IJCAI 2025 MONTREAL

# FedHAN: A Cache-Based Semi-Asynchronous Federated Learning Framework Defending Against Poisoning Attacks in Heterogeneous Clients

Xiaoding Wang[1], Bin Ye[1], Li Xu[1], Lizhao Wu[1], Sun-Yuan Hsieh[2], Jie Wu[3,4], Limei Lin[1*]

[1]Fujian Normal University

[2]National Cheng Kung University

[3]China Telecom Cloud Computing Research Institute

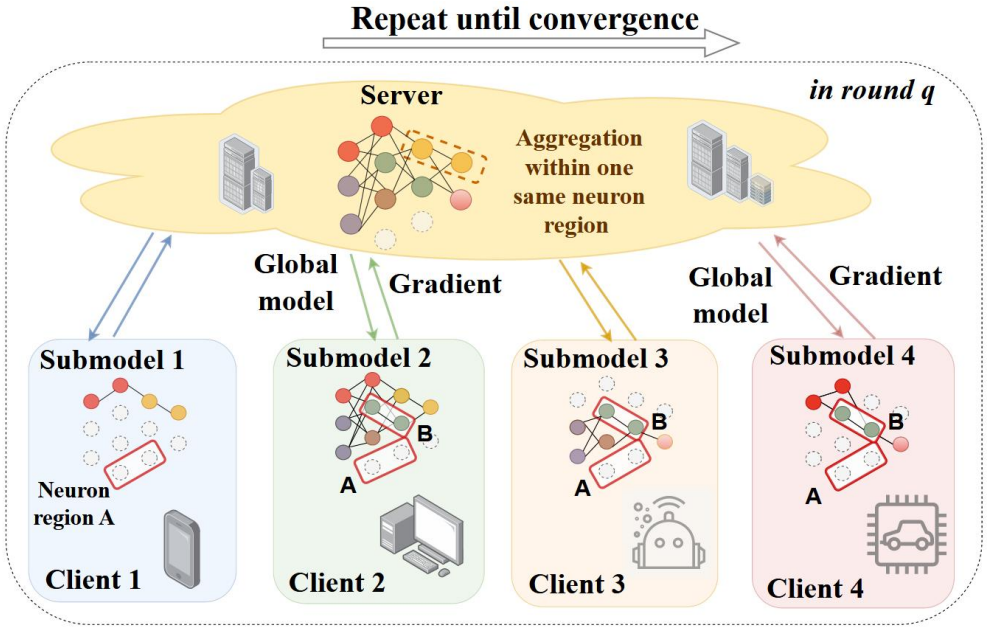
[4]Temple University

福建師範大學 FUJIAN NORMAL UNIVERSITY

國立成功大學 National Cheng Kung University

TEMPLE UNIVERSITY

# Challenges:

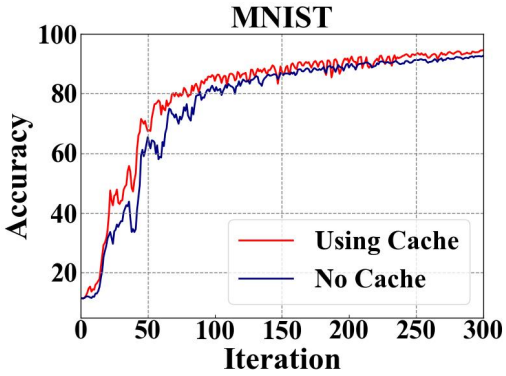



- Non-IID data: Few/biased local samples cause overfitting and cross-client inconsistency; unstable convergence.
- Semi-asynchronous participation: Only earliest K updates are aggregated; many uploads are stale → participation bias and drift.
- Heterogeneous sparsity: Clients use different mask widths; sparse, shape-mismatched updates leave parameters missing and skew aggregation.
- Detector-only pipelines:Identifying attackers doesn't undo historical contamination of the global model.
- Retrain-based recovery:Accurate yet communication-expensive and slow; impractical for online remediation.

