# 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 particularly promising for credit scoring, where financial institutions seek to develop collective models while maintaining strict data confidentiality under regulations such as GDPR [3] and emerging data protection laws [4, 5]. However, the distributed nature of FL introduces Byzantine vulnerabilities: malicious participants may submit arbitrary or carefully crafted gradients to corrupt the global model [6, 7].

This threat is especially severe in credit scoring applications, where adversaries may bias models toward high-risk approvals, destabilize training convergence, or inject backdoors that activate for specific applicant profiles. The challenge is further amplified by data heterogeneity—financial institutions serve distinct customer segments with varying risk profiles, demographic distributions, and default rates, producing non-IID gradient distributions that existing defenses struggle to accommodate [8, 9]. Recent optimization-based attacks such as ALIE [10], IPM [11], and MinMax [12] explicitly craft updates to evade detection by mimicking benign gradient statistics, rendering traditional robust aggregation methods insufficient.

We propose FedACT (**Fed**erated **A**utoencoder-**C**ommittee-**T**LBO), a three-stage Byzantine-resilient framework designed specifically for heterogeneous federated credit scoring:

1. **Autoencoder-based anomaly detection**: A variational autoencoder learns the manifold of benign gradients, computing dual-metric anomaly scores (reconstruction error + latent deviation) with MAD-based adaptive thresholding to partition gradients into normal, uncertain, and anomalous zones.
2. **Diversity-constrained committee voting**: Borderline cases in the uncertain zone are adjudicated by a committee of diverse normal clients, reducing false positives from legitimate heterogeneity while maintaining high attack recall.
3. **TLBO-based robust aggregation**: Verified gradients are aggregated using Teaching-Learning-Based Optimization with reputation-weighted fitness, iteratively refining the aggregated update.

Our contributions are summarized as follows:

- We propose a three-stage Byzantine defense framework that combines learned anomaly detection with uncertainty-aware committee verification,

specifically designed to accommodate heterogeneous gradient distributions in cross-silo federated learning.

- We introduce a diversity-constrained committee mechanism that selects maximally dissimilar normal clients as voters, reducing false positives from legitimate data heterogeneity while maintaining high attack detection recall.
- We conduct comprehensive experiments on real-world credit scoring datasets under twelve attack types (basic, optimization-based, and semantic) and four heterogeneity scenarios (IID, label skew, feature skew, quantity skew), demonstrating that FedACT achieves 89.9% overall detection recall with perfect precision on semantic attacks.

## 2. Related Work

### *2.1. Byzantine Attacks in Federated Learning*

Byzantine attacks in federated learning range from basic perturbations to sophisticated optimization-based strategies [13, 14]. Basic attacks include sign-flipping (negating gradient directions), Gaussian noise injection, and scaling attacks that amplify gradient magnitudes [6]. While easily detectable in isolation, these attacks can be effective when combined with adaptive strategies.

More sophisticated optimization-based attacks explicitly evade detection. ALIE (*A Little Is Enough*) [10] crafts malicious updates within the statistical tails of benign gradient distributions. IPM (*Inner Product Manipulation*) [11] manipulates inner products to circumvent distance-based defenses. MinMax [12] solves a constrained optimization problem to maximize model corruption while minimizing detectability. Trim attack [15] targets trimmed mean defenses specifically.

Semantic attacks pose unique challenges as they may produce seemingly normal gradients while achieving malicious objectives. Backdoor attacks [16, 17] inject hidden triggers that cause misclassification on specific inputs. Label-flipping corrupts training labels to degrade model performance. Free-rider attacks [18] submit minimal or copied updates to exploit model improvements without contribution. Collusion attacks coordinate multiple malicious clients to amplify attack effectiveness [12].

### 2.2. Byzantine-Resilient Aggregation

Existing defenses employ three main paradigms [9]. *Robust statistics* methods replace vulnerable averaging with robust estimators: coordinate-wise median [19], trimmed mean that discards extreme values, and geometric median (RFA) [20] that minimizes sum of distances. These methods assume honest gradients cluster tightly, which fails under heterogeneity.

