EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

Edge-Driven Federated Learning System for Secure and Scalable Financial Data Sharing With Privacy Protection

Wanlin Zhang[1,*]

[1]Economics and Management School of Jiaozuo Normal College, Jiaozuo 454000, China

Abstract

INTRODUCTION: Cross-institutional collaborative financial risk modeling can improve fraud detection and anti-money laundering performance. However, data sensitivity, institutional isolation, non-IID data distributions, and heterogeneous edge nodes make traditional federated learning difficult in terms of privacy protection, communication efficiency, and robustness. OBJECTIVES: This paper aims to develop an edge-driven federated learning framework for secure cross-institutional financial risk modeling, enabling collaborative learning without sharing raw financial data. METHODS: The proposed framework integrates risk-relationship-aware dynamic grouping, privacy-constrained representation transmission, robust secure aggregation, and low-dimensional summary consistency checking. By limiting the scope of information exchange and controlling representation sharing, the framework reduces representation deviation and mitigates the influence of abnormal updates. RESULTS: Experiments were conducted on five public or synthetic financial risk datasets. The proposed method achieved the best federated AUPRC on four datasets and was only slightly lower than DSFL on the Elliptic dataset. Under client dropout and malicious-node settings, the method maintained relatively stable AUPRC performance and showed favorable communication efficiency and robustness. CONCLUSION: The results indicate that the proposed edge-driven federated learning framework can support privacy-preserving and robust cross-institutional financial risk modeling. It provides an effective solution for collaborative fraud detection and anti-money laundering under data isolation, heterogeneous edge environments, and adversarial conditions.

Keywords: Risk-Aware Sharding, Privacy-Constrained Representation Distillation, Verifiable Robust aggregation, Cross-Institutional Risk Control, non-IID Learning

Received on 16 December 2025, accepted on 17 April 2026, published on 16 September 2026

doi: 10.4108/eetsis.14493

*Corresponding author. Email: zwl_1801@126.com

1. Introduction

With the development of open banking, real-time payment, cross-border settlement, and digital asset trading, financial institutions increasingly need to conduct collaborative risk modeling. Fraud detection, anti-money laundering, credit risk identification, and abnormal transaction monitoring often rely on behavioral correlations between different accounts, institutions, and regions. The data of a single institution is usually insufficient to depict complex capital flow paths, account relationships, and dynamically changing risk patterns. However, financial data is strictly regulated and constrained. Customer identities, account behaviors, transaction records, and risk labels are usually not directly shared. Therefore, centralized modeling faces issues such as privacy, compliance, and data silos. Federated learning and privacy protection technologies provide a feasible path for multi-institutional collaborative modeling, enabling joint modeling without sharing the original data [1][2].

Meanwhile, financial risk control systems are increasingly being deployed at the edge, such as payment gateways, regional risk control nodes, bank branch systems, and mobile terminals. These nodes vary significantly in terms of communication capabilities, online stability, and computing resources, which increases the complexity of federated training [3][4]. Therefore, achieving privacy-preserving financial risk modeling in an edge heterogeneous

environment has become an important research issue in the intersection of financial intelligence and federated learning.

Existing studies have explored technologies such as financial federated modeling, edge federated learning, knowledge distillation, and secure aggregation. Some methods enhance cross-institutional anomaly detection capabilities through federated learning, secure multi-party computation, or privacy protection mechanisms [5][6]. These methods often pay less attention to practical issues on the edge side, such as latency, bandwidth fluctuations, client dynamic disconnections, and limited computing resources. Other studies alleviate non-IID data and system heterogeneity issues through hierarchical aggregation, personalized learning, knowledge transfer, or representation distillation. However, financial risk detection itself has characteristics such as rare abnormal samples, extremely imbalanced categories, uneven risk distribution across institutions, and complex transaction relationships. If clients choose to rely solely on random sampling or a single communication metric for aggregation, key risk patterns may be weakened or even ignored.

Secure aggregation, differential privacy, and robust aggregation can enhance privacy protection capabilities and increase the resistance to abnormal updates. However, existing methods are more focused on general computing efficiency or general attack protection design, and do not adequately consider the interrelationships among financial risk distribution characteristics, edge network structure, node stability, and privacy budget [7][8]. In the financial edge environment, cross-institutional risk modeling is a collaborative optimization problem that is simultaneously influenced by communication conditions, data heterogeneity, privacy constraints, risk distribution, and abnormal updates.

To achieve cross-institutional financial risk modeling, this paper proposes a privacy-preserving federated learning framework based on edge computing, exploring how to group institutions with similar conditions for training, how to share information without uploading the original data, and how to more securely combine the training results of all parties. During grouping, factors such as network speed, bandwidth quality, node stability, data distribution differences, historical risk situations, and privacy restrictions are comprehensively considered to allow nodes with similar conditions to collaborate and learn together as much as possible. In the information sharing stage, instead of directly exchanging transaction data, representative information (such as prototypes or statistical features) that has been compressed is used to convey knowledge, reducing the risk of privacy leakage. In the final result combination, the aggregation method is optimized, and consistency of updates is checked to reduce interference caused by unstable situations.

This paper regards cross-institutional financial risk modeling as a real-world problem, which is simultaneously influenced by factors such as unstable network conditions, significant data differences, and privacy constraints. The overall approach integrates grouping, information sharing, and result merging into a continuous process, making the training process more closely resemble how a real system operates. The related methods have been tested on datasets and platforms such as AMLSim, IBM AML-Data, IEEE-CIS, PaySim, and Elliptic Bitcoin.

2. Related Works

Privacy-preserving federated learning in the financial domain is mainly used for cross-institutional fraud detection, anti-money laundering, abnormal transaction identification, credit risk assessment, and transaction risk analysis. Compared to common image or text tasks, the number of abnormal samples in financial risk identification is very small, the data categories are severely imbalanced, and it involves a large number of account relationships and constantly changing risk behaviors. To identify risks more accurately, the model usually needs to simultaneously utilize transaction records, account history, transaction networks, merchant relationships, device information, geographical location, cross-institutional behaviors, and other various kinds of information. Therefore, in recent years, many studies have begun to explore federated graph learning for credit card fraud detection and federated modeling methods for financial fraud scenarios. These studies show that even if data cannot be directly shared, cross-institutional collaboration can still improve the risk identification effect [9][10].

The common data forms currently available include tabular transaction data, synthetic anti-money laundering networks, and blockchain transaction graphs. Recent studies have further discovered that methods such as graph contrast learning, graph neural networks, and heterogeneous graph modeling, when applied in situations with fewer labels and complex relationship structures, can more effectively identify suspicious transactions and abnormal behaviors [11][12][13].

The evaluation metrics used for financial fraud detection and anti-money laundering typically include AUPRC, F1, Precision, Recall, ROC-AUC, Recall@Fixed Precision, convergence rounds, offline robustness, and running delay, etc. Since the abnormal samples in financial risk tasks are usually very few, AUPRC can better reflect the model's ability to identify high-risk samples compared to the overall accuracy. Recent studies on network analysis of financial fraud detection and anti-money laundering also indicate that model evaluation needs to consider false alarm costs, result interpretability, and operational limitations in the real business environment [14][15].

Existing studies usually combine federated learning with graph learning, adaptive aggregation, or knowledge distillation to conduct fraud detection without sharing the original transaction data. However, many methods focus more on feature synergy and improvement of prediction performance, while paying less attention to edge environment issues such as communication delay, bandwidth fluctuations, insufficient computing resources, and dynamic client disconnections. Recent research on non-IID federated distillation indicates that knowledge transfer can alleviate data deviations between different clients, but most of these methods are designed for general heterogeneous learning scenarios and are not specifically tailored for financial risk modeling [16][17]. The risk-topology-aware dynamic edge

grouping method proposed in this paper further incorporates factors such as communication status, online rate, data variance, and risk distribution into the client organization process.

Recent studies have shown that even in federated learning, where the original data is not directly shared, model gradients, feature representations, or intermediate parameters can still lead to privacy leakage. Differential privacy, although it can reduce the risk of leakage, cannot completely prevent gradient leakage or data reconstruction attacks in all cases [18][19][20]. In financial edge scenarios, relying solely on a single privacy mechanism often makes it difficult to simultaneously balance the privacy budget, communication costs, and the ability to identify a small number of high-risk samples. Therefore, the privacy-constrained representation distillation method proposed in this paper takes into account the model effect, privacy budget, and potential data reconstruction risks during the information transmission process.

Edge federated learning and non-IID learning provide important foundations for related research. Existing methods usually alleviate the problems of slow training and performance degradation caused by different data distributions at the client side through clustering, shared representations, dynamic aggregation, or data-free knowledge distillation [21][22]. However, financial risk detection also needs to address issues such as scarce abnormal samples, significant cross-institutional risk differences, limited privacy budget, and interference from abnormal clients.

Secure aggregation, verifiable aggregation, and robust aggregation correspond to different security goals. Secure aggregation is mainly used to hide the update content of individual clients; verifiable aggregation is used to check whether the aggregated results have been tampered with; and robust aggregation is used to reduce the impact of abnormal or malicious updates. In recent years, some verifiable secure aggregation methods have begun to focus on issues such as verification of aggregated results, prevention of client collusion, recovery from client disconnections, and privacy protection in federated learning [23][24][25]. However, these methods usually have difficulty distinguishing the differences between changes in the distribution of financial risks and malicious client behaviors.

The robust aggregation method is also an important related direction of this paper. Malicious clients may bypass traditional robust aggregation methods through more covert or targeted model poisoning attacks [26][27][28]. To reduce the impact of such problems, the robust and secure aggregation method proposed in this paper introduces low-dimensional summaries, group-level robust estimation, and consistency checks under masked aggregation, to reduce the interference from abnormal updates and unstable nodes.

Existing research has promoted the development of financial federated learning from multiple directions, such as financial risk modeling, graph structure risk identification, privacy protection training, edge non-IID learning, and secure aggregation. The focus of this paper is to integrate client organization, restricted knowledge sharing, and robust aggregation into a continuous training process. Through this framework, the model can simultaneously consider real-world factors such as risk distribution, network structure, privacy restrictions, and abnormal updates, providing a more comprehensive solution for financial collaborative modeling with communication constraints and significant data differences.