# Motivation: Historical Information Forgetting



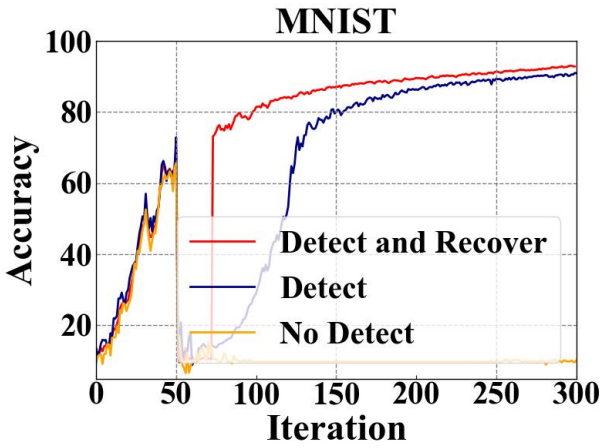
- **Setting:** Semi-asynchronous FL with heterogeneous sparse models and Non-IID clients (N/K≪N, stale updates).Color code: yellow = received global, blue = previous local
- **Observation:** Without cache, aggregation drifts toward currently participating clients → oscillation and slow convergence; with cache, convergence is faster and smoother.
- **limitation:** Earliest-K aggregation ignores consistent but unreceived history; heterogeneous sparsity leaves parameters missing; uniform weighting/zero-fill inflates variance.
- **Need:** Cache-based imputation + staleness-aware exponential weighting + cosine-similarity selection of historical updates to reduce bias/variance and stabilize convergence.