*Distance-based selection* methods identify outliers through pairwise distances. Krum [6] selects the gradient with minimum sum of distances to nearest neighbors. Multi-Krum extends this to select multiple gradients. Bulyan [21] combines Krum selection with trimmed mean. However, optimization-based attacks can craft updates that appear close to benign gradients.

*Trust-anchored methods* require additional information. FLTrust [22] uses a server-held clean dataset to compute trust scores, which may be unavailable in privacy-sensitive domains. Recent learning-based approaches use autoencoders [23] or isolation forests [24] for gradient-level anomaly detection, but typically make binary decisions without handling borderline cases.

### 2.3. Federated Credit Scoring

Federated learning has gained traction in credit scoring applications where data sharing faces regulatory and competitive barriers [25]. Yang et al. [26] propose an explainable federated learning method combined with blockchain for secure credit modeling. He et al. [27] develop a privacy-preserving decentralized approach using multi-party computation. However, existing work largely assumes honest participation, leaving Byzantine resilience underexplored.

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

## 3. Problem Formulation

### 3.1. Federated Learning Setup

Consider $N$ financial institutions (clients) collaboratively training a credit scoring model $\mathbf{w} \in \mathbb{R}^d$. Each client $i$ holds a private dataset $\mathcal{D}_i = \{(\mathbf{x}_j, y_j)\}_{j=1}^{n_i}$

Table 1: Comparison with existing Byzantine-resilient methods. ✓: effective, ∼: partial, ×: ineffective.

| **Method** | **Basic** | **Optim.** | **Semantic** | **No Server Data** | **Hetero.** |
|---|---|---|---|---|---|
| Krum [6] | ✓ | ∼ | ∼ | ✓ | ∼ |
| Bulyan [21] | ✓ | ∼ | ∼ | ✓ | ∼ |
| Median [19] | ✓ | × | × | ✓ | ∼ |
| RFA [20] | ✓ | ∼ | × | ✓ | ∼ |
| FLTrust [22] | ✓ | ∼ | ∼ | × | ∼ |
| AutoFL [24] | ✓ | ✓ | ∼ | ✓ | ∼ |
| **FedACT (Ours)** | ✓ | ✓ | ✓ | ✓ | ✓ |

where $\mathbf{x}_j$ represents applicant features and $y_j \in \{0, 1\}$ indicates default status. The local objective for client $i$ is:

$$F_i(\mathbf{w}) = \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(\mathbf{w}; \mathbf{x}_j, y_j) \tag{1}$$

where $\ell(\cdot)$ is the binary cross-entropy loss. The global objective minimizes the weighted average:

$$\min_{\mathbf{w}} F(\mathbf{w}) = \sum_{i=1}^{N} \frac{n_i}{n} F_i(\mathbf{w}), \quad n = \sum_{i=1}^{N} n_i \tag{2}$$

At each communication round $t$, clients receive the global model $\mathbf{w}^{(t)}$, perform local training to obtain gradients $\mathbf{g}_i^{(t)} = \nabla F_i(\mathbf{w}^{(t)})$, and transmit updates to the central server for aggregation.

### *3.2. Threat Model*

We consider $M < N/2$ Byzantine clients controlled by an adversary with the following capabilities:

- **White-box knowledge**: The adversary knows the defense mechanism, model architecture, and training algorithm.
- **Adaptive strategy**: Attack methods can adapt across rounds based on observed model behavior.
- **Coordination**: Malicious clients may collude to amplify attack effectiveness.

- **Arbitrary updates**: Byzantine clients can submit any gradient value, not constrained by their local data.

