Privacy-Preserving Financial Fraud Detection Using Federated Heterogeneous Temporal GraphSAGE

Sabab Islam[1], Md. Abul Kalam Azad[2], Abdul Kadar Muhammad Masum[3], Ramona Birau[4], Virgil Popescu[5], Roxana Mihaela Nioata (Chireac)[6], Gabriela Ana Maria Lupu (Filip)[7]
College of Business, Texas A&M University Commerce, Texas, USA[1]
Department of Business and Technology Management, Islamic University of Technology, Dhaka 1704, Bangladesh[2]
Department of Computer Science and Engineering, Southeast University, Dhaka 1208, Bangladesh[3]
"Eugeniu Carada" Doctoral School of Economic Sciences, University of Craiova, Craiova, Romania[4,6,7]
Faculty of Economics and Business Administration, University of Craiova, Craiova, Romania[5]

Abstract*—Digital financial ecosystems face mounting exposure to fraudulent transactions that collectively account for trillions of dollars in losses each year. Existing approaches suffer from three recurring deficiencies: they represent all transaction participants within a single undifferentiated node space, rely on fixed decision boundaries incapable of accommodating evolving fraud distributions, and fail to exploit the semantic diversity among entity categories including accounts, merchants, and transaction types. This study introduces FHT-GraphSAGE, a federated heterogeneous temporal graph learning framework for privacy-preserving financial fraud detection. The architecture combines heterogeneous temporal graph construction, a relation-aware GraphSAGE encoder with sinusoidal temporal edge embeddings, and a Federated Averaging optimization scheme that enables cross-institutional collaborative learning without exposing raw transaction records. Evaluation across two complementary benchmarks, the PaySim Mobile Money Dataset and the Credit Card Fraud 2023 Dataset, demonstrates consistent superiority over ten competitive baselines. On PaySim, FHT-GraphSAGE achieves an accuracy of 0.963, an F1-score of 0.939, an AUC-ROC of 0.986, and, most importantly under severe class imbalance, an AUC-PR of 0.724. On the Credit Card Fraud 2023 Dataset, it attains an accuracy of 0.941, an F1-score of 0.913, an AUC-ROC of 0.969, and an AUC-PR of 0.721. As AUC-PR is the most informative metric for minority-class detection under extreme imbalance, these two figures (0.724 and 0.721) constitute the primary evidence of the framework's fraud-detection capability. Results are reported as the mean over five independent runs with distinct random seeds, and the improvements over the strongest baseline are statistically significant ($p < 0.01$). Ablation experiments confirm the non-redundant contribution of each architectural component, and robustness evaluations demonstrate stable minority-class detection performance down to a fraud ratio of 0.25%.

***Keywords*—Financial fraud detection; federated learning; heterogeneous temporal graph; GraphSAGE; privacy-preserving machine learning; graph neural networks**

I. Introduction

Unauthorized activity in digital payment infrastructure causes an estimated $5.127 trillion in worldwide monetary losses annually [1]. Beyond the immediate financial damage, incidents at this scale erode public trust in banking systems and constrain broader participation in formal economic structures. Rapid digitalization of payment services has compounded these risks: transaction volumes now far exceed what human reviewers can examine, while coordinated attacks simultaneously exploit multiple accounts, payment instruments, and merchant channels [2]. Methods that evaluate each transaction independently are therefore structurally incapable of capturing the relational dimension of contemporary fraud.

Several inherent characteristics of this problem compound detection difficulty. Adversarial strategies evolve continuously, outpacing the update cycles of rule-based systems [2]. Legal and regulatory constraints prevent financial institutions from pooling raw transaction histories, even when doing so would materially improve detection [3]. Fraudulent activity constitutes a negligibly small fraction of total transaction volume, which drives standard classification objectives toward the majority class and suppresses minority-class recall [4].

