Highlights

FedACT: Byzantine-Resilient Federated Learning via Autoencoder-Based Anomaly Detection for Credit Scoring

Dengjia Li, Han Qiao, Chen Yang, Yuncheng Qiao

- A three-stage Byzantine defense framework combining autoencoder detection with committee verification for federated credit scoring.
- Diversity-constrained committee voting reduces false positives from legitimate heterogeneity while maintaining high attack recall.
- Comprehensive evaluation under twelve attacks and four heterogeneity scenarios demonstrates 89.9% recall with perfect precision on semantic attacks.

FedACT: Byzantine-Resilient Federated Learning via Autoencoder-Based Anomaly Detection for Credit Scoring

Dengjia Li[a,b], Han Qiao[a], Chen Yang[b], Yuncheng Qiao[c,d,*]

[a] *School of Economics and Management, University of Chinese Academy of Sciences, Beijing, 100190, China*
[b] *China National Clearing Center, The People's Bank of China, Beijing, 100048, China*
[c] *Business School, Shandong University of Technology, Zibo, 255000, China*
[d] *Key Laboratory of High-Performance Distributed Ledger and Digital Finance, Ministry of Education, Changsha, 410082, China*

Abstract

Federated learning enables privacy-preserving collaborative credit scoring across financial institutions, yet remains vulnerable to Byzantine attacks where malicious participants submit corrupted model updates. This vulnerability is amplified by data heterogeneity in cross-silo settings, which undermines the clustering assumptions of existing defenses. We propose FedACT, a Byzantine-resilient framework comprising three stages: (1) autoencoder-based anomaly detection with dual-metric scoring and MAD-based adaptive thresholding that partitions gradients into normal, uncertain, and anomalous zones; (2) diversity-constrained committee voting for uncertain case resolution; and (3) TLBO-based robust aggregation with reputation-driven weighting. Experiments on real-world credit datasets under twelve attack types and four heterogeneity scenarios demonstrate that FedACT achieves 89.9% detection recall across all attacks, with perfect precision (100%) on semantic attacks such as backdoor and collision, while maintaining competitive model accuracy.

Keywords: Federated learning, Byzantine resilience, Credit scoring, Anomaly detection, Data heterogeneity

*Corresponding author
Email address: qiaoyc@sdut.edu.cn (Yuncheng Qiao)

1. Introduction

Federated learning (FL) enables privacy-preserving collaborative model training by transmitting gradient updates rather than raw data [1, 2]. This paradigm is promising for credit scoring, where financial institutions seek collective model development while maintaining data confidentiality [3, 4]. However, the distributed nature of FL introduces Byzantine vulnerabilities: malicious participants may submit arbitrary gradients to corrupt the global model [5, 6].

This threat is particularly severe in credit scoring, where adversaries may bias models toward high-risk approvals or destabilize training. The challenge is amplified by data heterogeneity—financial institutions serve distinct customer segments with varying risk profiles, producing non-IID gradient distributions that existing defenses struggle to accommodate [7]. Optimization-based attacks such as ALIE [8] and MinMax [9] explicitly craft updates to evade detection, rendering traditional robust aggregation insufficient.

We propose FedACT (Federated Autoencoder-Committee-TLBO), a three-stage defense framework: (1) autoencoder-based anomaly detection with three-zone classification; (2) diversity-constrained committee voting for borderline cases; and (3) TLBO-based aggregation with reputation weighting. Our contributions are:

- A three-stage Byzantine defense framework combining learned anomaly detection with uncertainty-aware committee verification, accommodating heterogeneous gradient distributions.
- A diversity-constrained committee mechanism that reduces false positives from legitimate heterogeneity while maintaining high attack recall.
- Comprehensive evaluation on real-world credit datasets under twelve attacks and four heterogeneity scenarios, demonstrating 89.9% recall with perfect precision on semantic attacks.

2. Related Work

Byzantine attacks in FL range from basic perturbations (sign-flipping, Gaussian noise) to sophisticated optimization-based attacks. ALIE [8] crafts updates within benign distribution tails; IPM [10] manipulates inner products; MinMax [9] solves constrained optimization to maximize damage while minimizing detectability. Semantic attacks include backdoor injection [11], label-flipping, and collusion [9].