# Motivation: Robust Recovery After Detection

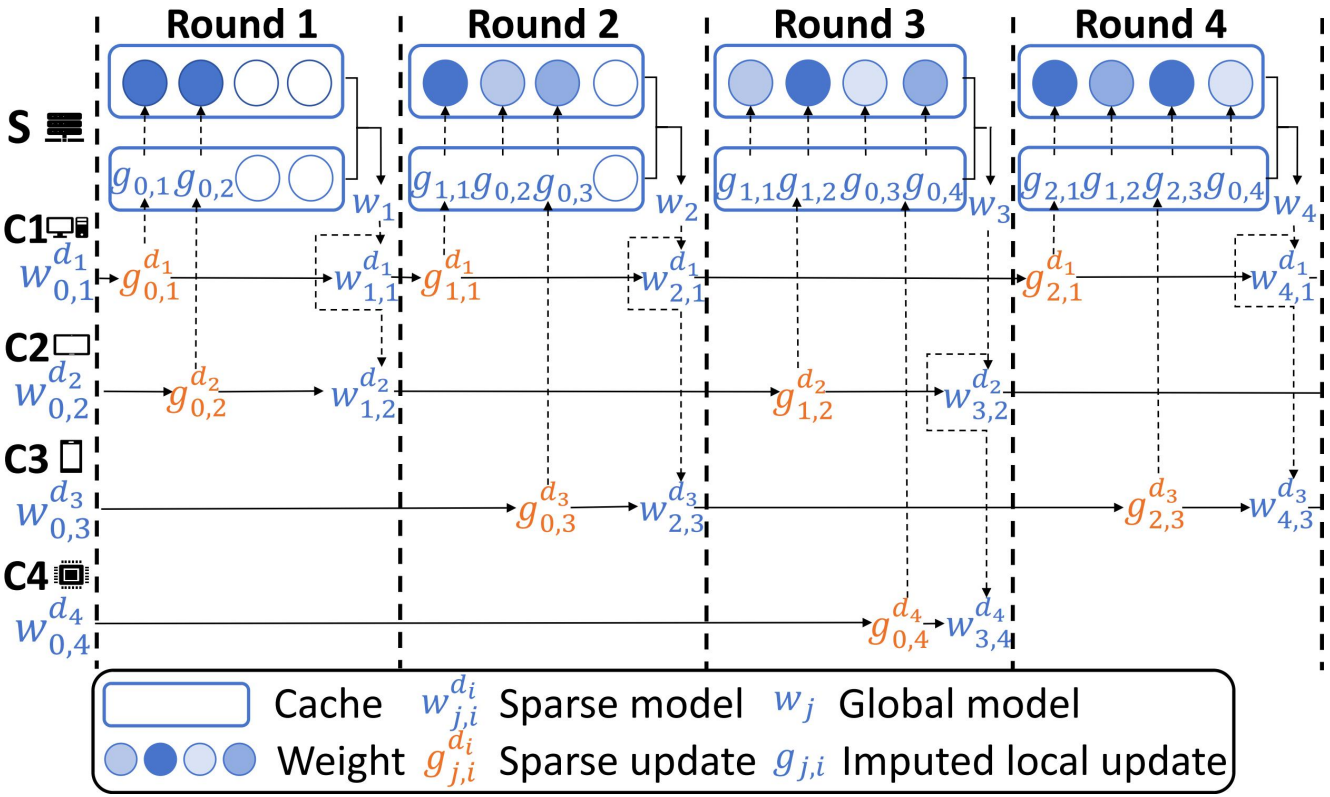


- **Setup:** Poisoning attacks (trim/backdoor/LF) in semi-asynchronous, heterogeneous FL; compare “No Detect” vs “Detect” vs “Detect + Recover”.
- **Observation:** Detect + Recover reaches higher accuracy earlier and is more stable; Detect alone cannot remove historical contamination.
- **Limitation:** Removing attackers does not undo the polluted global state; prior recovery is not integrated with detection and overlooks heterogeneity/staleness.
- **Need:** Triggered recovery using Cauchy mean value theorem + L-BFGS to predict updates; combine precise and estimated updates to lower ASR and rapidly restore accuracy.



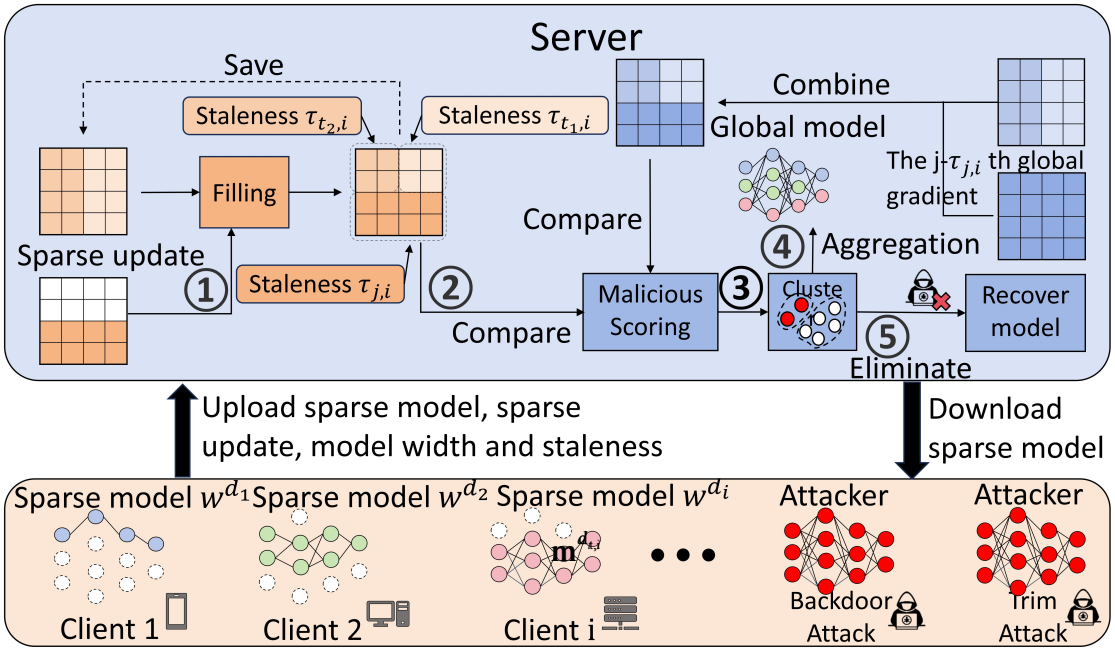
# Model Update






- Semi-asynchronous FL with heterogeneous, mask-based sparse models; each client uploads a sparse update plus width and staleness.
- Server keeps a per-client cache of the latest imputed update and a decaying staleness weight.
- Aggregate the first K arrivals, then augment with cosine-similar cached history from non-arrived clients to reduce bias/variance.
- Result: faster, smoother convergence under Non-IID data and partial participation

# Framework Overview




① **imputed local updates :** $\bar{\boldsymbol{g}}(\boldsymbol{w}_{j,i}) = \boldsymbol{g}(\boldsymbol{w}_{t,i}^{\boldsymbol{d}_{t,i}}, \boldsymbol{\xi}_{t,i}) \cup (\bar{\boldsymbol{g}}(\boldsymbol{w}_{j-1,i}) \odot \widetilde{\boldsymbol{m}}^{\boldsymbol{d}_{t,i}})$