3. Methodology

3.1. Problem Formulation

In a cross-institutional financial risk modeling system, there are K financial institution nodes, represented by $V = \{1,2,\dots,K\}$. Each node $k \in V$ only stores its own local data set D_k. That is to say, transaction records, account information, and risk labels will not be uploaded to other nodes or the central server. The global data set is merely for ease of understanding and does not represent the existence of a centralized database:

$$D_k = \{(x_{k,i}, y_{k,i})\}_{i=1}^{n_k}, D = \bigcup_{k=1}^{K} D_k, N = \sum_{k=1}^{K} n_k. \quad (1)$$

The data of each node is composed of many financial events. $x_{k,i}$ represents the i-th transaction or financial event within the node k, and $y_{k,i} \in \{0,1\}$i is the corresponding result label, where 1 indicates an anomaly, and 0 indicates normal. An event usually contains multi-level information, such as the information of the transaction itself, account-related information, the relationship between accounts, and some background information:

$$x_{k,i} = [x_{k,i}^{tr}, x_{k,i}^{acc}, x_{k,i}^{rel}, x_{k,i}^{ctx}]. \quad (2)$$

The function of the model f_θ is to output a risk probability for each event:

$$\hat{p}_{k,i} = f_\theta(x_{k,i}) \in [0,1]. \quad (3)$$

The customer, business, and risk control rules of different institutions are different, and their respective data distributions usually vary greatly and are not derived from the same pattern. Let $P_k(X,Y)$ represent the data distribution of the k-th institution. Then the overall data distribution can be written as

$$P(X,Y) = \sum_{k=1}^{K} \pi_k P_k(X,Y), \pi_k = \frac{n_k}{N}, \quad (4)$$

where generally $P_i(X,Y) \neq P_j(X,Y)$. Such heterogeneity may appear in label proportions, transaction types, account structures, and geographic distributions.

In order to distinguish between the situations of "there is a lot of high-risk data within the node" and "the training behavior of the node itself is abnormal", each node will record three types of status information. F_k^t is used to represent the

risk situation within this node, such as the proportion of abnormal transactions and the distribution of risk types; E_k^t is used to describe the network status, for example, latency, bandwidth, and whether the node is stably online. A_k^t is used to record whether there are any abnormalities during the training process, such as excessive deviation in update results, frequent disconnections, or failure in checks, etc. It should be emphasized here that a node having a large number of high-risk samples does not necessarily mean that this node is untrustworthy. Additionally, each node will also save a remaining privacy budget ϵ_k^t, which determines how much information the node can share in the future, such as risk representation, category prototypes, or statistical results, etc.

$$\sum_{t=1}^{T} \epsilon_k^t \le \epsilon_k. \tag{5}$$

The budget here is mainly used to control the scope of information representation exchange, and it does not equate to the complete sense of client-level differential privacy protection.

The method presented in this paper will dynamically determine in each training round how the clients are grouped, which nodes can exchange information with each other, and the weights that different nodes' update results will have when aggregated. Let V^t represent the set of online nodes in the t-th round, and $Q^t = \{Q_1^t, \dots, Q_{M_t}^t\}$ represents the dynamic grouping results generated in this round:

$$Q^t = \text{Shard}(G^t, \{F_k^t, E_k^t, A_k^t, \epsilon_k^t\}_{k \in V^t}), \tag{6}$$

G^t represents the current network connection structure. The entire grouping process will take into account factors such as risk coverage, communication conditions, privacy budget, and the stability of node collaboration. The entire training process can be summarized as follows:

$$\min_{\theta, \{Q^t\}, \{\phi_k^t\}, \{w_k^t\}} \sum_{t=1}^{T} \left[L_{risk}^t + \lambda_s L_{shard}^t + \lambda_d L_{distill}^t + \right. \tag{7}$$

Among them, L_{risk}^t, L_{shard}^t, $L_{distill}^t$, C_{comm}^t, P_{priv}^t, and R_{rob}^t correspond to risk detection loss, grouping cost, distillation loss, communication cost, privacy budget consumption, and robustness constraint, respectively. Formula (7) here is a comprehensive description of the entire training process.

During the information exchange stage, each node does not upload the original data, but only shares the processed representation information or prototype statistical results:

$$\tilde{h}_k^t = M_k\left(g_{\phi_k^t}(D_k); \epsilon_k^t\right). \tag{8}$$

In the aggregation, nodes upload updates processed by clipping, masking, or summarization:

$$\theta^{t+1} = \theta^t + \eta_t \cdot \text{Agg}\left(\{\tilde{\Delta}_k^t, w_k^t\}_{k \in V^t}\right), \tag{9}$$

where $\tilde{\Delta}_k^t$ is the securely processed update and w_k^t is the aggregation weight. Aggregation outputs, including deviation scores, dropout records, and verification results, are fed back to A_k^{t+1}, achieving sharding, distillation, aggregation, and feedback.

In testing, each node performs local prediction using the trained model $\theta^\star$:

$$\hat{y}_{k,j}^{test} = \mathbb{I}\left(f_{\theta^\star}(x_{k,j}^{test}) \ge \rho\right), \tag{10}$$

where the threshold ρ is determined on the validation set.

This paper assumes that the central aggregator will complete the training and aggregation in the normal process, but it may still attempt to infer the data distribution or sensitive features from the information uploaded by the clients. There may be a certain proportion (not exceeding α) of abnormal or malicious clients who may interfere with the model training. However, the attackers cannot directly obtain the original data of the normal clients, nor can they disrupt the masking mechanism, or completely control all the nodes in a certain group. In this threat scenario, this research endeavors to minimize the risk of information leakage, reduce the impact of obvious abnormal updates on the model, and maintain the stability of the training process even when nodes are disconnected or the network is unstable.

3.2. Overall Framework

The federated learning framework based on edge computing is shown in Figure 1, which is mainly used for cross-institutional financial risk identification. In this framework, different financial institutions or edge nodes respectively store their own local data, and the model is used to determine the risk probability of transactions, accounts, or financial events. Each node only processes local data. The central aggregator mainly coordinates the status of each node, performs dynamic grouping, and summarizes the protected model updates, which achieves the global model training and distribution without directly accessing the original financial data.

The entire framework consists of the training phase and the inference phase. During the training process, the system will successively perform state updates, dynamic grouping, local training, information compression and sharing within groups, robust and secure aggregation, and state feedback. In the inference phase, each node directly uses the trained model to handle local transactions or risk events.

Before round t, the system updates lightweight states, including the financial risk state F_k^t, edge topology state E_k^t, collaborative anomaly state A_k^t, and remaining privacy budget ϵ_k^t. The status information is mainly used to describe the current network and training conditions of the nodes, and does not include sensitive contents such as original transaction records, account information, or identity data. RT-DES will dynamically adjust the node groups based on this information, allowing nodes with similar conditions to be trained together first.

After the grouping is completed, each node first trains the model locally and then only shares the processed risk features or statistical results within the group. Subsequently, VRSA will check and merge the results uploaded by each group, and try to filter out abnormal updates, malicious nodes, or interference caused by unstable networks.

This framework is more like a continuously adjusting collaborative training process that ensures data security while also striving to balance the effectiveness of risk identification, communication costs, and system stability.

Figure 1 shows the overall framework for cross-institutional financial risk identification, including the training and inference stages. During the training process, RT-DES dynamically adjusts the node grouping, PC-ERD is responsible for information sharing and representation compression within the groups, and VRSA completes secure aggregation while preventing the leakage of original financial data. During inference, each node directly uses the trained model to process local financial events and outputs the corresponding risk probability.

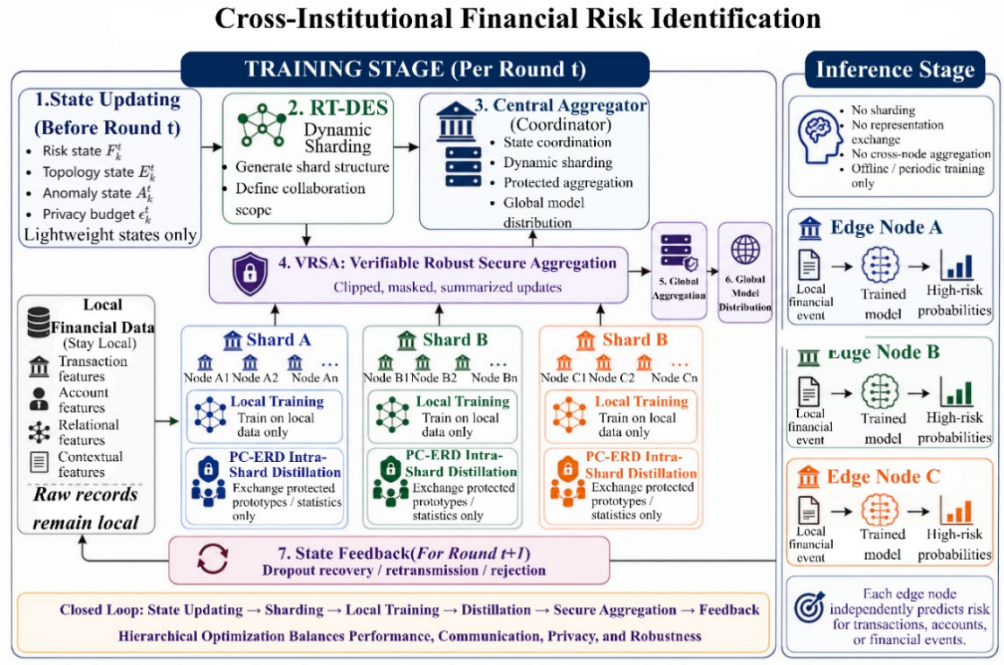

Figure 1. Overall framework of the proposed edge-driven federated learning system

3.3. Module Descriptions

This framework is mainly composed of three parts: dynamic grouping based on risk relationships and network structures, information compression and knowledge transmission under privacy constraints, and secure aggregation with consistency checks. The system will separately analyze the financial risk state F_k^t, edge topology state E_k^t, and collaborative anomaly state A_k^t to guide the clients on how to group, how to share information within the group, and how to identify abnormal updates during the aggregation process.

RT-DES: Risk–Topology-Aware Dynamic Edge Sharding

As shown in Figure 2, RT-DES generates the grouping structure before each round of training, which is used to determine the scope of subsequent information sharing and grouping-level aggregation:

$$Q^t = \text{Shard}(V^t, \{F_k^t, E_k^t, A_k^t, \epsilon_k^t\}_{k \in V^t}). \quad (11)$$

Here, F_k^t is used to represent the risk coverage situation and the degree of category complementarity of the node, E_k^t is used to estimate whether the communication between different nodes is cost-effective, ϵ_k^t indicates whether this node is suitable for participating in information compression and sharing, and A_k^t is used to determine whether there is abnormal collaborative behavior. It should be noted that a high risk level of a node does not mean it is "untrustworthy". The abnormal behavior here refers more to whether the behavior during the training process is stable or normal.

