B.3 Proof of the Malicious Head-Weight Upper Bound

For $m \in \mathcal{M}^{(t)}$ and $u \in \mathcal{B}^{(t)}$, the assumed score gap $h_m^{(t)} \geq h_u^{(t)} + \Delta_h^{(t)}$ implies

$$\exp(-\beta_h h_m^{(t)}) \leq \exp(-\beta_h \Delta_h^{(t)}) \exp(-\beta_h h_u^{(t)}). \tag{45}$$

Summing over malicious clients and comparing with the benign denominator of the softmax

$$\Omega_{h,\mathrm{sm}}^{(t)} = \frac{\sum_{m \in \mathcal{M}^{(t)}} \exp(-\beta_h h_m^{(t)})}{\sum_{k \in \mathcal{U}^{(t)}} \exp(-\beta_h h_k^{(t)})} \tag{46}$$

yields the stated bound.

B.4 Proof of the Head-Capping Bound

For each $k \in \mathcal{R}^{(t)}$, the unnormalized capped weight satisfies $\widehat{w}_k^{h,(t)} \leq \kappa$. Summing over $\mathcal{R}^{(t)}$ and normalizing by the total weight gives the stated upper bound; when $\kappa = 0$ the nontrusted head weight vanishes.

B.5 Proof of the Backbone Aggregation Bias

Expand $\widehat{g}_b^{(t)} - g_b^{(t)}$ into malicious residual terms and benign reweighting terms, then apply the triangle inequality and the bounded-update assumption $\|\Delta_{k,b}^{(t)}\|_2 \leq G_b$.

B.6 Computational Complexity Derivation

Let n_k be the number of samples on client k, B be the batch size, P_b and P_h be the backbone and head parameter sizes, $P_M = P_b + P_h$, and P_v be the CVAE parameter size. Let M_k be the local training steps and M_k^{vae} be the one-time CVAE pretraining steps.

The one-time CVAE pretraining cost is

$$\mathcal{T}_k^{\text{vae-pre}} = \mathcal{O}(M_k^{\mathrm{vae}} P_v). \tag{47}$$

In each FL round, client-side purification requires original-sample inference, CVAE reconstruction, and reconstructed-sample inference. The per-client edge cost is

$$\mathcal{T}_k^{\mathrm{edge}} = \mathcal{O}\left(\frac{n_k}{B}(2P_M + P_v) + M_k P_M\right). \tag{48}$$

Server-side risk processing is $\mathcal{O}(K \log K)$, and parameter aggregation is

$$\mathcal{T}_{\mathrm{server}}^{\mathrm{agg}} = \mathcal{O}(KP_b + KP_h) = \mathcal{O}(KP_M). \tag{49}$$

Therefore, the per-round computation of FedHRA is

$$\begin{aligned} \mathcal{T}_{\mathrm{FedHRA}} = \mathcal{O}\Bigg(& \sum_{k=1}^{K} \left[\frac{n_k}{B}(2P_M + P_v) + M_k P_M\right] \\ & + K \log K + KP_M \Bigg). \end{aligned} \tag{50}$$

Compared with FedAvg, the additional repeated cost mainly comes from edge-side purification and low-dimensional server-side risk processing. The server aggregation remains in the same order as FedAvg.

B.7 Communication Complexity Derivation

FedHRA sends and receives model parameters as in FedAvg. In addition, each client uploads Q low-dimensional risk statistics. The per-round communication cost is

$$\begin{aligned} \mathcal{C}_{\mathrm{FedHRA}} &= \mathcal{O}(KP_M) + \mathcal{O}(KP_M) + \mathcal{O}(KQ) \\ &= \mathcal{O}(KP_M + KQ). \end{aligned} \tag{51}$$

Since $Q \ll P_M$, the extra communication overhead is negligible compared with model transmission.

C METHOD FRAMEWORK ILLUSTRATIONS

Fig. 5 details the client-side purification pipeline, and Fig. 6 details the server-side hierarchical robust aggregation.