We evaluate robustness against twelve attack types organized into three categories:

1. **Basic attacks** (3 types): Sign-flip, Gaussian noise, Scaling
2. **Optimization-based attacks** (5 types): Little, ALIE, IPM, MinMax, Trim
3. **Semantic attacks** (4 types): Label-flip, Backdoor, Free-rider, Collision

### *3.3. Data Heterogeneity*

Cross-silo federated learning exhibits significant data heterogeneity as institutions serve different customer segments. We consider four heterogeneity scenarios:

- **IID**: Data uniformly distributed across clients (baseline).
- **Label skew**: Default rates vary across clients following Dirichlet distribution with $\alpha = 0.5$.
- **Feature skew**: Feature distributions differ due to regional/demographic variations.
- **Quantity skew**: Dataset sizes follow power-law distribution, modeling institution size disparities.

## 4. The FedACT Framework

FedACT defends against Byzantine attacks through three integrated stages: autoencoder-based anomaly detection with three-zone classification, diversity-constrained committee voting for borderline resolution, and TLBO-based robust aggregation. Figure 1 illustrates the overall architecture.

### *4.1. Stage 1: Autoencoder-Based Anomaly Detection*

The first stage learns a low-dimensional manifold representation of benign gradients using a variational autoencoder, then computes anomaly scores to partition gradients into three zones.

#### *4.1.1. Gradient Preprocessing*

For high-dimensional gradient vectors ($d > 10{,}000$), we apply stratified subsampling to reduce computational cost while preserving structural information. The gradient is partitioned into $B$ blocks, and representative