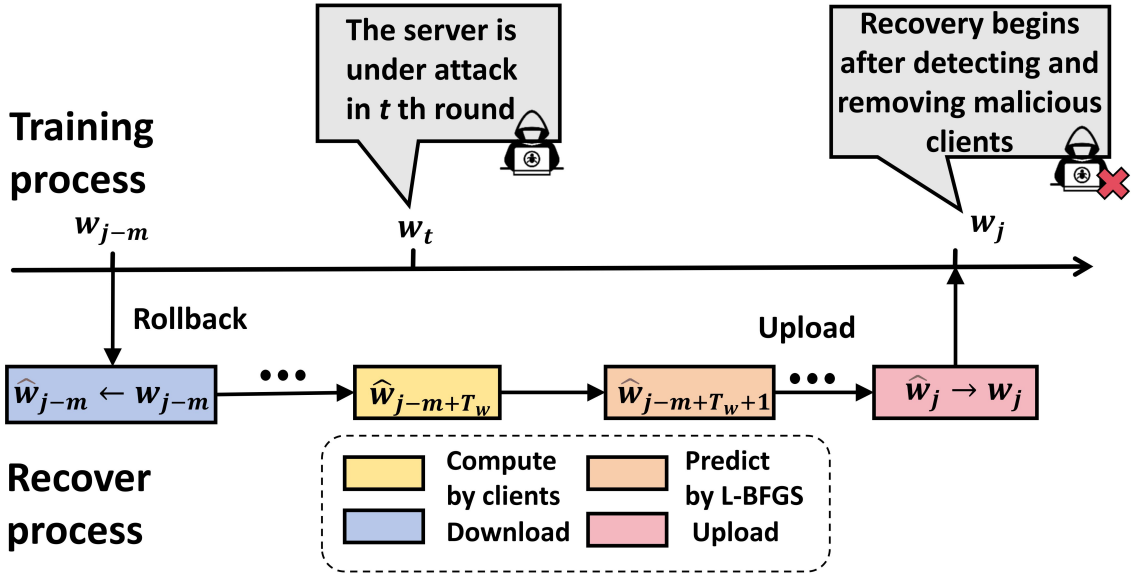
② **weight of the imputed update :** $\boldsymbol{M}_{j,i} = \boldsymbol{\beta}^{\boldsymbol{\tau}_{j,i}} \boldsymbol{m}^{\boldsymbol{d}_{t,i}} \cup (\boldsymbol{\beta} \boldsymbol{M}_{j-1,i} \odot \widetilde{\boldsymbol{m}}^{\boldsymbol{d}_{t,i}})$

③ **final weight** $\boldsymbol{p}'_{j,i} = \boldsymbol{M}_{j,i}(\boldsymbol{m}^{\boldsymbol{d}_{t,i}} \cup \boldsymbol{\alpha} \widetilde{\boldsymbol{m}}^{\boldsymbol{d}_{t,i}})$,

④**aggregation:** $\bar{\boldsymbol{g}}(\boldsymbol{w}_j) = \sum_{i=1}^{K} \boldsymbol{p}_{j,i} \bar{\boldsymbol{g}}(\boldsymbol{w}_{j,i})$

- Complete incoming sparse updates using the cache; track staleness and maintain aligned global views per client.
- Staleness-weighted aggregation of benign updates; select consistent history by cosine similarity.
- Detect attacks via deviation between imputed local and global updates (DBSCAN → k-means).

# Recovery Process




***the latest imputed global update*** : $\check{g}(w_{j,i}) = g(w_{j-T_{j,i}}) \odot m^{d_{t,i}} \cup \check{g}(w_{j-1,i}) \odot \widetilde{m}^{d_{t,i}}$

**suspicious score** : $s_{j,i} = \|g(w_{j,i}) - \bar{g}(w_{j,i})\|_2$

- Detect attackers, then roll back a few rounds to a clean checkpoint.
- Reconstruct missing/heterogeneous sparse updates using L-BFGS guided prediction, mixed with early and periodic precise client computations.
- Upload recovered updates and resume training, yielding lower ASR and faster return to pre-attack accuracy.