Current detection pipelines address individual aspects of this problem without resolving the full set of challenges. Rule-based filters offer transparency but demand continuous manual recalibration as tactics shift [5]. Conventional supervised classifiers process transactions as statistically independent records, discarding the relational network that fraud rings exploit [6]. Graph neural network methods encode structural topology but the majority still operate on static adjacency matrices that become stale as new interactions arrive. Fixed decision boundaries degrade progressively under concept drift [7]. Privacy-preserving approaches that could overcome cross-institutional data barriers have rarely been combined with graph-structured fraud representations [8]. These gaps collectively indicate that a unified solution must address relational heterogeneity, temporal dynamics, distributed privacy constraints, and generalization across structurally dissimilar financial-fraud datasets simultaneously.

The proposed framework satisfies these four requirements within a single coherent architecture. First, transaction records from each participating institution are transformed into heterogeneous temporal graphs that explicitly separate customer, merchant, and transaction node categories, connected by semantically typed directed edges. Second, a temporal edge encoding function maps interaction time gaps into continuous sinusoidal embeddings, injecting sequential behavioral context into neighborhood aggregation. Third, a relation-aware GraphSAGE encoder applies separate aggregation pathways per relation type, fusing the resulting representations through a learnable projection to produce inductive node embeddings that

generalize to previously unseen entities. Fourth, a Federated Averaging protocol coordinates parameter aggregation across institutional clients without requiring any institution to transmit raw records, satisfying practical privacy and regulatory requirements.

By unifying heterogeneous temporal graph construction, relation-aware inductive aggregation, and privacy-preserving federated optimization within a single coherent architecture, FHT-GraphSAGE represents a meaningful advancement in collaborative financial fraud detection. Beyond its immediate contributions to transaction security, this work opens broader pathways for applying federated graph learning to domains characterized by complex, distributed sensitive data, evolving relational structure, and strict institutional privacy requirements, including healthcare claim auditing, telecommunications anomaly detection, and supply chain integrity verification.

II. Literature Review

Foundational work on transaction fraud relied on statistical profiling derived from historical transaction records. Bhattacharyya et al. [9] benchmarked several data mining classifiers on credit card transactions and demonstrated that minority-class scarcity reliably degrades detection across all evaluated methods. Bolton and Hand [10] introduced a peer-group analysis approach that flags cardholders whose expenditure trajectories diverge significantly from those of statistically comparable users. With the maturation of supervised learning, Whitrow et al. [11] established that aggregating records into temporal behavioral windows substantially elevates classifier performance over raw transaction features. Dal Pozzolo et al. [12] showed that sliding-window retraining only partially compensates for concept drift and requires manual scheduling of update cycles. Roy et al. [6] demonstrated measurable accuracy gains from feed-forward neural networks relative to shallow classifiers on nonlinear spending patterns. Wang et al. [13] combined reconstruction-based autoencoders with ensemble classifiers to improve performance under severe label imbalance. A limitation shared across all these approaches is that each transaction is evaluated in isolation, without access to the broader relational network within which fraud rings operate.

Representing financial activity as a graph provides a principled mechanism for reasoning over entity relationships and interaction patterns. Akoglu et al. [14] surveyed graph anomaly detection spanning community-level, subgraph-level, and node-level methods. Liu et al. [15] applied isolation-based anomaly scoring to graph-structured data, though this approach discards both edge semantics and temporal ordering. The introduction of graph neural networks substantially expanded representational capability. Kipf and Welling [16] proposed spectral graph convolutions that propagate neighborhood features through a symmetrically normalized adjacency operator. Hamilton et al. [17] developed GraphSAGE, an inductive framework that learns node embeddings by sampling and aggregating local neighborhoods, supporting generalization to previously unobserved nodes. Dou et al. [18] designed an attention-weighted graph network capable of identifying fraudsters who deliberately camouflage their activity by mimicking legitimate behavioral patterns. Liu et al. [19] constructed a heterogeneous financial graph that incorporates multiple relationship categories to reduce feature inconsistency across edge types. More recently, Khosravi et al. [20] proposed an attentional spatial-temporal GNN for transaction fraud detection, while Islam et al. [21] introduced a layer-weighted GCN to better separate fraud patterns across heterogeneous transaction types, illustrating continued efforts to refine relational representations. A persistent limitation across these graph-based methods is dependence on static adjacency matrices that do not evolve as new transactions arrive, making them structurally insensitive to behavioral drift over time.