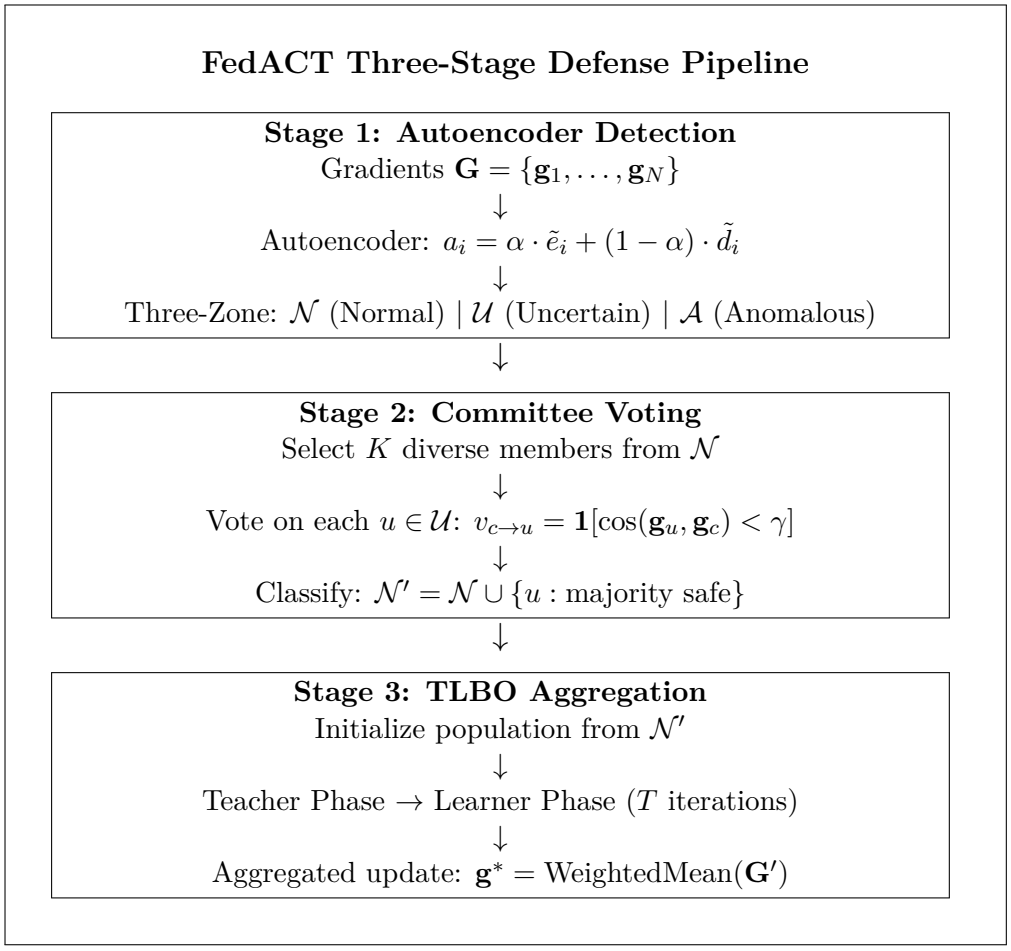


Figure 1: FedACT framework architecture showing the three-stage defense pipeline.

dimensions are sampled from each block proportionally to variance. The subsampled dimension adapts to input size:

$$d' = \begin{cases} d & \text{if } d \leq 5{,}000 \\ 5{,}000 & \text{if } d > 5{,}000 \end{cases} \tag{3}$$

*4.1.2. Autoencoder Architecture*

The encoder $\phi_\theta : \mathbb{R}^{d'} \to \mathbb{R}^k$ and decoder $\psi_\theta : \mathbb{R}^k \to \mathbb{R}^{d'}$ are trained on historical normal gradients $\mathcal{G}_{\text{hist}}$ to minimize reconstruction loss:

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{G}_{\text{hist}}|} \sum_{\mathbf{g} \in \mathcal{G}_{\text{hist}}} \|\psi_\theta(\phi_\theta(\mathbf{g})) - \mathbf{g}\|_2^2 \tag{4}$$

The latent dimension $k$ adapts to gradient dimensionality:

$$k = \begin{cases} 32 & \text{if } d' < 5{,}000 \\ 64 & \text{if } 5{,}000 \leq d' \leq 10{,}000 \\ 128 & \text{otherwise} \end{cases} \tag{5}$$

Both encoder and decoder use two hidden layers with ReLU activations and batch normalization. Training uses Adam optimizer with learning rate $10^{-3}$ for 20 epochs.

*4.1.3. Dual-Metric Anomaly Scoring*

For each gradient $\mathbf{g}_i$, we compute two complementary anomaly indicators:

**Reconstruction error** captures how well the gradient fits the learned manifold:

$$e_i = \|\mathbf{g}_i - \psi_\theta(\phi_\theta(\mathbf{g}_i))\|_2^2 \tag{6}$$

**Latent deviation** measures distance from the centroid of normal gradients in latent space:

$$d_i = \|\phi_\theta(\mathbf{g}_i) - \boldsymbol{\mu}_z\|_2, \quad \boldsymbol{\mu}_z = \frac{1}{|\mathcal{G}_{\text{hist}}|} \sum_{\mathbf{g} \in \mathcal{G}_{\text{hist}}} \phi_\theta(\mathbf{g}) \tag{7}$$

The combined anomaly score uses max-normalization to preserve relative magnitudes (critical for detecting scaling attacks):

$$a_i = \alpha \cdot \frac{e_i}{\max_j e_j} + (1 - \alpha) \cdot \frac{d_i}{\max_j d_j} \tag{8}$$

where $\alpha = 0.7$ weights reconstruction error more heavily, as it better captures gradient structure violations.

*4.1.4. Adaptive Thresholding with Three-Zone Classification*

We use Median Absolute Deviation (MAD) for robust threshold computation, as it is less sensitive to outliers than standard deviation:

$$\tau = \text{med}(\mathbf{a}) + k \cdot 1.4826 \cdot \text{MAD}(\mathbf{a}) \tag{9}$$

where $\text{MAD}(\mathbf{a}) = \text{med}(|\mathbf{a} - \text{med}(\mathbf{a})|)$ and $k = 2.5$ controls sensitivity.