To ensure the reproducibility of the grouping process, the degree of matching between nodes is weighted by several intuitive factors, including communication cost, risk complementarity, whether the privacy budget matches, and the stability of the collaboration process. All indicators will first be uniformly scaled to the range of 0 to 1. The weights can be adjusted on the validation set or fixed, and the results under different weight settings can also be reported.

RT-DES employs a constrained greedy grouping method. Nodes with higher anomaly scores will not be grouped in the

initial group. Other nodes will be allocated to the group with the lower overall cost based on factors such as group capacity, drop probability, privacy budget, and risk coverage. If certain nodes cannot be reasonably allocated, they will only undergo local training in the current round. Overall, RT-DES is more like a heuristic grouping strategy that balances communication costs, risk coverage, privacy constraints, and system stability. It provides a dynamic collaboration scope for subsequent representation distillation and secure aggregation.

Figure 2 illustrates how RT-DES divides the groups Q^t before each round of training. The generated groups determine which nodes can share and aggregate information together, which is the collaboration scope of PC-ERD and VRSA. Meanwhile, it achieves a holistic balance among communication overhead, privacy protection, risk coverage, and system stability.

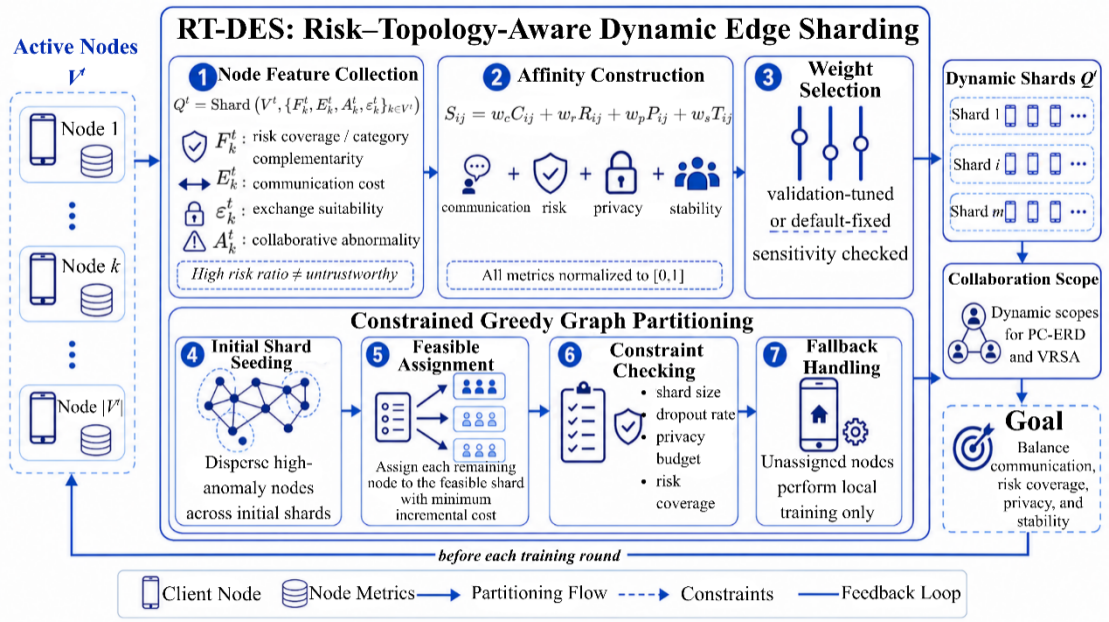

Figure 2. Overview of RT-DES: Risk–Topology-Aware Dynamic Edge Sharding

PC-ERD: Privacy-Constrained Edge Representation Distillation

The function of PC-ERD is to conduct limited information sharing within the same group without exchanging the original financial data, account details, or identity information. If relying only on the aggregation of ordinary model parameters, an institution may have difficulty learning rare fraud patterns or abnormal transaction features in other institutions. However, directly sharing samples or complete features would bring privacy risks. Therefore, PC-ERD will share the protected category prototypes or statistical representations within the group Q_m^t generated by RT-DES. The scope of information exchange is also limited by the group structure and the remaining privacy budget.

As shown in Figure 3, the overall process of PC-ERD includes representing perturbations, intra-group prototype exchange, and information distillation subject to budget constraints.

Node k will first convert financial events into low-dimensional representations through the local encoder g_{ϕ_k}:

$$z_{k,i} = g_{\phi_k}(x_{k,i}). \tag{12}$$

Before exchanging the prototypes, the nodes will add noise to these representations or category prototypes based on the current privacy budget to protect them:

$$\tilde{z}_{k,i} = z_{k,i} + \xi_{k,i}, \xi_{k,i} \sim \mathcal{N}(0, \sigma_k^2 I). \tag{13}$$

The nodes will first compress the original financial events into low-dimensional feature representations, and then add certain perturbations before sharing, thereby achieving a balance between information sharing and privacy protection.

Figure 3 illustrates the basic process of PC-ERD: Nodes first convert local financial events into low-dimensional representations, then add certain perturbations to these representations, and subsequently share the processed prototypes or statistical information only within the current group Q_m^tt.

VRSA: Verifiable Robust Secure Aggregation

As shown in Figure 4, VRSA can complete the group-level aggregation without directly exposing the complete model update content of each node. At the same time, it tries to minimize the impact of abnormal updates, malicious poisoning attacks, and unstable nodes. The term “verifiable” means that the system can use consistency checks to

determine whether the group aggregation results are abnormal.

In the t-th training round, node k will first perform pruning on its local update:

$$\Delta\bar{\theta}_k^t = \Delta\theta_k^t \cdot \min\left(1, \frac{\rho}{\| \Delta\theta_k^t \|_2 + \eta}\right), \tag{14}$$

where ρ is the clipping threshold and η is a smoothing constant. It then generates a low-dimensional summary:

$$q_k^t = H^t \Delta\bar{\theta}_k^t + v_k^t, \tag{15}$$

H^t is generated by an openly available random seed; v_k^t represents the aggregated noise added. These aggregated pieces of information are mainly used for estimating the update deviation and conducting consistency checks.

Within the group Q_m^t, the aggregator will first calculate a relatively robust center update, and then distribute the weights based on the deviation between each node's result and the center result:

$$a_k^t = \frac{\exp(-\mu e_k^t)}{\sum_{j \in Q_m^t} \exp(-\mu\, e_j^t)}. \tag{16}$$

When allocating weights, factors such as the stability of the nodes in the past, the records of disconnections, and the financial risk status will also be taken into account. Subsequently, each node will upload the weighted updates that have been protected by zero-sum masking:

$$\widetilde{\Delta\theta}_k^t = a_k^t \Delta\bar{\theta}_k^t + r_k^t, \sum_{k \in Q_m^t} r_k^t = 0. \tag{17}$$

In this way, the aggregator can only restore the overall update result at the group level. If an abnormality occurs in a group, the system will choose to reset or directly reject the update of that group, and simultaneously feed back the relevant records to RT-DES. During this process, VRSA can reduce the impact of abnormal updates on the training process and improve the stability of the overall training.

Figure 4 illustrates the overall process of VRSA, including steps such as updating the clipping, generating summaries, weighting based on deviations, uploading using zero-sum masks, and verifying through consistency labels, thereby achieving group-level aggregation. The group results of each round will be accepted, restored, retransmitted, or rejected depending on the situation, and at the same time, relevant stability records, disconnection situations, and risk information will be fed back to RT-DES.

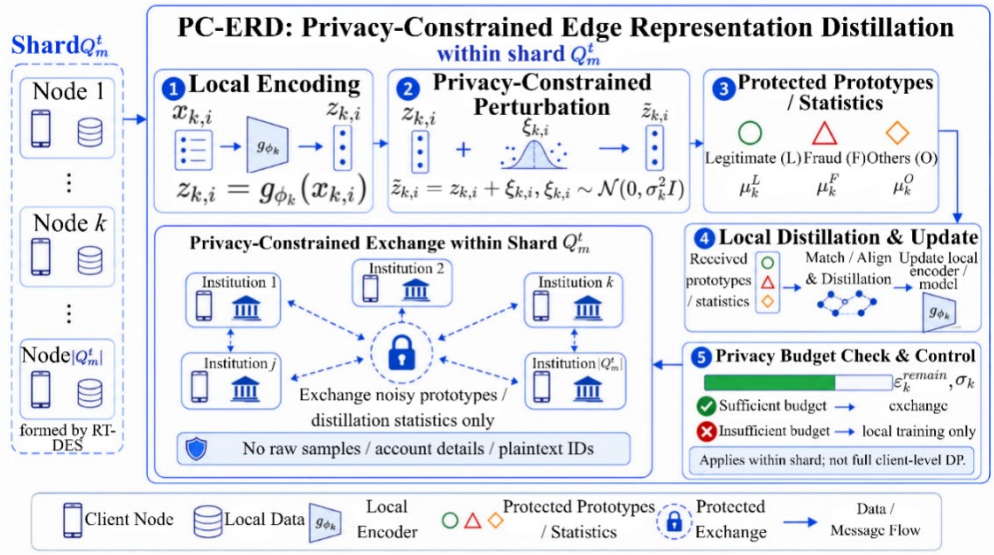

Figure 3. Privacy-Constrained Edge Representation Distillation within a Shard

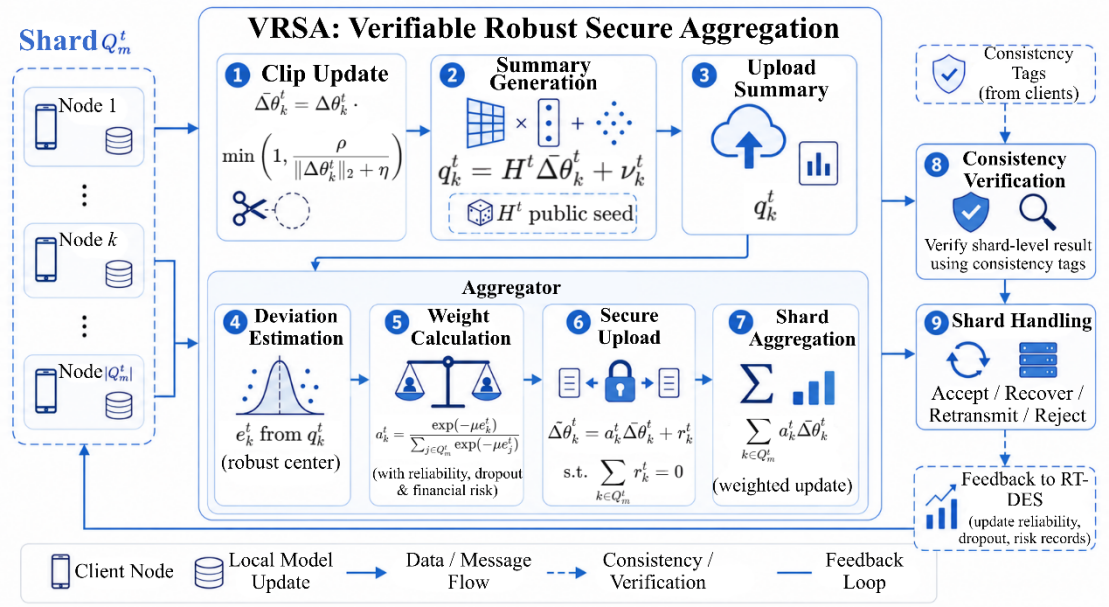

Figure 4. Workflow of Verifiable Robust Secure Aggregation (VRSA)