Capturing temporal dynamics in fraud requires architectures that revise node representations incrementally as interactions occur. Rossi et al. [22] introduced the Temporal Graph Network, which maintains a persistent memory state per node and refreshes it through a gated recurrent mechanism upon each observed interaction. Elapsed time since the most recent event is encoded through a learnable projection mapping scalar gaps to continuous embeddings. Li et al. [23] developed a heterogeneous dynamic graph architecture for spatiotemporal forecasting that constructs typed subgraphs and refreshes their adjacency through multi-head self-attention over recent observation windows. Dal Pozzolo et al. [24] showed that adaptive sliding-window retraining partially tracks distributional shift but remains constrained by periodic update schedules that lag real-time fraud evolution. Cui et al. [25] recently combined temporal-spatial-semantic graph convolution with reinforcement learning and federated coordination in a single framework, representing one of the few prior attempts to jointly address temporal, relational, and privacy-preserving objectives. However, their reinforcement-learning-driven adaptation differs fundamentally from the sinusoidal temporal encoding and relation-specific aggregation pathways proposed here.

Velickovic et al. [26] introduced graph attention networks that compute neighbor contribution weights through a shared attention function over concatenated node feature pairs. Brody et al. [27] subsequently identified a fundamental limitation in this formulation: applying the linear transform before the nonlinearity causes neighbor ranking to be independent of the querying node, yielding static rather than dynamic attention. Their revised model restructures the computation so that independent linear projections of the query and key nodes are combined inside the nonlinearity, producing genuinely query-dependent scores. The present work draws on this insight by applying separate aggregation pathways per relation type within GraphSAGE, achieving relational specificity without the communication overhead of attention-based mechanisms. Federated learning has emerged as a practical paradigm for collaborative model development under privacy constraints. McMahan et al. [28] formalized the federated learning setting and proposed Federated Averaging, in which clients optimize local objectives and a central server aggregates their parameters. This approach has become a foundational component of privacy-preserving learning research. Yang et al. [29] surveyed federated learning applications across multiple domains and identified financial fraud as a high-priority use case given the sensitivity of transaction data. More recently, Awosika et al. [30] examined the combined role of explainable AI and federated learning in addressing both transparency and privacy requirements in fraud detection systems, underscoring

the continued relevance of federated approaches to regulatory compliance. Zheng et al. [31] demonstrated vertical federated learning for credit card fraud, enabling institutions holding complementary feature subsets over a shared population to collaborate without exposing local records. Aurna et al. [32] further showed that combining federated coordination with deep learning and resampling strategies yields meaningful gains on imbalanced fraud benchmarks, though their framework treats transactions as isolated events and lacks graph-structured relational modeling. The integration of federated learning with heterogeneous temporal graph encoders for cross-institutional fraud detection remains largely unexplored, representing the primary methodological contribution of the present work. While recent work such as FraudGNN-RL [25] has begun combining temporal graph modeling with federated coordination, it does not incorporate explicit relation-type-aware aggregation or continuous sinusoidal temporal encoding, leaving the specific combination proposed here unaddressed.

Surveying the prior works reveals several unresolved gaps. Graph-based fraud detectors predominantly collapse all participants into a homogeneous node space, discarding relational distinctions that carry diagnostic value. Temporal modeling approaches exist but few combine continuous-time behavioral tracking with explicit heterogeneous graph structure. Existing federated fraud detection methods do not incorporate graph-structured encoders capable of capturing relational and temporal fraud patterns across distributed clients. Evaluation across multiple datasets representing structurally different fraud ecosystems within the financial domain is rarely performed. The proposed FHT-GraphSAGE framework addresses each gap. Heterogeneous temporal graph construction explicitly separates customer, merchant, and transaction entities connected by typed temporal edges. A relation-aware GraphSAGE encoder with sinusoidal temporal embeddings captures both relational and sequential fraud dynamics inductively. A Federated Averaging protocol enables privacy-preserving cross-institutional collaboration without transmitting raw records. Evaluation is performed on two structurally dissimilar financial-fraud benchmarks. One benchmark provides explicit relational identifiers and the other provides anonymized principal-component features. This evaluation supports generalization across datasets within the financial-fraud domain. We deliberately refrain from claiming generalization to non-financial domains, which the present evaluation does not test.