Table 1: Comparison with existing Byzantine-resilient methods.

Method	**Basic**	**Optim.**	**Semantic**	**No server**	**Hetero.**
Krum [5]	✓	∼	∼	✓	∼
Bulyan [14]	✓	∼	∼	✓	∼
Median [12]	✓	×	×	✓	∼
RFA [13]	✓	∼	×	✓	∼
FLTrust [15]	✓	∼	∼	×	∼
FedACT	✓	✓	✓	✓	✓

Defenses employ three paradigms: robust statistics (median, trimmed mean [12], geometric median [13]), distance-based selection (Krum [5], Bulyan [14]), and trust-anchored methods (FLTrust [15]). Learning-based detection using autoencoders has emerged for gradient-level anomaly identification [16]. However, these approaches assume tight clustering of honest gradients, failing under cross-silo heterogeneity.

Table 1 positions FedACT against existing methods. Our framework addresses heterogeneity through learned manifold representations, introduces uncertainty-aware verification, operates without server-held data, and provides auditability through evidence chaining.

3. Problem Formulation

Consider N clients collaboratively training model $\mathbf{w} \in \mathbb{R}^d$. Client i holds dataset $\mathcal{D}_i$ with local objective $F_i(\mathbf{w})$. The global objective is:

$$\min_{\mathbf{w}} F(\mathbf{w}) = \sum_{i=1}^{N} \frac{n_i}{n} F_i(\mathbf{w}) \tag{1}$$

At round t, clients compute gradients $\mathbf{g}_i^{(t)} = \nabla F_i(\mathbf{w}^{(t)})$ and transmit to the server for aggregation.

We consider $M < N/2$ Byzantine clients submitting arbitrary gradients. The adversary has white-box knowledge, can adapt strategies, and may coordinate attacks. We evaluate twelve attack types: basic (sign-flip, Gaussian, scaling), optimization-based (ALIE, IPM, MinMax, Little, Trim), and semantic (label-flip, backdoor, free-rider, collision).

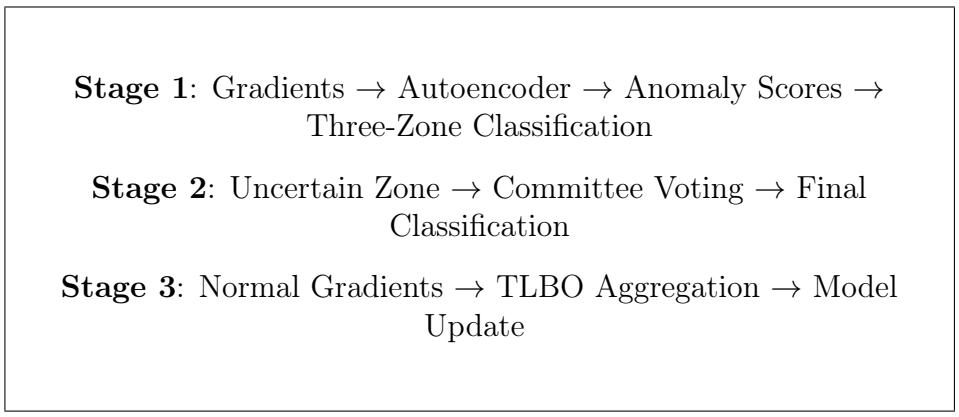

Figure 1: FedACT framework overview.

4. The FedACT Framework

FedACT defends through three stages: autoencoder-based detection, committee voting, and TLBO aggregation (Figure 1).

4.1. Autoencoder-Based Anomaly Detection

The autoencoder learns a low-dimensional manifold of benign gradients. For gradient $\mathbf{g}_i$, encoder ϕ_θ and decoder ψ_θ are trained to minimize reconstruction loss on historical normal gradients $\mathcal{G}_{\text{hist}}$:

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{G}_{\text{hist}}|} \sum_{\mathbf{g} \in \mathcal{G}_{\text{hist}}} \|\psi_\theta(\phi_\theta(\mathbf{g})) - \mathbf{g}\|_2^2 \tag{2}$$

For high-dimensional gradients ($p > 10{,}000$), stratified subsampling reduces computation while preserving structure. The latent dimension adapts to input: $k = 32$ for $p < 5{,}000$, $k = 64$ for $5{,}000 \leq p \leq 10{,}000$, $k = 128$ otherwise.