3.4. Objective Function and Optimization

This paper adopts a hierarchical approximate optimization approach. RT-DES is responsible for making group decisions and dividing the nodes into different collaborative units; PC-ERD mainly optimizes the training within groups and the information sharing across groups. VRSA ensures the stability and reliability of the aggregation process through consistency checks.

At the beginning of round t, the server identifies the online node set V^t, distributes the global model θ^t, and generates the shard structure:

$$Q^t = \{Q_1^t, Q_2^t, \ldots, Q_{M_t}^t\}. \tag{18}$$

Given shard Q_m^t, each node $k \in Q_m^t$optimizes the local PC-ERD objective:

$$\min_{\theta_k, \phi_k} L_k^t = L_{pred,k}^t + \beta_1 \sum_{c=1}^{C} \| p_k^{(t,c)} - \bar{p}_m^{(t,c)} \|_2^2 + \tag{19}$$

$L_{pred,k}^t$ represents the weighted prediction loss. The prototype distillation part is used to make the locally learned representations and the category prototypes within the group as consistent as possible. $L_{ib,k}^t$ is used to reduce redundancy or possibly sensitive information. $L_{rec,k}^t$ (achieved through gradient inversion) is used to reduce the risk of recovering the original data. This objective function is optimized only within the scope allowed by the current group structure and privacy budget.

After the local training is completed, VRSA will estimate the updated deviations based on the low-dimensional summaries uploaded by each node, and then calculate the aggregation weights accordingly. The server will only perform the final aggregation on the groups that pass the consistency check:

$$\theta^{t+1} = \theta^t + \eta_g \sum_{m \in M_{valid}^t} \lambda_m^t \widehat{\Delta\Theta}_m^t, \tag{20}$$

where M_{valid}^t is the set of valid shards and $\widehat{\Delta\Theta}_m^t$ is the aggregated update of shard m. By default, shard weights are normalized by valid sample size:

$$\lambda_m^t = \frac{N_m^t}{\sum_{j \in M_{valid}^t} N_j^t}, N_m^t = \sum_{k \in Q_m^t} n_k . \tag{21}$$

After the aggregation, various states will be updated: communication delay, bandwidth, and online status will be updated to E_k^{t+1}; update deviation, disconnection status, loss change, and verification failure records are updated to A_k^{t+1}; the privacy consumption of the exchange is updated to ϵ_k^{t+1} Here, F_k^t only represents the local risk distribution and is not mixed with abnormal collaborative states.

In each training round, the data is first grouped by RT-DES, then locally trained by PC-ERD, followed by aggregation by VRSA. After that, the model is deployed, locally verified, and the status is updated. The server only receives the weighted verification results or aggregated statistical information. The validation set is used for parameter tuning and early stopping, while the test set is only used for reporting the final results.

The default settings are T=100, E=3, batch size=1024, using AdamW, learning rate 10^{-3}, η_g=1.0, with a dimension of 128, noise 0.8, $\beta_1 = 0.5$, $\beta_2 = 10^{-3}$, and $\beta_3 = 0.1$. Overall, these three modules are not optimized together but are adjusted through feedback to balance the system among effectiveness, communication, privacy, and stability as much as possible.

4. Experiment and Results

4.1. Experimental Setup

Datasets, Labels, and Splits

This paper utilizes three main datasets and two extended datasets, as summarized in Table 1. The AMLSim, IBM AML-Data HI-Small, and IEEE-CIS fraud detection datasets are used for the main performance comparison, ablation experiments, and robustness tests. The PaySim Synthetic Mobile Money and Elliptic Bitcoin Dataset are used for supplementary verification, such as mobile payment scenarios and transaction network risk identification. This paper builds a federated edge experimental environment using public data and synthetic methods. The client divisions, communication states, and online conditions are all generated according to rules.

Table 1. Datasets, Labels, and Splits

Dataset	Type	Positive Label	Data Split	Usage
AMLSim	Synthetic AML	laundering	approx. 700k/150k/150k	Main experiments, ablation
IBM AML-Data HI-Small	Synthetic AML	laundering/suspicious	approx. 3.56M/0.76M/0.76M	Main experiments, non-IID
IEEE-CIS	Public payment fraud	isFraud = 1	approx. 413k/89k/89k	Main experiments, identity missingness
PaySim	Synthetic mobile payment	isFraud = 1	approx. 4.45M/0.95M/0.95M	Extended validation
Elliptic Bitcoin	Bitcoin transaction graph	illicit	time steps 1–30/31–40/41–49	Extended graph experiments

All datasets are divided in chronological order, resulting in training sets, validation sets, and test sets, in an effort to avoid future information leakage. For IEEE-CIS, the test set is re-divided based on TransactionDT within the publicly available training set. For Elliptic, the division is carried out according to time steps, where unlabeled nodes do not participate in supervised training, threshold selection, or metric calculation. AMLSim, IBM AML-Data, and PaySim are themselves synthetic or semi-synthetic data, so the experimental results are mainly used for method validation, and their effects in real production environments cannot be directly equated.

Data Preprocessing and Leakage Prevention

For the tabular data, the preprocessing procedure is summarized in Table 2. Numerical features are first standardized using the statistics of the training set. The categorical features are first truncated by frequency, and then represented by embeddings: categories that appear less than 20 times in the training set are uniformly regarded as "unknown", and each field retains at most the top 10,000 most common values. The embedding dimensions are mapped according to the specified values:

$$d_{emb} = \min(32, \lceil \sqrt{|C|} \rceil).$$

The steps for handling missing values are to add a missing flag and perform numerical filling. Numerical fields are filled with the median of the training set, while categorical fields are directly filled with "unknown". In the IEEE-CIS dataset, the transaction table and the identity table are connected through Transaction ID. Transactions without matching identity information will retain the missing flag.

Table 2. Data Preprocessing and Leakage-Prevention Rules

Dataset	Removed Fields	Leakage-Prevention Treatment
AMLSim	label, alert flag, posterior rule-based fields	Only prediction-time features are used
IBM AML-Data	label, SAR/alert posterior fields	Preprocessors are fitted only on training data
IEEE-CIS	isFraud	The official unlabeled test set is not used
PaySim	isFraud, isFlaggedFraud	Rule-based fraud indicators are excluded
Elliptic	class label, node id	Community partitioning is fitted only during training time steps

All preprocessing information, such as normalization parameters, category tables, fill-in values, community divisions in the graph structure, prototype statistics, grouping status, and neighborhood risk statistics, is calculated from the training set, local data, or the current training round. The

validation set is only used for parameter tuning, early stopping, and threshold selection. The test set labels will not be involved in any form of preprocessing, grouping, statistical calculation, or threshold setting.

Federated Partitioning and Edge-State Simulation

This study partitions each dataset into multiple simulated institutions, as summarized in Table 3. Each institution only used a portion of the data. During the splitting process, the original elements were basically not changed. For instance, the category proportions and feature structures remained the same, only being redistributed to different institutions.

Table 3. Federated Client Partitioning

Dataset	K	Client Partitioning Rule	Non-IID Statistics
AMLSim	20	bank_id; otherwise merged account-graph communities	label JSD, sample-size CV
IBM AML-Data	30	From the Bank, large banks are split by account communities	label JSD, transaction-type JSD
IEEE-CIS	24	User–device clusters from ProductCD, card1, addr1, DeviceType, and time windows	label JSD, identity missing rate
PaySim	40	Accounts, transaction types, and step windows	type JSD, amount shift
Elliptic	20	Louvain communities within training time steps, plus temporal partitioning	illicit ratio, average degree, modularity

Label skew is the difference between institutions were only roughly estimated. One was used to show the distribution variations, and the other was used to check if the data volume was evenly distributed. Additionally, the actual bank network data could not be obtained, so the delays, bandwidth, and online conditions were all simulated rather than based on actual records.

Training, Evaluation, and Statistical Protocols

The experimental environment and model settings are summarized in Table 4. All experiments were conducted on a machine equipped with an RTX 4090 GPU with 24 GB memory, running Ubuntu 22.04, Python 3.10, PyTorch 2.3.1, CUDA 12.1, and PyTorch Geometric 2.5. For the model architecture, tabular datasets were processed using a three-layer MLP, while the Elliptic dataset was modeled using a two-layer GraphSAGE. Unless otherwise specified, all methods followed the same experimental settings, with only minor parameter adjustments when connecting specific modules.