# Main Results



| Dataset | Mask level | Dirichlet distribution | N/K=1000/20 | | |
|---|---|---|---|---|---|
| | | | FedHAN | TWAFL | SASGD |
| MNIST | 0.5 | 0.8 | 98.91 | 95.37 | 90.14 |
| | | 0.3 | 97.13 | 93.47 | 90.25 |
| | 0.2 | 0.8 | 97.77 | 94.87 | 89.54 |
| | | 0.3 | 95.19 | 91.05 | 84.25 |
| Fashion MNIST | 0.5 | 0.8 | 82.93 | 78.45 | 65.75 |
| | | 0.3 | 80.41 | 73.31 | 52.05 |
| | 0.2 | 0.8 | 77.44 | 74.67 | 63.74 |
| | | 0.3 | 74.62 | 70.37 | 58.63 |
| CIFAR-10 | 0.5 | 0.8 | 47.31 | 41.42 | 31.51 |
| | | 0.3 | 45.16 | 37.89 | 25.05 |
| | 0.2 | 0.8 | 45.87 | 37.98 | 27.76 |
| | | 0.3 | 43.94 | 34.49 | 21.94 |

- Test accuracy: FedHAN is best and converges smoother in all settings.On MNIST, FMNIST, CIFAR-10 with N/K=1000/20 and mask 0.2/0.5, FedHAN attains the best accuracy and smoother convergence. Example (MNIST, mask 0.5, Dir=0.8): 98.91 vs 95.37 (TWAFL) vs 90.14 (SASGD)

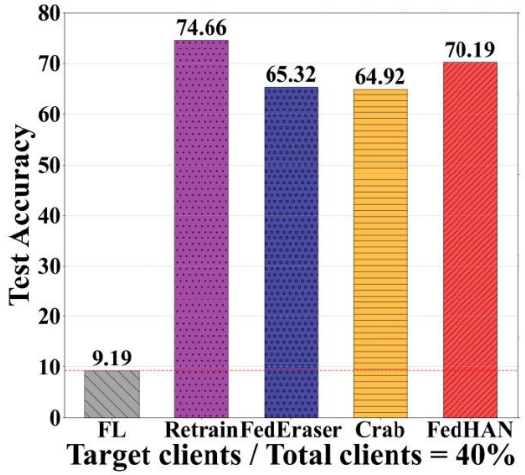
# Detection Results



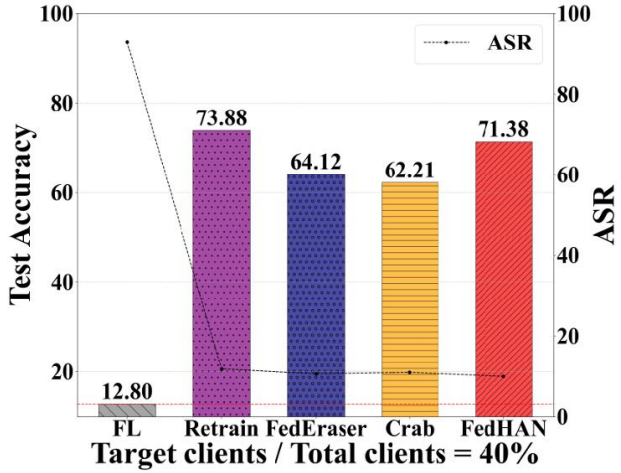
| Dataset | Detector | Trim attack | | | Backdoor attack | | | LF attack | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | DACC | FPR | FNR | DACC | FPR | FNR | DACC | FPR | FNR |
| MNIST | FoolsGold | 0.92 | 0.09 | 0.03 | 0.83 | 0.31 | 0.44 | 0.79 | 0.43 | 0.36 |
| | FLDetector | 0.96 | 0.04 | 0.02 | 0.84 | 0.15 | 0.11 | 0.81 | 0.32 | 0.11 |
| | FedHAN | **1.00** | **0.00** | **0.00** | **0.91** | **0.15** | **0.05** | **0.90** | **0.12** | **0.07** |
| Fashion-MNIST | FoolsGold | 0.91 | 0.08 | 0.04 | 0.77 | 0.67 | 0.15 | 0.72 | 0.64 | 0.22 |
| | FLDetector | 0.95 | 0.05 | 0.02 | 0.80 | 0.41 | 0.10 | 0.81 | 0.30 | 0.11 |
| | FedHAN | **1.00** | **0.00** | **0.00** | **0.90** | **0.14** | **0.05** | **0.87** | **0.21** | **0.13** |
| CIFAR-10 | FoolsGold | 0.93 | 0.09 | 0.02 | 0.63 | 0.83 | 0.32 | 0.65 | 0.81 | 0.33 |
| | FLDetector | 0.95 | 0.05 | 0.02 | 0.75 | 0.78 | 0.17 | 0.75 | 0.71 | 0.15 |
| | FedHAN | **1.00** | **0.00** | **0.00** | **0.87** | **0.12** | **0.08** | **0.82** | **0.31** | **0.14** |

- Under trim/backdoor/LF, FedHAN achieves high DACC with low FPR/FNR. Trim: 1.00 on all three datasets; Backdoor: 0.91/0.90/0.87 (MNIST/FMNSIT/CIFAR-10); LF: 0.90/0.87/0.82.