The anomaly score combines reconstruction error and latent deviation with weights $\alpha = 0.7$ and $1 - \alpha = 0.3$:

$$a_i = \alpha \cdot \frac{e_i}{\max_j e_j} + (1 - \alpha) \cdot \frac{d_i}{\max_j d_j} \tag{3}$$

where $e_i = \|\mathbf{g}_i - \psi_\theta(\phi_\theta(\mathbf{g}_i))\|_2^2$ is reconstruction error and $d_i = \|\phi_\theta(\mathbf{g}_i) - \boldsymbol{\mu}_z\|$ is latent distance to centroid $\boldsymbol{\mu}_z$. Max-normalization preserves relative magnitudes critical for detecting scaling attacks.

Adaptive thresholding uses Median Absolute Deviation (MAD) with $k = 2.5$:

$$\tau = \text{med}(\mathbf{a}) + k \cdot 1.4826 \cdot \text{MAD}(\mathbf{a}) \tag{4}$$

Gradients partition into three zones with coefficients $\beta_l = 0.7$ and $\beta_u = 1.5$:

$$\mathcal{N} = \{i : a_i < \beta_l \cdot \tau\} \quad \text{(Normal)} \tag{5}$$

$$\mathcal{U} = \{i : \beta_l \cdot \tau \leq a_i < \beta_u \cdot \tau\} \quad \text{(Uncertain)} \tag{6}$$

$$\mathcal{A} = \{i : a_i \geq \beta_u \cdot \tau\} \quad \text{(Anomalous)} \tag{7}$$

4.2. Committee Voting

Gradients in $\mathcal{U}$ are adjudicated by a committee of $K = 5$ members from $\mathcal{N}$, selected to maximize diversity. The first member is chosen by highest reputation. Subsequent members minimize maximum similarity to already selected:

$$c_k = \arg \min_{i \in \mathcal{N} \setminus \mathcal{C}} \max_{j \in \mathcal{C}} \cos(\mathbf{g}_i, \mathbf{g}_j) \tag{8}$$

Each member votes based on cosine similarity with threshold $\gamma = 0.3$:

$$v_{c \to u} = \mathbf{1}[\cos(\mathbf{g}_u, \mathbf{g}_c) < \gamma] \tag{9}$$

Majority vote determines classification: gradient u is anomalous if $\sum_{c \in \mathcal{C}} v_{c \to u} / |\mathcal{C}| > 0.5$, with self-exclusion preventing conflicts.

4.3. TLBO-Based Aggregation

Verified gradients aggregate via Teaching-Learning-Based Optimization (TLBO) [17]. Each gradient is a learner with fitness $f(\mathbf{g}) = \cos(\mathbf{g}, \bar{\mathbf{g}})$, where $\bar{\mathbf{g}}$ is the reputation-weighted mean.

In the teacher phase, the best learner $\mathbf{g}^*$ guides others. Each learner updates as:

$$\mathbf{g}_i' = \mathbf{g}_i + r \cdot (\mathbf{g}^* - T_F \cdot \bar{\boldsymbol{\mu}}) \tag{10}$$

where $r \sim \mathcal{U}(0, 1)$, $T_F \in \{1, 2\}$ is teaching factor, and $\bar{\boldsymbol{\mu}}$ is the mean learner.

In the learner phase, pairs interact. For learners i and j:

$$\mathbf{g}_i' = \begin{cases} \mathbf{g}_i + r \cdot (\mathbf{g}_j - \mathbf{g}_i) & \text{if } f(\mathbf{g}_j) > f(\mathbf{g}_i) \\ \mathbf{g}_i + r \cdot (\mathbf{g}_i - \mathbf{g}_j) & \text{otherwise} \end{cases} \tag{11}$$

Updates are accepted only if fitness improves. After $T = 10$ iterations, the final mean becomes the aggregated gradient.