Table 4. Training, Attack, and Evaluation Protocols

Item	Setting
Optimizer	AdamW
Learning rate/weight decay	$10^{-3}/10^{-4}$
Batch size/local epochs/rounds	1024/3/100
Representation dimension	128
Repetitions	5 seeds, mean ± standard deviation
Malicious-client ratio	5%, 10%, 20%, 30%
Attack types	Direction flipping, label flipping, and Gaussian noise
Main metrics	AUPRC, F1, Recall@Precision = 0.90/0.95
System metrics	Communication, convergence rounds, training time, inference time
Privacy metrics	Attribute inference AUC, membership inference AUC, reconstruction MSE

AUPRC is the primary evaluation metric. F1 and recall under fixed precision are calculated only by selecting thresholds on the validation set. The server only receives the weighted validation results or the aggregated statistical information that has undergone security aggregation. The training time includes local computation, communication, and aggregation overheads. The inference stage only calculates the forward propagation. Robustness is measured by the relative decrease in AUPRC:

$$\text{DropRate} = \frac{M_{clean} - M_{attack}}{M_{clean}} \times 100\%.$$

Here, M denotes AUPRC by default.

4.2. Baseline Methods

The comparison methods can be classified into several categories, including the situation where only local training is conducted, the ideal scenario where the data is centralized for training, and different types of federated learning methods (standard methods, methods under non-IID settings, methods with privacy constraints, and methods with robust mechanisms). All federated methods use the same set of

experimental settings, the hyperparameters are also selected within the same validation range, and the test set does not participate in any parameter tuning.

Local training serves as the lowest reference, while centralized training serves as the theoretical upper limit reference. These are used to aid understanding but do not participate in communication or privacy-related comparisons. Other federated methods cover common baseline types and are used together with the method proposed in this paper as the overall comparison objects.

For fair comparison, the prototype-based method uniformly uses the same representation dimensions and exchange frequencies; the privacy-related methods use the same set of noise settings; the robustness experiments uniformly involve malicious clients, attack types, and attack intensities. The statistical method for communication overhead is also consistent, calculated based on the number of model updates, prototypes, summaries, consistency labels, and masked related messages. The ablation experiments mainly examine what changes occur when certain modules are removed, different strategies are replaced, different robust aggregation methods are used, or the backbone network with parameter matching is replaced.

4.3. Quantitative Results

This section compares local training, centralized training, several common baseline methods for federated learning, privacy protection methods, robust aggregation methods, and the method proposed in this paper. All federated methods use the same data partition, number of clients, model structure, input features, batch size, number of local training rounds, and communication rounds. Each experiment is repeated five times, and the results are reported as the mean ± standard deviation. AUPRC does not depend on the threshold, while the threshold for F1 is determined only on the validation set. Statistical significance is tested using a two-sided paired t-test, and corrected using the Holm–Bonferroni method.

Main Performance Comparison

As shown in Table 5, the method proposed in this paper achieved the best overall federated AUPRC results on AMLSim, AML-Data, IEEE-CIS, and PaySim, while it was slightly lower than the DSFL-based methods on Elliptic Bitcoin. The performance of F1 was also largely consistent, indicating that the overall effect was relatively stable in most tabular data and transaction flow risk tasks.

After applying the Holm–Bonferroni correction, only the AUPRC improvement on PaySim remained statistically significant. Therefore, these results are more appropriately interpreted as an overall improvement under the current experimental setup. The results of centralized training still exceeded those of most federated methods, which also indicates that the performance gap still exists under privacy constraints and non-IID partition conditions.

Convergence Analysis

On AML-Data, we additionally recorded the validation AUPRC for each round during the training process. Table 6 presents the results for the validation set, while Table 5 shows the results for the test set. These two sets cannot be directly compared. Figure 5 illustrates the training trajectories for five random seeds, including the average curve and the ±1 standard deviation fluctuation range.

.

Table 5. Main Performance Comparison on Five Datasets

Method	AMLSim AUPRC/F1	AML-Data AUPRC/F1	IEEE-CIS AUPRC/F1	PaySim AUPRC/F1	Elliptic AUPRC/F1
Local-only	0.672±0.020/ 0.593±0.022	0.641±0.028/ 0.561±0.031	0.548±0.023/ 0.492±0.025	0.701±0.010/ 0.636±0.012	0.512±0.046/ 0.462±0.051
Centralized upper bound	0.836±0.013/ 0.750±0.015	0.817±0.018/ 0.729±0.021	0.681±0.018/ 0.604±0.021	0.866±0.006/ 0.784±0.009	0.634±0.039/ 0.579±0.043
FedAvg	0.716±0.017/ 0.633±0.019	0.693±0.024/ 0.612±0.027	0.587±0.020/ 0.528±0.023	0.744±0.009/ 0.672±0.011	0.556±0.039/ 0.503±0.044
FedProx	0.735±0.018/ 0.652±0.020	0.714±0.023/ 0.635±0.025	0.609±0.022/ 0.548±0.024	0.763±0.008/ 0.691±0.010	0.566±0.043/ 0.511±0.046
DP-FedAvg	0.701±0.024/ 0.618±0.025	0.678±0.030/ 0.598±0.032	0.571±0.021/ 0.512±0.024	0.731±0.011/ 0.659±0.013	0.536±0.050/ 0.482±0.055
FedEDS-style	0.748±0.021/ 0.668±0.022	0.729±0.026/ 0.651±0.027	0.634±0.021/ 0.573±0.024	0.778±0.010/ 0.706±0.012	0.584±0.048/ 0.528±0.052
DSFL-style	0.774±0.017/ 0.692±0.019	0.754±0.021/ 0.676±0.023	0.641±0.019/ 0.581±0.022	0.804±0.007/ 0.733±0.009	0.626±0.038/ 0.570±0.041
SecAgg+Median	0.782±0.014/ 0.700±0.016	0.764±0.020/ 0.686±0.022	0.648±0.018/ 0.588±0.021	0.794±0.009/ 0.724±0.011	0.611±0.035/ 0.552±0.039
SCAFFOLD	0.746±0.018/ 0.664±0.020	0.724±0.022/ 0.646±0.024	0.625±0.020/ 0.563±0.023	0.771±0.008/ 0.698±0.010	0.572±0.041/ 0.518±0.045
FedProto	0.765±0.016/ 0.684±0.018	0.745±0.024/ 0.667±0.026	0.651±0.019/ 0.590±0.022	0.787±0.008/ 0.716±0.010	0.593±0.046/ 0.538±0.050
Krum/ Multi-Krum	0.752±0.020/ 0.672±0.021	0.738±0.025/ 0.660±0.027	0.628±0.022/ 0.568±0.024	0.781±0.010/ 0.710±0.012	0.579±0.049/ 0.525±0.053

Proposed method	0.813±0.016/ 0.728±0.018	0.797±0.021/ 0.711±0.024	0.664±0.020/ 0.595±0.023	0.848±0.007/ 0.768±0.010	0.619±0.043/ 0.558±0.047
raw p-value	0.012/0.018	0.009/0.027	0.084/0.112	0.002/0.006	0.214/0.338
Holm p-value	0.084/0.108	0.072/0.135	0.336/0.336	0.020/0.054	0.428/0.428

Table 6. Validation AUPRC Convergence on AML-Data

Communication Round	FedAvg	FedProx	DP-FedAvg	FedEDS-style	DSFL-style	SecAgg+Median	Proposed method
10	0.519±0.026	0.535±0.024	0.507±0.031	0.546±0.027	0.563±0.025	0.557±0.024	0.603±0.025
20	0.596±0.024	0.614±0.023	0.582±0.029	0.631±0.025	0.646±0.023	0.650±0.022	0.702±0.023
30	0.652±0.027	0.669±0.025	0.639±0.030	0.684±0.026	0.707±0.024	0.716±0.023	0.754±0.026
40	0.675±0.025	0.694±0.022	0.659±0.028	0.710±0.025	0.731±0.023	0.737±0.022	0.781±0.022
50	0.688±0.026	0.706±0.024	0.671±0.031	0.722±0.024	0.744±0.022	0.751±0.021	0.775±0.027
60	0.684±0.029	0.701±0.026	0.666±0.033	0.716±0.027	0.739±0.025	0.747±0.024	0.789±0.024

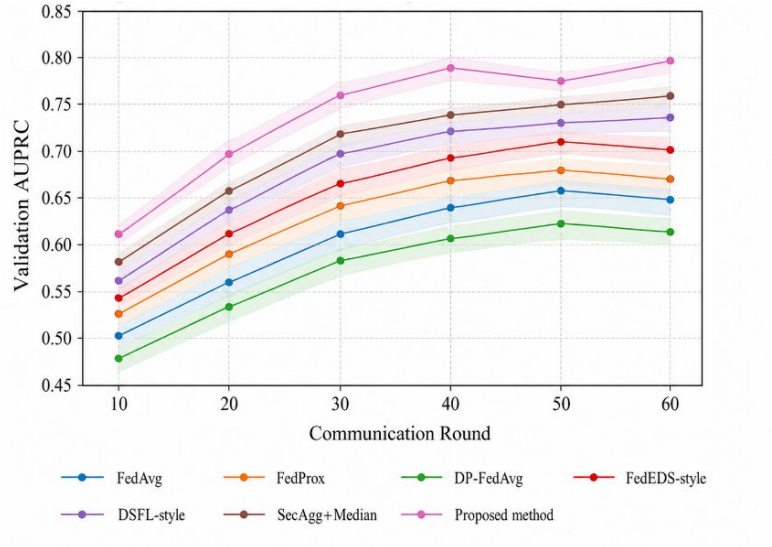

Figure 5. Validation AUPRC convergence curves on AML-Data

As shown in Table 6 and Figure 5, as the communication rounds increase, the effectiveness of most methods generally rises. The method proposed in this paper shows a rapid improvement in the first 30 rounds and can maintain a high level in most rounds, indicating that convergence is relatively faster under the current settings. This trend mainly reflects the dynamic changes during the training process and should be viewed together with the ablation results.

Efficiency, Communication, and Computational Cost

This section compares the parameter size, FLOPs, communication overhead, convergence rounds, training time, and inference cost on the AML-Data dataset. All methods use the same basic model structure, but due to the different additional modules, there will be differences in the parameter quantity and computational cost. The calculation method for communication overhead is consistent with that in Section 4.1, including model updates, prototypes, summaries, consistency labels, and messages related to masks.