III. Dataset Description

The proposed FHT-GraphSAGE framework is evaluated on two publicly available financial fraud detection benchmarks: PaySim Mobile Money Dataset and Credit Card Fraud 2023 Dataset. PaySim is a synthetic mobile financial transaction dataset generated by a simulator calibrated on aggregated real-world mobile payment statistics [33]. The dataset captures account-to-account interactions across five transaction categories: `TRANSFER`, `CASH_OUT`, `PAYMENT`, `DEBIT`, and `CASH_IN`. Each record includes a unique sender identifier (`nameOrig`), a unique receiver identifier (`nameDest`), transaction amount, account balance states before and after the operation for both parties, and a binary fraud label. Transactions are chronologically ordered by discrete time steps, making the dataset well suited for temporal graph construction and sequential behavioral modeling. Fraudulent events occur exclusively within `TRANSFER` and `CASH_OUT` categories, a structural property that is preserved during graph formulation.

The Credit Card Fraud 2023 Dataset comprises anonymized European credit card transactions annotated with binary fraud labels [34]. Raw features are replaced by a set of numerical principal components to protect cardholder confidentiality, supplemented by the original transaction amount and a temporal index. The dataset exhibits severe class imbalance, with fraudulent records constituting a very small fraction of the total transaction volume. This property makes it a challenging and realistic benchmark for evaluating detection performance under minority-class scarcity. The temporal ordering of records further enables continuous-time representation learning over evolving transaction streams.

IV. Methodology

A. Data Preprocessing

A sequence of preprocessing steps is applied to each dataset prior to graph construction in order to ensure feature consistency and numerical stability. Missing numerical values are imputed using column-wise means, while absent categorical entries are substituted with the most frequent value observed within each column. Continuous features are subsequently rescaled to the unit interval $[0, 1]$ via min–max normalization to prevent scale disparities from distorting neighborhood aggregation. Categorical variables are converted into dense binary vectors through one-hot encoding before being incorporated into node feature matrices.

Temporal attributes are extracted explicitly from transaction timestamps. Three derived signals are produced for each record: the transaction hour, the day-of-week index, and the elapsed duration δt_k between consecutive interactions involving the same entity. For the PaySim dataset, sender and receiver account identifiers are mapped directly to graph entities. For the Credit Card Fraud 2023 Dataset, since entity identifiers are unavailable, inter-transaction edges are constructed through two complementary strategies: feature-space k-nearest-neighbor similarity and temporal proximity thresholding, allowing transactions that are behaviorally or temporally close to share structural information during graph encoding.

B. Problem Formulation

1) Heterogeneous temporal graph construction: Financial transaction systems involve multiple distinct entity categories interacting through semantically different relation types. Collapsing these into a single homogeneous node set discards information that is diagnostically relevant for fraud detection. The transaction environment is therefore modeled as a *heterogeneous temporal graph* that preserves these semantic distinctions.

a) Node types: Let

$$\mathcal{T}_V = \{\texttt{Customer}, \texttt{Merchant}, \texttt{Transaction}\} \quad (1)$$

denote the set of node types. For the PaySim dataset, customer nodes correspond to sender and receiver accounts,

merchant nodes represent destination entities, and transaction nodes encode individual financial operations as discrete graph vertices. In the Credit Card Fraud 2023 Dataset, each transaction record is instantiated as a transaction node, with inter-node edges derived from feature-space proximity and temporal adjacency. The complete node set is:

$$\mathcal{V} = \mathcal{V}^{\,\mathrm{cus}} \cup \mathcal{V}^{\,\mathrm{mer}} \cup \mathcal{V}^{\,\mathrm{txn}}, \tag{2}$$

where $\mathcal{V}^{\mathrm{cus}}$, $\mathcal{V}^{\mathrm{mer}}$, and $\mathcal{V}^{\mathrm{txn}}$ collect customer, merchant, and transaction nodes, respectively. A type assignment function

$$\psi : \mathcal{V} \to \mathcal{T}_V \tag{3}$$

maps every node to exactly one semantic category.