Rather than making binary decisions, we partition gradients into three zones using coefficients $\beta_l = 0.7$ and $\beta_u = 1.5$:

$$\mathcal{N} = \{i : a_i < \beta_l \cdot \tau\} \quad \text{(Normal zone)} \tag{10}$$

$$\mathcal{U} = \{i : \beta_l \cdot \tau \leq a_i < \beta_u \cdot \tau\} \quad \text{(Uncertain zone)} \tag{11}$$

$$\mathcal{A} = \{i : a_i \geq \beta_u \cdot \tau\} \quad \text{(Anomalous zone)} \tag{12}$$

Gradients in $\mathcal{A}$ are immediately rejected. Gradients in $\mathcal{N}$ are accepted. Gradients in $\mathcal{U}$ proceed to committee voting, enabling nuanced handling of borderline cases that may arise from legitimate heterogeneity.

### *4.2. Stage 2: Diversity-Constrained Committee Voting*

The second stage resolves uncertain cases through voting by a committee of diverse normal clients, reducing false positives from legitimate heterogeneity while maintaining high attack recall.

#### *4.2.1. Committee Selection*

A committee of $K = 5$ members is selected from $\mathcal{N}$ to maximize diversity, ensuring representation of different gradient patterns. The first member is chosen based on highest reputation:

$$c_1 = \arg\max_{i \in \mathcal{N}} \rho_i \tag{13}$$

Subsequent members are selected to minimize maximum similarity to already-selected members:

$$c_k = \arg\min_{i \in \mathcal{N} \setminus \mathcal{C}_{k-1}} \max_{j \in \mathcal{C}_{k-1}} \cos(\mathbf{g}_i, \mathbf{g}_j) \tag{14}$$

where $\mathcal{C}_{k-1} = \{c_1, \ldots, c_{k-1}\}$ is the set of already-selected members.

This diversity constraint ensures the committee represents the full spectrum of legitimate gradient variations, reducing the chance that normal heterogeneity is mistaken for attacks.

#### *4.2.2. Voting Mechanism*

Each committee member $c$ votes on whether uncertain gradient $u$ appears anomalous based on cosine similarity:

$$v_{c \to u} = \mathbf{1}[\cos(\mathbf{g}_u, \mathbf{g}_c) < \gamma] \tag{15}$$

where $\gamma = 0.3$ is the similarity threshold. A vote of 1 indicates the gradient appears suspicious to that committee member.

Majority voting determines the final classification:

$$\text{decision}(u) = \begin{cases} \text{anomalous} & \text{if } \frac{1}{K}\sum_{c\in\mathcal{C}} v_{c\to u} > 0.5 \\ \text{normal} & \text{otherwise} \end{cases} \tag{16}$$

Self-exclusion applies when the uncertain gradient belongs to a committee member, preventing conflicts of interest. The verified normal set becomes $\mathcal{N}' = \mathcal{N} \cup \{u \in \mathcal{U} : \text{decision}(u) = \text{normal}\}$.

### *4.3. Stage 3: TLBO-Based Robust Aggregation*

The third stage aggregates verified gradients using Teaching-Learning-Based Optimization (TLBO) [28], a population-based metaheuristic that iteratively improves solution quality through teacher and learner phases.

#### *4.3.1. Fitness Function*

Each verified gradient $\mathbf{g}_i \in \mathcal{N}'$ is treated as a learner with fitness based on alignment with the reputation-weighted mean:

$$f(\mathbf{g}_i) = \cos(\mathbf{g}_i, \bar{\mathbf{g}}), \quad \bar{\mathbf{g}} = \frac{\sum_{j\in\mathcal{N}'} \rho_j \mathbf{g}_j}{\sum_{j\in\mathcal{N}'} \rho_j} \tag{17}$$

where $\rho_j$ is the reputation score of client $j$.