Table 7. Efficiency and Computational Cost on AML-Data

Method	Parameters (M)	FLOPs/batch (G)	Communication (GB)	Rounds to 95% Final AUPRC	Training Time (min)	Inference Time (ms/1k samples)
FedAvg	1.21	0.84	18.3±1.1	36±4	88±7	7.7±0.4
FedProx	1.21	0.84	18.7±1.2	35±3	92±8	7.9±0.5
DP-FedAvg	1.21	0.86	19.8±1.4	37±4	105±9	8.3±0.5
FedEDS-style	1.39	0.97	21.1±1.5	35±4	111±12	9.0±0.5
DSFL-style	1.34	0.93	16.5±1.1	36±3	87±8	8.8±0.5

SecAgg+Median	1.29	0.91	17.0±1.0	37±3	90±9	8.5±0.4
Proposed method	1.52	1.08	14.2±0.9	29±3	79±7	9.9±0.6

As shown in Table 7, the proposed method achieves the lowest communication volume, which is 13.9% lower than DSFL-style, the lowest-communication baseline. It also reaches 95% of the final AUPRC in the fewest rounds. However, it has higher parameters, FLOPs, and inference time than several baselines, proving that its advantage is in communication efficiency and convergence speed.

4.4. Qualitative Results

To better understand the performance of the model, this paper selects a typical successful case from AMLSim and a failed case from the Elliptic Bitcoin dataset for illustration. These cases are mainly used to explain the model behavior and are not used to independently prove the causal effect of a certain module. All classification thresholds were determined on the validation set.

Successful Case: Multi-Hop Money-Laundering Path in AMLSim

Figure 6 illustrates the laundering path in AMLSim: A1 → A2 → A3 → A4 → A1. Its characteristic is that the transaction amount is large, the time interval is short, and it forms a local closed loop.

Table 8. Summary of the Successful AMLSim Case

Metric	SecAgg+Median	Proposed Method
Risk score range	0.52–0.64	0.81–0.88
Exceeds threshold 0.67	No	Yes
Neighborhood risk ratio	0.55–0.64	0.55–0.64
Prototype distance reduction	—	15.6%–21.7%
Ground-truth label	laundering	laundering

As shown in Figure 6 and Table 8, it can be seen that the SecAgg+Median method failed to identify this path, while the method in this paper provided risk scores above the threshold for all four transactions. The results show that such structured paths are more easily captured after cross-shard prototype alignment, indicating that the model is more sensitive to this circular fund flow pattern.

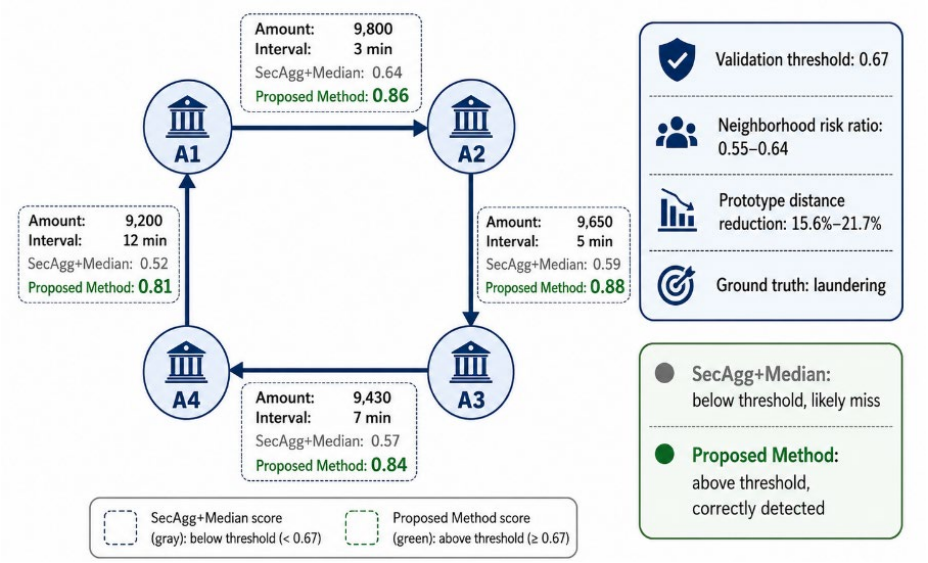

Figure 6. Successful Multi-hop Money Laundering Path Detection in AMLSim

The SecAgg+Median method failed to identify the circular transaction path of A1→A2→A3→A4→A1 because the risk scores it provided were all below the validation threshold of 0.67. However, the scores in this paper exceeded this threshold, thus successfully detecting this money laundering route.

Failure Case: Boundary Node Misclassification in Elliptic

Figure 7 illustrates a false positive case of Elliptic. Node E-2265 is connected to several unknown nodes as well as high-risk communities.

Table 9. Summary of the Elliptic Failure Case

Metric	Value
Node ID	E-2265
Ground-truth label	licit
Predicted probability	0.73
Validation threshold	0.60
Neighbor risk ratio/unknown-neighbor ratio	0.48/0.46

As shown in Figure 7 and Table 9, -2265 is actually a legitimate node, but the model still wrongly identifies it as an illegal node. This error might be related to incomplete labels in the blockchain transaction graph and limited contextual information. This case illustrates that the method presented in this paper may be influenced by surrounding high-risk signals, thereby overestimating the risk score of most normal nodes. The case here is mainly for illustrative purposes, and the true conclusion is based on the overall experiment and ablation results.

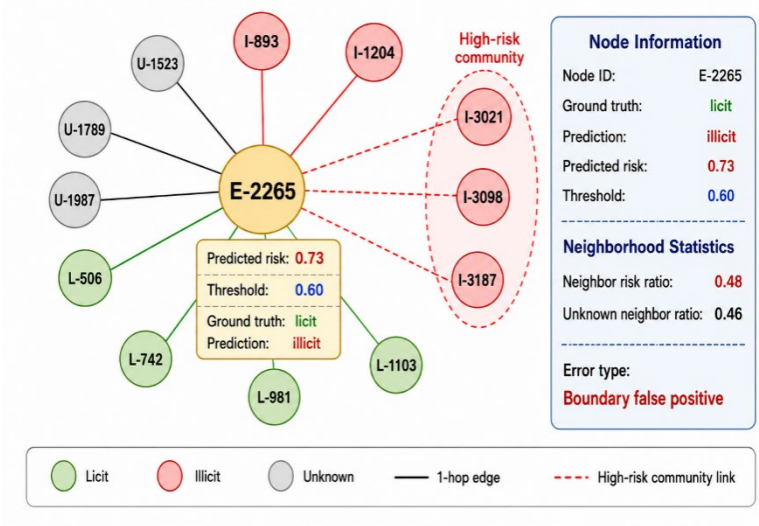

Figure 7. Elliptic boundary node misclassification case

Although E-2265 is a legitimate transaction node, due to its connection with numerous high-risk communities and unknown neighbors, the model ultimately assigned a risk score of 0.73, which was higher than the 0.60 threshold set during the verification stage. Therefore, it was wrongly classified as a high-risk node. This case demonstrates that in areas with relatively ambiguous structures, surrounding risk information can sometimes influence the model's judgment, thereby leading to false alarms.

4.5. Robustness Validation

To assess the stability of the model in non-ideal federated environments, this paper introduces three types of disturbances on the AML-Data: client disconnection, malicious client attack, and noisy updates. All experiments use the same data partitioning, number of clients, model structure, batch size, local training rounds, and communication rounds as described in 4.3.

The comparison methods include several common types of federated learning, privacy protection, and robust aggregation methods, such as FedAvg, FedProx, DP-FedAvg, FedEDS-style methods, DSFL-style methods, SecAgg+Median, SCAFFOLD, FedProto, as well as robust aggregation methods like Krum/Multi-Krum, Truncated Average, and Bulyan. The offline experiments use random disconnections; the malicious client experiments employ direction flipping and label flipping attacks; in addition, Gaussian noise is added to the client updates:

$$\Delta\theta_k^{mal} = \Delta\theta_k + \xi_k, \xi_k \sim \mathcal{N}(0, \sigma^2 I).$$

Performance degradation is measured by

$$\text{DropRate} = \frac{\text{AUPRC}_{clean} - \text{AUPRC}_{attack}}{\text{AUPRC}_{clean}} \times 100\%.$$

Overall Robustness Comparison

Figure 8 presents the robustness results of AML-Data under three interference conditions, including client disconnection, malicious client attack, and Gaussian noise injection. The figure shows the average results of five experiments, ±1 standard deviation, and the variation curves with different random seeds.

As shown in Figure 8(a), when the client disconnection rate increases from 0% to 40%, the performance of all methods will decline. The AUPRC of the method proposed in this paper drops from 0.803 to 0.692, a decrease of approximately 13.8%; SecAgg+Median drops from 0.762 to 0.653, a decrease of approximately 14.3%. Overall, the stability of this

method under disconnection conditions is close to that of other methods, with a slight advantage, but a higher disconnection rate will significantly affect the overall model effect.

Figure 8(b) presents the results of the malicious client attack. As the proportion of malicious clients increases, the performance of all methods declines. In cases of 20% and 30% malicious clients, the AUPRC of the method proposed in this paper is 0.704 and 0.642, respectively, which is approximately 20% lower than the normal setting. In most cases, it still outperforms several common robust aggregation methods, but as the attack proportion continues to increase, this advantage gradually weakens. Therefore, VRSA can be regarded as a mechanism for reducing the impact of abnormal updates.

From Figure 8(c), it can be seen that lower levels of noise only cause minor fluctuations. When the noise intensity increases to $\sigma = 0.30$, the AUPRC of the method in this paper drops to 0.684, indicating that even strong noise still significantly affects the local update. Overall, the method in this paper performs relatively stably under moderate interference, but it is not completely consistent under all random seeds, attack intensities, and interference conditions.