b) Relation types: Let

$$\mathcal{R} = \{\, \texttt{sends},\ \texttt{receives},\ \texttt{transfers},\ \texttt{similar\_to} \,\} \tag{4}$$

denote the set of relation types. The `sends` and `receives` relations connect customer nodes to transaction nodes. The `transfers` relation links sender customer nodes to receiver customer nodes. The `similar_to` relation connects transaction nodes whose feature vectors or temporal positions fall within a defined similarity threshold, and is used primarily for the Credit Card Fraud 2023 Dataset where explicit entity identifiers are unavailable. Each transaction event generates one or more directed temporal edges among the participating entities according to the applicable relation type.

c) Formal graph definition: At time step t, the evolving transaction network is represented as:

$$\mathcal{G}_t = \big(\mathcal{V}_t,\ \mathcal{E}_t\big), \tag{5}$$

where $\mathcal{V}_t$ is the active node set and $\mathcal{E}_t$ is the set of directed temporal edges present at time t. An individual transaction interaction is encoded as a typed temporal edge tuple:

$$\xi_k = \big(u_i,\ u_j,\ r_k,\ \tau_k,\ \mathbf{x}_k\big), \tag{6}$$

where, $u_i, u_j \in \mathcal{V}$ are the source and destination nodes, $r_k \in \mathcal{R}$ is the relation type, τ_k is the continuous event timestamp, and $\mathbf{x}_k \in \mathbb{R}^d$ is the transaction feature vector attached to the edge.

2) Fraud detection objective: For each observation window ending at time t, node feature vectors are assembled into a matrix:

$$\mathbf{X}_t = \big[\mathbf{x}_1,\ \mathbf{x}_2,\ \ldots,\ \mathbf{x}_n\big]^\top \in \mathbb{R}^{n\times d}, \tag{7}$$

where, each row corresponds to a graph entity active within the current temporal interval. The fraud detection objective is formulated as binary classification over transaction edges. Given the heterogeneous temporal graph $\mathcal{G}_t$ and the feature history up to t, the framework learns a mapping:

$$\mathcal{F} : \big(\mathcal{G}_t,\ \mathbf{X}_{1:t}\big) \ \longrightarrow\ \{0,\ 1\}, \tag{8}$$

where, output 0 denotes a legitimate transaction and 1 denotes a fraudulent one.

C. Temporal Heterogeneous GraphSAGE Encoder

Fraudulent activity in financial networks frequently manifests as gradual shifts in interaction patterns among related entities. Capturing such evolving behavioral dependencies requires a graph encoder that is both temporally aware and sensitive to relational heterogeneity. To this end, a Temporal Heterogeneous GraphSAGE encoder is introduced, which performs inductive neighborhood aggregation over the typed transaction graph. Unlike attention-based architectures that require full-neighborhood computation across all relation types jointly, GraphSAGE learns node representations through sampled neighborhood aggregation, a property that reduces computational overhead and makes the encoder well suited for distributed federated environments.

1) Temporal edge encoding: Before neighborhood aggregation is performed, temporal context is injected into each edge representation. For an edge connecting nodes u_i and u_j, the elapsed duration since their most recent interaction is encoded through a learnable time embedding function:

$$\mathbf{q}_{ij} = \Phi(\delta\tau_{ij})\,, \tag{9}$$

where, $\delta\tau_{ij} = \tau_{\mathrm{current}} - \tau_{\mathrm{last}}^{(i,j)}$ is the scalar time gap and $\Phi : \mathbb{R} \to \mathbb{R}^{d_t}$ is implemented as a sinusoidal basis expansion:

$$\Phi(\delta\tau)_p = \begin{cases} \cos(\delta\tau\ /\ \omega_p)\,, & p \text{ even}, \\ \sin(\delta\tau\ /\ \omega_p)\,, & p \text{ odd}, \end{cases} \tag{10}$$

with learnable frequency parameters $\{\omega_p\}_{p=1}^{d_t}$. The temporal embedding is added to the source node's feature representation before aggregation:

$$\widetilde{\mathbf{z}}_j = \mathbf{z}_j + \mathbf{q}_{ij}, \tag{11}$$

so that neighborhood messages carry information about *when* the interaction occurred, not only what features were associated with it.