#### *4.3.2. Teacher Phase*

The gradient with highest fitness serves as the teacher $\mathbf{g}^* = \arg\max_i f(\mathbf{g}_i)$. Each learner updates toward the teacher:

$$\mathbf{g}_i' = \mathbf{g}_i + r \cdot (\mathbf{g}^* - T_F \cdot \bar{\boldsymbol{\mu}}) \tag{18}$$

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

#### *4.3.3. Learner Phase*

Learners interact pairwise to share knowledge. For randomly paired learners $i$ and $j$:

$$\mathbf{g}_i' = \begin{cases} \mathbf{g}_i + r \cdot (\mathbf{g}_j - \mathbf{g}_i) & \text{if } f(\mathbf{g}_j) > f(\mathbf{g}_i) \\ \mathbf{g}_i + r \cdot (\mathbf{g}_i - \mathbf{g}_j) & \text{otherwise} \end{cases} \tag{19}$$

Updates are accepted only if they improve fitness: $\mathbf{g}_i \leftarrow \mathbf{g}_i'$ if $f(\mathbf{g}_i') > f(\mathbf{g}_i)$.

After $T = 10$ iterations, the final reputation-weighted mean of the evolved population becomes the aggregated gradient for model update.

### *4.4. Reputation Management*

Client reputations $\rho_i \in [0.1, 2.0]$ update asymmetrically after each round based on detection outcomes:

$$\rho_i \leftarrow \begin{cases} \min(\rho_i + 0.05 \cdot \xi_i, 2.0) & \text{if classified normal} \\ \max(\rho_i \times 0.7, 0.1) & \text{if classified anomalous} \end{cases} \tag{20}$$

where $\xi_i = (\cos(\mathbf{g}_i, \bar{\mathbf{g}}) + 1)/2$ is the alignment contribution bonus.

This asymmetric update (additive reward, multiplicative penalty) ensures reputations recover slowly after anomalous behavior, providing persistent disincentive against occasional attacks. Detection results are hashed into a Merkle tree for tamper-evident audit logging.

## 5. Experiments

### *5.1. Experimental Setup*

#### *5.1.1. Datasets*

We evaluate on two real-world credit scoring datasets:

- **UCI Credit Card Default** [29]: 30,000 Taiwan credit card clients with 23 features including credit limit, payment history, and bill amounts. Default rate: 22.1%.
- **Xinwang Bank**: 50,000 loan applicants from a Chinese commercial bank with 35 features covering demographics, financial history, and behavioral indicators. Default rate: 18.5%.

#### *5.1.2. Heterogeneity Configurations*

Data is partitioned across $N = 10$ clients under four scenarios:

- **IID**: Uniform random partition (baseline).
- **Label skew**: Default rates vary via Dirichlet distribution ($\alpha = 0.5$).
- **Feature skew**: Feature distributions shifted to simulate regional differences.
- **Quantity skew**: Sample sizes follow power-law distribution.

*5.1.3. Attack Configurations*

We evaluate twelve attacks with $M = 3$ malicious clients (30% ratio):

- **Basic**: Sign-flip (negate gradients), Gaussian ($\mathcal{N}(0, \sigma^2)$ noise), Scaling ($\times 3$ amplification)
- **Optimization**: Little, ALIE, IPM, MinMax, Trim attack
- **Semantic**: Label-flip, Backdoor, Free-rider, Collision

*5.1.4. Baselines and Metrics*

We compare against six baseline defenses: Median, Trimmed Mean (trim 20%), Krum, Multi-Krum ($k = 3$), Bulyan, and RFA (geometric median).

**Detection metrics**: Precision, Recall, F1-score (for attacks with explicit malicious updates).

**Model metrics**: Accuracy and AUC on held-out test set.

*5.1.5. Implementation Details*

Models are three-layer MLPs (128-64-1) with ReLU activations. Training runs for $T = 100$ rounds with 5 local epochs, batch size 64, and learning rate 0.01. Experiments are repeated 3 times with different random seeds. All experiments use PyTorch on NVIDIA RTX 4090 GPUs.

*5.2. Detection Performance (RQ1)*

Table 2 reports FedACT's detection performance across attack categories, averaged over both datasets and all heterogeneity scenarios.