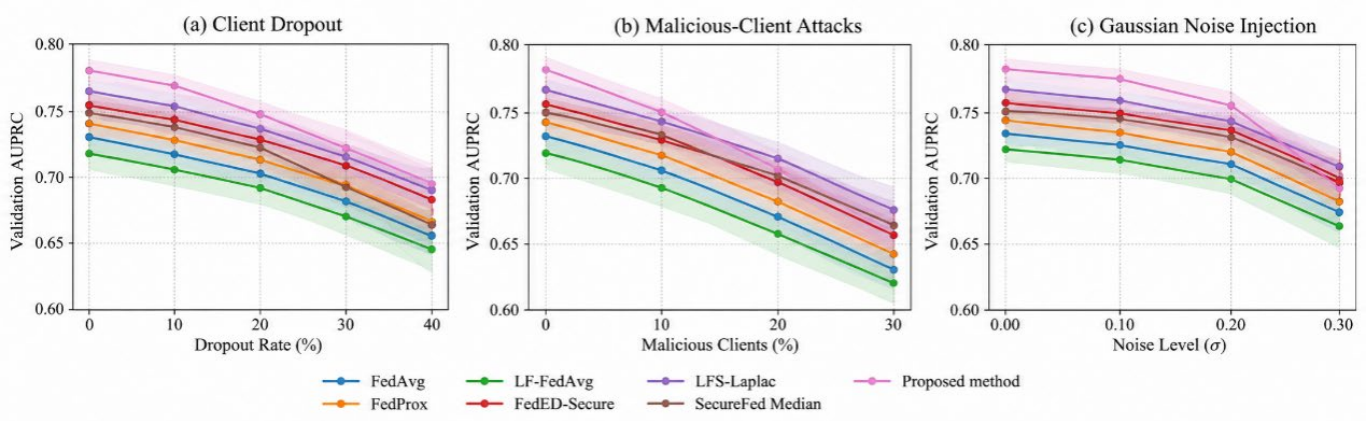

Figure. 8. Overall Robustness Comparison on AML-Data. Robustness results under client dropout, malicious-client attacks, and Gaussian noise injection

Boundary Conditions

This paper mainly tested several common interference scenarios, such as random client disconnections, direction reversal attacks, tag reversal attacks, and Gaussian noise. VRSA can reduce the impact of obviously abnormal updates through bias weighting and consistency checks, but it is not a complete Byzantine defense solution. For more complex attack scenarios, stronger detection and security mechanisms are needed to further verify.

Ablation Analysis of Robustness

To observe the functions of different modules under an interference environment, this paper sets up three experimental scenarios: 30% of the clients going offline, 20% of malicious clients, and Gaussian noise $\sigma = 0.20$.

Table 10. Robustness Results after Removing Different Modules

Method Variant	30% Dropout AUPRC	20% Malicious Clients AUPRC	$\boldsymbol{\sigma = 0.20}$ Noise AUPRC
Full Model	0.741±0.018	0.704±0.020	0.738±0.018
w/o RT-DES	0.675±0.023	0.689±0.022	0.719±0.020
w/o PC-ERD	0.714±0.020	0.672±0.024	0.669±0.025
w/o VRSA	0.725±0.019	0.631±0.027	0.681±0.024
w/o feedback	0.702±0.021	0.657±0.025	0.699±0.022

From Table 10, it can be seen that the impact of different modules varies in different scenarios. When there is a 30% drop in connection, the performance drops most significantly after removing RT-DES; under malicious client attacks, the removal of VRSA has the greatest impact; and in the case of noise updates, the performance drops the most after removing PC-ERD. This indicates that RT-DES is more inclined to enhance the stability of node collaboration, PC-ERD is more helpful in stabilizing the representation of information, and VRSA is more important for handling abnormal updates. However, these results mainly reflect the performance in the current experiment and cannot directly indicate that these modules will have the same effect in all scenarios.

4.6. Ablation Study

To analyze the respective roles played by RT-DES, PC-ERD, VRSA, and state feedback, this paper conducted ablation experiments on AMLSim, AML-Data, and IEEE-CIS. Besides simply removing the core modules, this paper also tested different alternative solutions, sub-module variants, robust aggregation methods, and control models with similar parameter quantities to determine whether the performance improvement was due to the proposed method itself or simply because the model was larger or used common robust strategies.

Ablation of Core Components, Submodules, and Replacement Strategies

Based on Table 11, removing any core module will cause the average AUPRC to decrease, with the removal of PC-ERD having the greatest impact. This indicates that cross-shard representation distillation has the most obvious relationship with performance improvement, while RT-DES and VRSA also contribute to the overall effect. Further replacement experiments show that this improvement is not solely due to ordinary shards, simple prototype sharing, conventional robust aggregation, or increased model parameters. However, after Holm correction, the differences between some variants are not significant, so the interpretation of the effect of individual modules still requires caution.

Table 11. Ablation Results of Core Components, Submodules, and Replacement Strategies

Type	Variant	Average AUPRC	ΔAUPRC	Holm p-value
Full model	Full Model	0.765±0.012	—	—
Core module	w/o RT-DES	0.737±0.014	↓0.028	0.060
Core module	w/o PC-ERD	0.719±0.016	↓0.046	0.048
Core module	w/o VRSA	0.736±0.015	↓0.029	0.055
Core module	w/o feedback	0.748±0.013	↓0.017	0.096
RT-DES submodule	w/o online-rate state	0.746±0.014	↓0.019	0.092
RT-DES submodule	w/o heterogeneity state	0.742±0.015	↓0.023	0.074
PC-ERD submodule	w/o representation noise	0.746±0.015	↓0.019	0.094
PC-ERD submodule	w/o adversarial reconstruction constraint	0.752±0.014	↓0.013	0.128
VRSA submodule	w/o deviation-based weighting	0.727±0.016	↓0.038	0.049
VRSA submodule	w/o consistency checking	0.741±0.015	↓0.024	0.076
Replacement strategy	Random Sharding replacing RT-DES	0.730±0.015	↓0.035	0.054
Replacement strategy	Prototype Sharing replacing PC-ERD	0.730±0.016	↓0.035	0.049
Robust aggregation replacement	SecAgg replacing VRSA	0.739±0.014	↓0.026	0.060
Robust aggregation replacement	Krum replacing VRSA	0.731±0.017	↓0.034	0.052
Robust aggregation replacement	Multi-Krum replacing VRSA	0.735±0.016	↓0.030	0.056
Robust aggregation replacement	Trimmed Mean replacing VRSA	0.736±0.015	↓0.029	0.058
Robust aggregation replacement	Bulyan replacing VRSA	0.731±0.017	↓0.034	0.050
Parameter-matched control	Param-matched backbone	0.681±0.019	↓0.084	0.018
Backbone baseline	FedAvg backbone only	0.667±0.020	↓0.098	0.011

Note: Holm p-values are obtained from paired t-tests between the Full Model and each variant, with multiple-comparison correction.

Supplementary Analysis of System Metrics, Privacy, and Robustness

On AML-Data, the communication cost of the complete model is lower, and its convergence speed is faster; however, this advantage is significantly weakened after removing RT-DES. Removing VRSA leads to a more pronounced decline in the model's performance under malicious client attacks, indicating that it is crucial for mitigating the impact of

abnormal updates. Removing PC-ERD makes the model more unstable in the presence of noisy updates, suggesting that representation distillation helps maintain the stability of representation learning.

Further black-box privacy attack experiments showed that without PC-ERD, the risks of attribute inference and membership inference will increase, while the reconstruction error will decrease. This indicates that PC-ERD can reduce representation leakage but cannot provide strict privacy protection.

Parameter Sensitivity Analysis

This paper also analyzed the effects of the noise scale σ_k, the target piece size m_{tar}, and the summary dimension d_q on the results of AML-Data.

Table 12. Sensitivity Results of Key Hyperparameters

Parameter Setting	AUPRC	Communication (GB)	Privacy Attack AUC ↓

$\sigma_k = 0.4$	0.812±0.014	14.1±0.9	0.62±0.03
$\sigma_k = 0.8$, default	0.803±0.013	14.2±0.9	0.56±0.02
$\sigma_k = 1.2$	0.786±0.016	14.3±1.0	0.53±0.02
$m_{tar} = 4$	0.787±0.017	15.6±1.1	—
$m_{tar} = 8$, default	0.803±0.013	14.2±0.9	—
$m_{tar} = 12$	0.796±0.015	14.8±1.0	—
$d_q = 64$	0.789±0.016	13.7±0.8	—
$d_q = 128$	0.798±0.014	14.0±0.9	—
$d_q = 256$, default	0.803±0.013	14.2±0.9	—

As shown in Table 12, $\sigma_k = 0.8$ achieves a relatively balanced trade-off among model performance, privacy risk, and communication overhead. Moreover, the overall performance of $m_{tar} = 8$ is also relatively stable. Overall, RT-DES is more inclined to enhance communication efficiency, PC-ERD is more conducive to maintaining representation stability, and VRSA is mainly used to mitigate the impact of abnormal updates. However, these results still heavily rely on the current experimental setup and are essentially empirical observations.

5. Discussion

The experimental results show that in the federated financial learning environment, the model performance is not only related to parameter aggregation or privacy protection, but also affected by factors such as risk distribution, network status, communication conditions, and aggregation methods. Recent research on federated learning for credit card fraud detection also indicates that even if data cannot be directly shared, multi-institutional collaborative training can still enhance the ability to identify financial risks [29]. Some anti-money laundering studies have also found that the structural relationships and time sequences between transactions are very important for identifying money laundering behaviors [30]. Compared with these works, this paper integrates risk representation, node differences, communication restrictions, privacy-constrained information exchange, and robust aggregation into a unified training process for unified consideration.

The ablation experiments show that RT-DES, PC-ERD, and VRSA correspond to different types of problems. RT-DES is more focused on improving communication efficiency and adapting to node disconnections, which is consistent with some research results on federated learning based on client grouping and prototype sharing [31]. PC-ERD is related to stability and a lower risk of privacy attacks. Existing studies have pointed out that even without sharing the original data, intermediate representations or gradient information may still pose a risk of data leakage [32]. However, PC-ERD is mainly an empirical risk mitigation method.

The method presented in this paper achieved the highest federated AUPRC on AMLSim, AML-Data, IEEE-CIS, and PaySim, but was slightly lower than the DSFL-based methods on Elliptic Bitcoin. This indicates that this framework is more suitable for tabular data and transaction flow scenarios. Additionally, not all improvements can be maintained as significant after Holm–Bonferroni correction. Therefore, these results are more appropriately understood as overall performance advantages under the current experimental conditions.

Some security aggregation studies emphasize that, while protecting the gradient privacy of the client side, it is also necessary to be able to verify whether the aggregated results have been tampered with [33]; other privacy aggregation methods consider how to resist poisoning attacks as much as possible under the condition of privacy restrictions [34]. Compared with these approaches, VRSA merely combines several commonly used methods, including low-dimensional summaries, weighted by shards, masking processing, and consistency checks. Its main function is to mitigate the impact of abnormal updates and unstable nodes. Experiments have proved that it is more stable when the proportion of malicious clients is not high, but when the attack becomes stronger, this advantage will gradually weaken.

This method is indeed more efficient in terms of communication and can achieve approximately 95% final AUPRC with fewer rounds. However, correspondingly, its model parameters, computational cost, and inference time will be higher than those of FedAvg, FedProx, and some security aggregation methods. Therefore, it is more suitable for application scenarios where communication resources are relatively limited, but the risk detection effect is highly valued.