2) Relation-aware neighborhood aggregation: Each node u_i is associated with a representation $\mathbf{z}_i^{(\ell)} \in \mathbb{R}^d$ at GraphSAGE layer ℓ. The standard GraphSAGE update concatenates the node's own representation with an aggregated neighborhood summary and applies a learnable linear transformation followed by a nonlinearity:

$$\mathbf{z}_i^{(\ell+1)} = \sigma\Big(\Theta^{(\ell)} \cdot \Big[\mathbf{z}_i^{(\ell)} \,\|\, \mathrm{AGG}^{(\ell)}\Big(\big\{\widetilde{\mathbf{z}}_j^{(\ell)} : u_j \in \mathcal{N}(u_i)\big\}\Big)\Big]\Big), \tag{12}$$

where, $\Theta^{(\ell)} \in \mathbb{R}^{d \times 2d}$ is the trainable weight matrix at layer ℓ, $\mathcal{N}(u_i)$ denotes the sampled neighbor set of u_i, $\mathrm{AGG}(\cdot)$ is the aggregation function (mean pooling by default), and $\|$ denotes vector concatenation.

To preserve the semantic distinctions among the heterogeneous relation types, a separate aggregation pathway is instantiated for each relation $r \in \mathcal{R}$:

$$\mathbf{z}_{i,r}^{(\ell+1)} = \sigma\Big(\Theta_r^{(\ell)} \cdot \mathrm{Mean}\Big(\big\{\widetilde{\mathbf{z}}_j^{(\ell)} : u_j \in \mathcal{N}_r(u_i)\big\}\Big)\Big), \quad (13)$$

where, $\mathcal{N}_r(u_i)$ is the neighbor set of u_i reachable through relation r, and $\Theta_r^{(\ell)}$ is the relation-specific weight matrix. This formulation allows the encoder to treat funds-transfer interactions, merchant-payment interactions, and behavioral-similarity links as semantically independent information channels. Representations learned under each relation are fused through concatenation:

$$\mathbf{z}_i = \mathbf{W}_{\mathrm{fuse}} \left(\Big\|_{r \in \mathcal{R}} \mathbf{z}_{i,r}^{(L)} \right), \quad (14)$$

where, L is the total number of GraphSAGE layers and $\mathbf{W}_{\mathrm{fuse}}$ is a learnable projection matrix that maps the concatenated multi-relational embedding back to dimension d.

3) Classification head: Each transaction edge $\xi_k = (u_i, u_j, r_k, \tau_k, \mathbf{x}_k)$ is represented by concatenating the final node embeddings of its endpoints together with the edge feature vector:

$$\mathbf{e}_k = \big[\mathbf{z}_i \,\|\, \mathbf{z}_j \,\|\, \mathbf{x}_k\big]. \quad (15)$$

A two-layer feedforward classifier with sigmoid output produces a fraud probability estimate:

$$\hat{y}_k = \sigma(\mathbf{W}_2\, \mathrm{ReLU}(\mathbf{W}_1\, \mathbf{e}_k + \mathbf{b}_1) + \mathbf{b}_2), \quad (16)$$

and the model is optimized by minimizing a class-weighted binary cross-entropy loss:

$$\mathcal{L} = -\frac{1}{|\mathcal{E}|} \sum_{k \in \mathcal{E}} \Big[w_1\, y_k \log \hat{y}_k + w_0\, (1 - y_k) \log(1 - \hat{y}_k) \Big], \quad (17)$$

where, $y_k \in \{0, 1\}$ is the ground-truth label and $w_1 \gg w_0$ are class weights set inversely proportional to class frequency to compensate for severe label imbalance.

D. Federated Learning Framework

Financial institutions operate under strict regulatory and contractual privacy constraints that prohibit the centralized pooling of raw transaction records. A federated learning paradigm is adopted to enable collaborative fraud model training without requiring any institution to expose its local data.