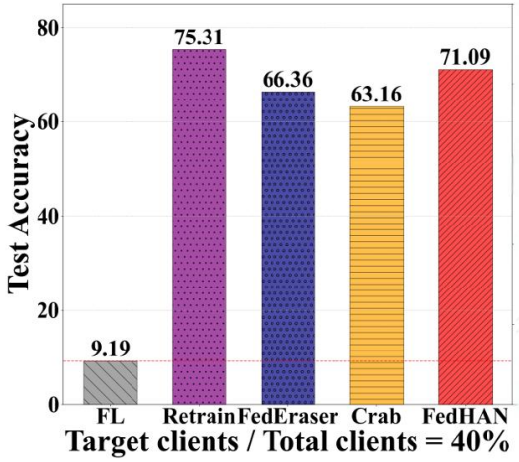
# Recovery Results




(a) Trim attack




(b) Backdoor attack




(c) LF attack

- Trim: FedHAN 70.2% TACC, close to Retrain 74.7%, higher than FedEraser 65.3% and Crab 64.9%; low ASR.
- Backdoor: FedHAN 71.4% vs Retrain 73.9%, above FedEraser/Crab; ASR drops to near zero.
- Label flipping: FedHAN 71.1% vs Retrain 75.3%, > FedEraser/Crab; low ASR.
- Takeaway: FedHAN approaches retraining accuracy with far lower cost and faster recovery across attacks.

## 4.2 Convergence Analysis

**Theorem 1.** *Assume Assumptions 1, 2, 3, 4 hold. Learning rate satisfies that* $\eta \leq \frac{1}{Ls_j}$ *and subjects to* $\mathcal{M}_j \geq 0$*, where*

$$\mathcal{M}_j = (\frac{\eta_j s_j}{2} - \sum_{l=j}^{J} \eta_l \sum_{t=j}^{l-1} \eta_t^2 \sum_{k=1}^{t} \alpha^{t-j} I_{l,k,t}).$$

*Then, we can obtain the following convergence result*

$$\frac{1}{J}\sum_{j=1}^{J} \mathcal{M}_j ||\nabla F(w_j)||_2^2 \leq \frac{1}{J}\sum_{j=1}^{J}(\frac{3\eta_j s_j}{2} C + \eta_j \sigma_c^2 \sum_{l=1}^{j} \sum_{t=l}^{j-1} \eta_t^2 s_t I_{l,k,t}) + \frac{F(w_1) - F(w^*)}{J},$$

*where* $C = G^2 + \sigma_e^2 \sum_{i=1}^{K} \mathbb{E}[p_{l,i}]$ *and* $I_{l,k,t} = 3L^2 B^2 s_t \alpha^{l-k}(l - \tau_k)$.

## 4.3 Theoretical Analysis on Suspicious Scores

**Theorem 2.** *Suppose that the update of each client's loss function is L-smooth, FedHAN is used as the aggregation rule, the learning rate* $\alpha$ *satisfies* $\alpha < \frac{1}{(N+2)L}$ *(*$N$ *is the window size). Suppose that malicious clients perform an untargeted model poisoning attack in each iteration by reversing the true model updates as the poisoning ones, that is, each malicious client* $i$ *sends* $-g(w_{j,i})$ *to the server in each iteration* $t$*. Then we find that the expected suspicious score of a benign client is smaller than that of a malicious client in each iteration* $t$*. Formally, we have the following inequality.*

$$E(s_i^t) < E(s_a^t), \forall i, \tag{10}$$

*where the expectation* $E$ *is taken with respect to the randomness in the clients' local training data,* $\mathcal{B}$ *is the set of benign clients, and* $M$ *is the set of malicious clients.*

# Conclusion:

- Cache-based semi-asynchronous FL with heterogeneous sparse models.
- Staleness-aware weighting and similarity-guided history stabilize training under Non-IID.
- Deviation-based detection + triggered recovery (Cauchy + L-BFGS) remove historical contamination.
- Results: higher accuracy, strong detection with low errors, fast recovery; robust under heterogeneity and staleness.