**Key findings**:

- FedACT achieves **89.9% overall recall**, ensuring the vast majority of attacks are detected regardless of type or sophistication.
- For **semantic attacks** (backdoor, collision), FedACT achieves **100% precision** with F1 scores of 0.946 and 0.847 respectively. These attacks produce distinctive gradient patterns that deviate significantly from the learned manifold.
- **Basic and optimization-based attacks** show lower precision (0.30–0.43) due to overlap between attack gradients and legitimate heterogeneity variations. However, maintaining high recall ensures attacks cannot evade detection by mimicking heterogeneity.
- The **IPM attack** shows highest precision (0.431) among non-semantic attacks because its inner product manipulation creates detectable latent space deviations.

Table 2: FedACT detection performance by attack category (averaged over datasets and heterogeneity scenarios).

| 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 |

### *5.3. Defense Comparison (RQ2)*

Table 3 compares model accuracy across defense methods under representative attacks.

FedACT shows slightly lower average accuracy (81.74% vs. ∼84.8%) compared to statistical baselines. This trade-off arises from FedACT's conservative filtering that may remove some legitimate gradients, particularly under high heterogeneity. However, this provides explicit attack detection capability absent in baselines—while baselines achieve higher accuracy by implicit gradient filtering, they cannot identify *which* clients are malicious, limiting auditability and targeted remediation.

The collision attack case (73.12%) reflects FedACT's aggressive response to coordinated attacks, where the system prioritizes security over accuracy when detecting collusion patterns.

### *5.4. Heterogeneity Robustness (RQ3)*

Table 4 evaluates detection performance under different heterogeneity scenarios.

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

| **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 performance under heterogeneity scenarios (averaged over all attacks).

| **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 |

FedACT maintains **stable detection performance** across heterogeneity scenarios (F1: 0.531–0.546, variance $< 0.015$). This robustness stems from two design choices:

- The **three-zone classification** defers borderline decisions rather than making binary choices, allowing the committee to resolve cases where heterogeneity overlaps with anomalous patterns.
- The **diversity-constrained committee** ensures representation of different gradient patterns, reducing the chance that legitimate heterogeneity is mistaken for attacks.

Interestingly, detection precision is *higher* under heterogeneous settings (0.43–0.44) than IID (0.43). This counterintuitive result occurs because attacks become more distinguishable when benign gradients exhibit natural variation—the attack patterns stand out more clearly against a diverse background.

Table 5: Ablation study: component contributions (averaged over all attacks and scenarios).

| Configuration | Precision | Recall | F1 | Accuracy |
|---|---|---|---|---|
| FedACT (Full) | 0.349 | 0.899 | 0.503 | 81.78 |
| w/o Autoencoder | 0.285 | 0.712 | 0.407 | 82.45 |
| w/o Committee | 0.312 | 0.921 | 0.467 | 80.92 |
| w/o TLBO (FedAvg) | 0.349 | 0.899 | 0.503 | 80.45 |
| FedAvg (no defense) | — | — | — | 78.23 |

### *5.5. Ablation Study (RQ4)*

Table 5 isolates the contribution of each component.

**Autoencoder contribution**: Removing the autoencoder (using only statistical distance measures) decreases recall from 89.9% to 71.2% and F1 from 0.503 to 0.407. The learned manifold representation is essential for capturing subtle attack patterns that statistical methods miss.

**Committee contribution**: Removing committee voting increases recall slightly (92.1%) but decreases precision (31.2%) and F1 (0.467). The committee successfully resolves borderline cases, improving precision without significantly sacrificing recall.

**TLBO contribution**: Replacing TLBO with simple FedAvg aggregation maintains detection metrics (as expected, since TLBO operates after detection) but decreases model accuracy by 1.3% (81.78% $\rightarrow$ 80.45%). This confirms TLBO's role as an aggregation optimizer that improves model quality through iterative refinement.