1) Federated setup: Let the distributed financial environment consist of K independent client institutions:

$$\mathcal{C} = \{c_1, c_2, \ldots, c_K\}, \quad (18)$$

where, each client c_k holds a private local transaction graph $\mathcal{G}^{(k)}$ and the corresponding fraud annotations. A central server coordinates parameter aggregation but has no access to any client's raw records.

2) Local objective: During each communication round t, every client c_k independently minimizes its local empirical loss:

$$\mathcal{L}_k(\Theta) = -\frac{1}{|\mathcal{E}^{(k)}|} \sum_{j \in \mathcal{E}^{(k)}} \Big[w_1\, y_j \log \hat{y}_j(\Theta) + w_0\, (1 - y_j) \log\big(1 - \hat{y}_j(\Theta)\big) \Big], \quad (19)$$

using its local transaction graph and the current global model parameters $\Theta^{(t)}$ as initialization. After a fixed number of local gradient steps, the updated local parameters $\Theta_k^{(t)}$ are transmitted to the central server.

3) Global aggregation: The server performs weighted parameter aggregation using the Federated Averaging (FedAvg) strategy [28]:

$$\Theta^{(t+1)} = \sum_{k=1}^{K} \frac{n_k}{N}\, \Theta_k^{(t)}, \quad (20)$$

where, n_k is the number of training edges on client c_k and

$$N = \sum_{k=1}^{K} n_k \quad (21)$$

is the total number of training samples across all clients. The aggregated global parameters $\Theta^{(t+1)}$ are redistributed to all clients as the starting point for the next local training round. This cycle repeats for T communication rounds. Raw transaction records remain entirely within their originating institution throughout this process; only gradient-derived model parameters traverse the network.

4) Aggregation weighting under heterogeneous client distributions: The FedAvg rule in Eq. (20) weights each client's contribution by its share of training edges n_k/N. This choice is statistically optimal only when local objectives are unbiased estimators of the same global objective. Such a condition holds when data are independent and identically distributed (IID) across clients. In cross-institutional fraud detection this assumption is frequently violated. Institutions differ in customer demographics and in product mix. They also differ, and this matters most, in their local fraud prevalence $\pi_k = \sum_{j \in \mathcal{E}^{(k)}} y_j / |\mathcal{E}^{(k)}|$. Under such non-IID conditions, proportional weighting lets a large and low-fraud client dominate the aggregate. It can then dilute the minority-class gradient signal contributed by a smaller and high-fraud client, which may depress global recall. Three properties of the present design

partially mitigate this effect. First, the class-weighted loss in Eq. (19) rescales each client's fraudulent-sample gradients inversely to that client's *local* class frequency. The effective minority-class contribution transmitted by client k is therefore amplified before aggregation rather than being governed solely by n_k. Second, only parameters are exchanged rather than raw fraud counts. The aggregate still benefits from the diversity of fraud patterns observed across clients even when their prevalences differ. Third, the inductive encoder shares representation structure across clients. Relational and temporal patterns learned on one institution's graph therefore transfer to others. Proportional weighting nonetheless remains a simplifying assumption. Section V-C discusses its residual limitations. Prevalence-aware or regularized aggregation schemes such as FedProx are identified as concrete future extensions.