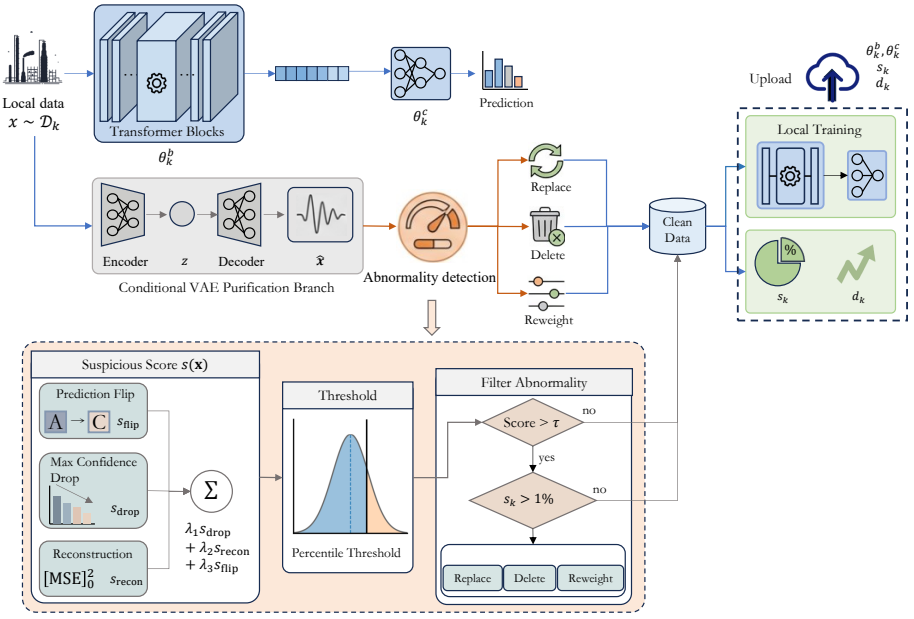


Figure 5: Client-side data purification. A conditional VAE reconstructs each local time-series sample onto the class-conditional normal manifold; suspiciousness is scored from prediction flipping, maximum confidence drop, and reconstruction error, and risky samples are purified by replacement, deletion, or reweighting before local training.

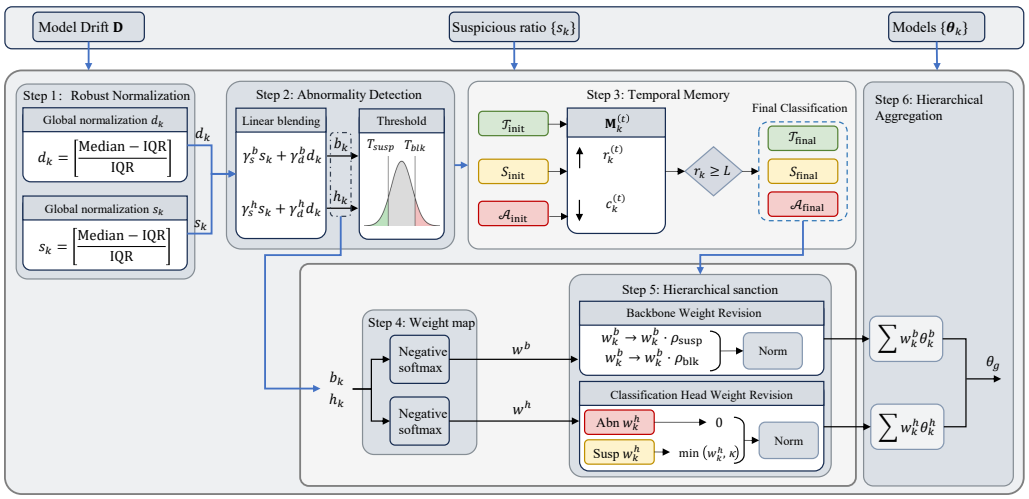


Figure 6: Server-side hierarchical robust aggregation. Steps 1 to 6 cover robust normalization, abnormality detection, temporal memory, weight mapping, hierarchical sanction, and hierarchical aggregation.