This work also has some limitations. AMLSim, IBM AML-Data, and PaySim are all simulated or semi-simulated data, and there is still a gap from the real financial environment. The privacy assessment currently mainly relies on tests such as attribute inference, membership inference, and reconstruction attacks, and cannot be equivalent to strict (ϵ,δ)-DP. The robustness experiments, although they consider some attack scenarios, such as more complex adaptive poisoning and collaborative backdoors, have not been covered. The delay, memory, and throughput during actual deployment have not yet been systematically tested. In the future, there may be a greater focus on real financial environments, lighter models, stronger attack detection, and easier auditing designs.

6. Conclusion

This paper proposes an edge-driven federated training framework to support collaborative risk identification across clients.

In the experimental part, multiple datasets such as AMLSim, AML-Data, IEEE-CIS, PaySim, and Elliptic Bitcoin were applied. The results showed that the performance was superior to the comparison methods on the first four datasets, but slightly inferior to the DSFL-like methods on Elliptic Bitcoin. Considering the influence of multiple statistical corrections, the improvements on some datasets were not stable after significance tests. Therefore, a more reasonable interpretation is that there is an overall advantage in the trend, but the magnitude of the advantage varies slightly among different datasets.

The ablation experiments explain the functions of each module. RT-DES mainly affects the client grouping and communication efficiency, PC-ERD improves the representation stability to some extent, VRSA alleviates the interference caused by abnormal updates to a certain degree, and the feedback mechanism is mainly used for the continuous adjustment of the state during multi-round training. From the results of the replacement experiments and parameter matching, these improvements are not solely caused by changes in model size or general aggregation strategies. However, accordingly, the computational and inference costs will also increase. This needs to be considered separately based on specific requirements during actual deployment.

The subsequent work can be further advanced in several directions, such as more realistic verification based on actual financial business scenarios, lightweight structural design, uncertainty modeling, graph structure risk propagation constraints, and more auditable aggregation mechanisms. These directions mainly focus on enhancing stability and practical usability.

Author Contribution

Conceptualization, W.Z.; methodology, W.Z.; software, W.Z.; validation, W.Z.; formal analysis, W.Z.; investigation, W.Z.; resources, W.Z.; data curation, W.Z.; writing—original draft preparation, W.Z.; writing—review and editing, W.Z.; visualization, W.Z.; supervision, W.Z.; project administration, W.Z.; funding acquisition, W.Z. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement

The datasets can be requested by corresponding author.

References

[1] Asif H, Min S, Wang X, Vaidya J. US-UK PETs prize challenge: Anomaly detection via privacy-enhanced federated learning. IEEE Transactions on Privacy, 2024;1:3-18.

[2] Arora S, Beams A, Chatzigiannis P, Meiser S, Patel K, Raghuraman S, Zamani M. Privacy-preserving financial anomaly detection via federated learning & multi-party computation. Proceedings of the 2024 annual computer security applications conference workshops (ACSAC workshops); Honolulu, HI, USA, 9-10 December 2024. p. 270-279.

[3] Jia N, Qu Z, Ye B, Wang Y, Hu S, Guo S. A comprehensive survey on communication-efficient federated learning in mobile edge environments. IEEE Communications Surveys & Tutorials, 2025;27(6):3710-3741.

[4] Luo L, Zhang C, Yu H, Sun G, Luo S, Dustdar S. Communication-efficient federated learning with adaptive aggregation for heterogeneous client-edge-cloud network. IEEE Transactions on Services Computing, 2024;17(6):3241-3255.

[5] Khan MSI, Gupta A, Seneviratne O, Patterson S. Fed-RD: Privacy-preserving federated learning for financial crime detection. Proceedings of the 2024 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr), Hoboken, NJ, USA, October 22-23, 2024; p. 1-9.

[6] Park DY, Ko IY, Lee TH, Lee J. Federated Gradient Boosting for Financial Fraud Detection: An Empirical Study in the Banking Sector. Proceedings of the 34th ACM International Conference on Information and Knowledge Management; Seoul, Korea, November 10-14, 2025. p. 5089-5093.

[7] Zhou H, Zheng Y, Jia X. Towards robust and privacy-preserving federated learning in edge computing. Computer Networks, 2024;243:110321.

[8] Wang L, Huang M, Zhang Z, Li M, Wang J, Gai K. RaSA: Robust and adaptive secure aggregation for edge-assisted hierarchical federated learning. IEEE Transactions on Information Forensics and Security, 2025;20:4280-4295.

[9] Tang Y, Liang Y. Credit card fraud detection based on federated graph learning. Expert Systems with Applications, 2024;256:124979.

[10] Huang Q, Zhu Y, Liao T, Huang Z. A Federated Learning-Based Approach for Robust and Accurate Financial Fraud Detection. Proceedings of the 2025 International Conference on Artificial Intelligence and Digital Finance; Chengdu, China, June 13-15, 2025. p. 20-24.

[11] Lu H, Wang H. Graph contrastive pre-training for anti-money laundering. International Journal of Computational Intelligence Systems, 2024;17(1):307.

[12] Bakhshinejad N, Nguyen UT, Ghahremani S, Soltani R. A Graph-Based Deep Learning Model for the Anti-Money Laundering Task of Transaction Monitoring. Proceedings of the 16th International Joint Conference on Computational Intelligence (IJCCI), Porto, Portugal, 20-22 November 2024. p. 496-507.

[13] Johannessen F, Jullum M. Finding money launderers using heterogeneous graph neural networks. The Journal of Finance and Data Science, 2025;11:100175.

[14] Chen Y, Zhao C, Xu Y, Nie C, Zhang Y. Deep learning in financial fraud detection: Innovations, challenges, and applications. Data Science and Management, 2026;9(2):100162.

[15] Deprez B, Vanderschueren T, Baesens B, Verdonck T, Verbeke W. Network Analytics for Anti-money Laundering—A Systematic Literature Review and Experimental Evaluation. INFORMS Journal on Data Science, 2025;5(2):81-190.

[16] Yao D, Pan W, Dai Y, Wan Y, Ding X, Yu C, Sun L. FedGKD: Toward heterogeneous federated learning via global knowledge distillation. IEEE Transactions on Computers, 2023;73(1):3-17.

[17] Wu S, Chen J, Nie X, Wang Y, Zhou X, Lu L, Menhaj W. Global prototype distillation for heterogeneous federated learning. Scientific Reports, 2024;14(1):12057.
[18] Luo G, Chen N, He J, Jin B, Zhang Z, Li Y. Privacy-preserving clustering federated learning for non-IID data. Future Generation Computer Systems, 2024;154:384-395.
[19] Hu J, Du J, Wang Z, Pang X, Zhou Y, Sun P, Ren K. Does differential privacy really protect federated learning from gradient leakage attacks? IEEE Transactions on Mobile Computing, 2024;23(12):12635-12649.
[20] Tan Q, Li Q, Zhao Y, Liu Z, Guo X, Xu K. Defending against data reconstruction attacks in federated learning: An information theory approach. Proceedings of the 33rd USENIX security symposium (USENIX Security 24), Philadelphia, PA, USA, August 14–16, 2024. p. 325-342.
[21] Chen L, Zhang W, Dong C, Zhao D, Zeng X, Qiao S, Tan CW. Fedtkd: A trustworthy heterogeneous federated learning based on adaptive knowledge distillation. Entropy, 2024;26(1):96.
[22] Shao J, Wu F, Zhang J. Selective knowledge sharing for privacy-preserving federated distillation without a good teacher. Nature Communications, 2024;15(1):349.
[23] Wang R, Xiong L, Geng J, Xie C, Li R. An effective and verifiable secure aggregation scheme with privacy-preserving for federated learning. Journal of Systems Architecture, 2025;161:103364.
[24] Ming Y, Wang S, Wang C, Liu H, Deng Y, Zhao Y, Feng J. VCSA: Verifiable and collusion-resistant secure aggregation for federated learning using symmetric homomorphic encryption. Journal of Systems Architecture, 2024;156:103279.
[25] Zhou S, Wang L, Chen L, Wang Y, Yuan K. Group verifiable secure aggregate federated learning based on secret sharing. Scientific Reports, 2025;15(1):9712.
[26] Mai P, Yan R, Pang Y. Rflpa: A robust federated learning framework against poisoning attacks with secure aggregation. Advances in Neural Information Processing Systems, 2024;37:104329-104356.
[27] Yang H, Gu D, He J. A robust and efficient federated learning algorithm against adaptive model poisoning attacks. IEEE Internet of Things Journal, 2024;11(9):16289-16302.
[28] Wei K, Li J, Ding M, Ma C, Jeon YS, Poor HV. Covert model poisoning against federated learning: Algorithm design and optimization. IEEE Transactions on Dependable and Secure Computing, 2023;21(3):1196-1209.
[29] Reddy VVK, Reddy RVK, Munaga MSK, Karnam B, Maddila SK, Kolli CS. Deep learning-based credit card fraud detection in federated learning. Expert Systems with Applications, 2024;255:124493.
[30] Shadrooh S, Nørvåg K. SMoTeF: Smurf money laundering detection using temporal order and flow analysis: S. Shadrooh and K. Nørvåg. Applied Intelligence, 2024;54(15):7461-7478.
[31] Ye R, Xu M, Wang J, Xu C, Chen S, Wang Y. Feddisco: Federated learning with discrepancy-aware collaboration. Proceedings of the 40th International Conference on Machine Learning; Honolulu, Hawaii, USA, July 23-29, 2023. JMLR.org. p. 39879-39902.
[32] Noorbakhsh SL, Zhang B, Hong Y, Wang B. Inf2Guard: An Information-Theoretic framework for learning Privacy-Preserving representations against inference attacks. Proceedings of the 33rd USENIX Conference on Security Symposium; Philadelphia, PA, USA, August 14-16, 2024. Berkeley: USENIX Association, USA. 2024. p. 2405-2422.
[33] Ni L, Gong X, Li J, Tang Y, Luan Z, Zhang J. rfedfw: Secure and trustable aggregation scheme for byzantine-robust federated learning in internet of things. Information Sciences, 2024;653:119784.
[34] Guan M, Bao H, Li Z, Pan H, Huang C, Dai HN. SAMFL: Secure Aggregation Mechanism for Federated Learning with Byzantine-robustness by functional encryption. Journal of Systems Architecture, 2024;157:103304.