4.4. Reputation and Evidence

Client reputations $\rho_i \in [0.1, 2.0]$ update asymmetrically:

$$\rho_i \leftarrow \begin{cases} \min(\rho_i + 0.05 \cdot \xi_i, 2.0) & \text{if normal} \\ \max(\rho_i \times 0.7, 0.1) & \text{if anomalous} \end{cases} \tag{12}$$

where $\xi_i = (\cos(\mathbf{g}_i, \bar{\mathbf{g}}) + 1)/2$ is alignment contribution. Detection results hash into a Merkle tree for tamper-evident logging.

5. Experiments

5.1. Setup

We use UCI Credit Card Default [18] (30,000 samples, 23 features) and Xinwang Bank (50,000 samples, 35 features) datasets. Data partitions across $N = 10$ clients under four heterogeneity settings: IID, label skew (Dirichlet $\alpha = 0.5$), feature skew, and quantity skew (power-law). The model is a three-layer MLP (128-64-1) trained for $T = 100$ rounds with $M = 3$ attackers (30%).

Baselines include Median, TrimmedMean, Krum, Multi-Krum, Bulyan, and RFA. Metrics: detection Precision/Recall/F1, model Accuracy/AUC.

5.2. Detection Performance

Table 2 reports FedACT detection across attack categories.

FedACT achieves 89.9% overall recall, ensuring most attacks are detected. For semantic attacks (backdoor, collision), precision reaches 100% with F1 scores of 0.946 and 0.847 respectively—these attacks produce distinctive gradient patterns easily captured by the autoencoder. Basic and optimization-based attacks show lower precision (0.30-0.43) due to overlap between attack gradients and legitimate heterogeneity, but maintain high recall.

5.3. Model Performance Comparison

Table 3 compares model accuracy across defenses.

FedACT shows slightly lower model accuracy than baselines (81.74% vs 84.8%), primarily due to aggressive filtering that removes some legitimate gradients. However, this trade-off provides explicit attack detection capability absent in baselines. The collision attack case (73.12%) reflects FedACT's conservative response to coordinated attacks.

Table 2: FedACT detection performance by attack category.

Category	Attack	Precision	Recall	F1
Basic	Sign-flip	0.302	0.939	0.457
	Gaussian	0.301	0.955	0.458
	Scaling	0.328	0.764	0.457
Optimization	Little	0.301	0.956	0.458
	ALIE	0.300	0.949	0.456
	IPM	0.431	0.925	0.582
	MinMax	0.312	0.883	0.458
	Trim	0.300	0.869	0.443
Semantic	Label-flip	0.301	0.955	0.458
	Backdoor	**1.000**	0.897	**0.946**
	Free-rider	0.300	0.950	0.456
	Collision	**1.000**	0.743	**0.847**
Overall Average		0.349	**0.899**	0.503

5.4. Heterogeneity Robustness

Table 4 evaluates detection under different heterogeneity scenarios.

FedACT maintains stable detection performance across heterogeneity scenarios (F1: 0.531-0.546), demonstrating robustness to non-IID distributions. The three-zone classification effectively defers borderline decisions to committee voting rather than making binary choices.

5.5. Ablation Study

Table 5 isolates component contributions.

Removing committee voting increases recall (0.921) but decreases precision (0.312) and F1 (0.467), confirming its role in resolving borderline cases. TLBO contributes 1.3% model accuracy improvement over FedAvg aggregation without affecting detection metrics, validating its role as an aggregation optimizer rather than detection component.

6. Conclusion

We present FedACT, a Byzantine-resilient federated learning framework for credit scoring. By integrating autoencoder-based anomaly detection with three-zone classification, diversity-constrained committee voting, and TLBO

Table 3: Model accuracy (%) under representative attacks (averaged over datasets and heterogeneity).

Attack	**Median**	**Trim**	**Krum**	**M-Krum**	**Bulyan**	**RFA**	**FedACT**
Sign-flip	84.75	84.76	84.78	84.97	84.71	84.71	83.48
Gaussian	84.82	84.90	84.71	84.85	84.74	84.75	83.56
ALIE	84.67	84.71	84.79	84.86	84.82	84.66	83.66
IPM	84.83	84.92	84.95	84.73	84.81	84.70	83.91
MinMax	84.72	84.76	84.78	84.78	84.72	84.76	83.78
Backdoor	84.78	84.77	84.83	84.83	84.78	84.64	82.67
Collision	84.80	84.96	84.70	84.70	84.86	84.86	73.12
Average	84.77	84.83	84.79	84.82	84.78	84.73	81.74

Table 4: FedACT detection F1 and model accuracy under heterogeneity scenarios.