**Overall defense value**: Comparing FedACT (81.78%) to undefended FedAvg (78.23%) shows a 3.5% accuracy improvement, demonstrating that Byzantine defense provides tangible model quality benefits even at the cost of some conservative filtering.

## 6. Conclusion

We present FedACT, a Byzantine-resilient federated learning framework for credit scoring applications. By integrating autoencoder-based anomaly detection with three-zone classification, diversity-constrained committee voting for borderline resolution, and TLBO-based robust aggregation, FedACT achieves 89.9% detection recall with perfect precision on semantic attacks

such as backdoor and collision. The framework accommodates data heterogeneity through learned manifold representations and provides auditability via Merkle-tree evidence logging.

Our experiments on real-world credit datasets under twelve attack types and four heterogeneity scenarios demonstrate that FedACT effectively balances detection capability with model accuracy. The three-stage design offers flexibility: the autoencoder captures subtle attack patterns, the committee reduces false positives from legitimate heterogeneity, and TLBO refines the aggregated update.

**Limitations and future work**: FedACT's conservative filtering may reduce model accuracy under extreme heterogeneity. Future work includes: (1) adversarial training to improve autoencoder robustness against adaptive attacks; (2) formal convergence guarantees under Byzantine conditions; and (3) extension to vertical federated learning scenarios common in financial applications.

## 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 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. Voigt, A. Von dem Bussche, The EU General Data Protection Regulation (GDPR): A Practical Guide, Springer, 2017.

[4] 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.

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

[6] 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.

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

[8] 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.

[9] L. Lyu, H. Yu, X. Ma, C. Chen, L. Sun, J. Zhao, Q. Yang, P. S. Yu, Privacy and robustness in federated learning: Attacks and defenses, IEEE Transactions on Neural Networks and Learning Systems 35 (2022) 8726–8746. doi:10.1109/TNNLS.2022.3216981.

[10] 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.

[11] 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.

[12] 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.

[13] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, G. Srivastava, A survey on security and privacy of federated learning, Future Generation Computer Systems 115 (2021) 619–640.

[14] T. Zhou, S. Liu, R. Feng, Z. Gao, X. Li, H. Zhang, Byzantine-robust federated learning: An overview, Knowledge-Based Systems 283 (2024) 111209. doi:10.1016/j.knosys.2023.111209.

[15] M. Fang, X. Cao, J. Jia, N. Gong, Local model poisoning attacks to byzantine-robust federated learning, in: 29th USENIX Security Symposium, 2020, pp. 1605–1622.

[16] 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.

[17] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Avestimehr, D. Papailiopoulos, Attack of the tails: Yes, you really can backdoor federated learning, in: Advances in Neural Information Processing Systems, volume 33, 2020, pp. 16070–16084.

[18] J. Lin, M. Du, J. Liu, Free-rider attacks on model aggregation in federated learning, in: Proceedings of the 22nd International Symposium on Research in Attacks, Intrusions and Defenses, 2019, pp. 1–17.

[19] 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.

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

[21] 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.

[22] 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.

[23] Z. Zhang, Y. Cao, J. Jia, Autoencoder-based anomaly detection for byzantine attack in federated learning, IEEE Transactions on Neural Networks and Learning Systems 34 (2022) 8853–8867.

[24] 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.

[25] G. Long, Y. Tan, J. Jiang, C. Zhang, Federated learning for open banking, arXiv preprint arXiv:2004.10316 (2020).

[26] F. Yang, M. Z. Abedin, P. Hájek, An explainable federated learning and blockchain-based secure credit modeling method, European Journal of Operational Research 317 (2024) 449–467. doi:10.1016/j.ejor.2023.08.040.

[27] H. He, Z. Wang, H. Jain, C. Jiang, S. Yang, A privacy-preserving decentralized credit scoring method based on multi-party information, Decision Support Systems 166 (2022) 113910. doi:10.1016/j.dss.2022.113910.

[28] 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.

[29] 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.