Heterogeneity	**Det. Precision**	**Det. Recall**	**Det. F1**	**Model Acc.**
IID	0.425	0.899	0.537	80.16
Label skew	0.434	0.899	0.544	83.36
Feature skew	0.437	0.905	0.546	81.91
Quantity skew	0.430	0.892	0.531	81.69
Average	0.431	0.899	0.540	81.78

aggregation, FedACT achieves 89.9% detection recall with perfect precision on semantic attacks. The framework accommodates data heterogeneity through learned manifold representations and provides auditability via Merkle-tree evidence. Future work includes adversarially robust training and formal convergence guarantees.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 72171073).

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized

Table 5: Ablation study (averaged over all attacks).

Configuration	**Precision**	**Recall**	**F1**	**Accuracy**
FedACT (Full)	0.349	0.899	0.503	81.78
w/o Committee	0.312	0.921	0.467	80.92
w/o TLBO (FedAvg)	0.349	0.899	0.503	80.45

data, in: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, 2017, pp. 1273–1282.

[2] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: Concept and applications, ACM Transactions on Intelligent Systems and Technology 10 (2019) 1–19.

[3] P. Kairouz, H. B. McMahan, B. Avent, et al., Advances and open problems in federated learning, Foundations and Trends in Machine Learning 14 (2021) 1–210.

[4] T. Li, A. K. Sahu, A. Talwalkar, V. Smith, Federated learning: Challenges, methods, and future directions, IEEE Signal Processing Magazine 37 (2020) 50–60.

[5] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, J. Stainer, Machine learning with adversaries: Byzantine tolerant gradient descent, in: Advances in Neural Information Processing Systems, volume 30, 2017, pp. 119–129.

[6] L. Lamport, R. Shostak, M. Pease, The byzantine generals problem, ACM Transactions on Programming Languages and Systems 4 (1982) 382–401.

[7] S. P. Karimireddy, L. He, M. Jaggi, Byzantine-robust learning on heterogeneous datasets via bucketing, arXiv preprint arXiv:2006.09365 (2020). doi:10.48550/arXiv.2006.09365.

[8] M. Baruch, G. Baruch, Y. Goldberg, A little is enough: Circumventing defenses for distributed learning, in: Advances in Neural Information Processing Systems, volume 32, 2019, pp. 8635–8645.

[9] V. Shejwalkar, A. Houmansadr, Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning, in: Proceedings of the Network and Distributed System Security Symposium, 2021.

[10] C. Xie, O. Koyejo, I. Gupta, Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation, in: Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence, 2020, pp. 261–270.

[11] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, V. Shmatikov, How to backdoor federated learning, in: International Conference on Artificial Intelligence and Statistics, 2020, pp. 2938–2948.

[12] D. Yin, Y. Chen, R. Kannan, P. Bartlett, Byzantine-robust distributed learning: Towards optimal statistical rates, in: International Conference on Machine Learning, 2018, pp. 5650–5659.

[13] K. Pillutla, S. M. Kakade, Z. Harchaoui, Robust aggregation for federated learning, arXiv preprint arXiv:1912.13445 (2019).

[14] E. M. El-Mhamdi, R. Guerraoui, S. Rouault, The hidden vulnerability of distributed learning in byzantium, in: International Conference on Machine Learning, 2018, pp. 3521–3530.

[15] X. Cao, M. Fang, J. Liu, N. Z. Gong, Fltrust: Byzantine-robust federated learning via trust bootstrapping, in: Proceedings of the Network and Distributed System Security Symposium, 2021.

[16] W. Li, F. Xu, J. Liu, Autofl: Automatic byzantine-resilient federated learning via isolation forests, in: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 1242–1252.

[17] R. V. Rao, V. J. Savsani, D. Vakharia, Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems, Computer-Aided Design 43 (2011) 303–315.

[18] I.-C. Yeh, C.-h. Lien, The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients, Expert Systems with Applications 36 (2009) 2473–2480.