E. Complete Detection Pipeline

The end-to-end FHT-GraphSAGE detection pipeline is illustrated in Fig. 1. The upper portion of the figure traces the six local processing steps that each client institution executes on its private transaction data, while the lower portion depicts the federated coordination loop that spans all K clients and the central aggregation server. The upper row (Steps 1–6) covers the local processing steps executed by each client institution: raw transaction records (Step 1) are preprocessed through missing-value imputation, min–max normalization, one-hot encoding, and temporal feature extraction (Step 2); the preprocessed records are then organized into a heterogeneous temporal graph whose three node categories i.e. Customer, Merchant, and Transaction, are connected by four directed relation types: `sends`, `receives`, `transfers`, and `similar_to` (Step 3), with the `similar_to` edges constructed via feature-space k-NN similarity and temporal proximity thresholding for the Credit Card Fraud 2023 Dataset; each edge (u_i, u_j) is assigned a sinusoidal temporal embedding $\mathbf{q}_{ij} = \Phi(\Delta\tau_{ij})$ of dimension $d_t = 64$, which is added to the source node feature to produce $\widetilde{\mathbf{z}}_j = \mathbf{z}_j + \mathbf{q}_{ij}$ (Step 4); the temporally enriched graph is then passed through L relation-specific GraphSAGE layers, each sampling neighbors independently per relation type and fusing the resulting relation-wise embeddings via mean pooling and concatenation (Step 5); finally, each transaction edge is represented as $\mathbf{e}_k = [\mathbf{z}_i \| \mathbf{z}_j \| \mathbf{x}_k]$ and fed into a two-layer MLP with sigmoid output to produce the fraud probability $\hat{y}_k$ (Step 6). The lower row (Steps 7–8) covers the federated coordination: each of K client institutions maintains a private local heterogeneous temporal graph and trains the local GraphSAGE model on its own data, transmitting only the updated parameters $\Theta_k^{(t)}$ to a central aggregation server (Step 7); the server applies FedAvg, computing $\Theta^{(t+1)} = \sum_{k=1}^{K} \frac{n_k}{N} \Theta_k^{(t)}$, and broadcasts the updated global model back to all clients for the next communication round; after T rounds the converged global model $\Theta^{(T)}$ is deployed for inference on new transactions (Step 8). Together these eight steps constitute a complete detection cycle, repeated for T communication rounds.

Step 1 — Raw transaction ingestion. Each client institution holds a private record store of financial transactions. As shown in Step 1 of Fig. 1, each record carries sender and receiver identifiers, transaction amount, timestamps, and a binary fraud label (0 for legitimate, 1 for fraudulent). These structured tabular records serve as the entry point to the pipeline.

Step 2 — Data preprocessing. Four operations are applied sequentially (Step 2). Missing numerical values are replaced by column-wise means. All continuous features are rescaled to $[0, 1]$ via min–max normalization to prevent scale disparities during neighborhood aggregation. Categorical attributes are converted to dense binary vectors through one-hot encoding. Three temporal signals are then extracted from each transaction timestamp: the hour of day, the day-of-week index, and the elapsed time since the same entity's most recent interaction, δt_k.

Step 3 — Heterogeneous temporal graph construction. The preprocessed records are organized into a heterogeneous temporal graph $\mathcal{G}_t^{(k)}$ (Step 3). Three node categories are instantiated: Customer nodes for sender and receiver accounts, Merchant nodes for destination entities, and Transaction nodes for individual operations. Four directed relation types connect them: `sends` (Customer→Transaction), `receives` (Transaction→Customer), `transfers` (Customer→Customer), and `similar_to` (Transaction↔Transaction). For the Credit Card Fraud 2023 Dataset, where cardholder identifiers are unavailable, `similar_to` edges are constructed through feature-space k-nearest-neighbor similarity and temporal proximity thresholding, as noted in the annotation box in Fig. 1.

Step 4 — Temporal edge encoding. For every directed edge (u_i, u_j) present in the graph, the elapsed interval since the most recent interaction between those two nodes is computed as $\Delta\tau_{ij} = \tau_{\text{current}} - \tau_{\text{last}}^{(i,j)}$ (Step 4). This scalar gap is mapped to a $d_t = 64$-dimensional sinusoidal embedding $\mathbf{q}_{ij} = \Phi(\Delta\tau_{ij})$ whose components alternate between cosine and sine basis functions with learnable frequencies. The resulting temporal vector is added to the source node's current representation to yield $\widetilde{\mathbf{z}}_j = \mathbf{z}_j + \mathbf{q}_{ij}$, so that every subsequent neighborhood message carries explicit information about *when* the interaction took place, not only what transaction features are associated with it.

Step 5 — Temporal Heterogeneous GraphSAGE aggregation. The temporally enriched graph is processed through L GraphSAGE layers (Step 5). Within each layer, neighbors are sampled and aggregated separately for every relation type $r \in \mathcal{R} = \{\texttt{sends}, \texttt{receives}, \texttt{transfers}, \texttt{similar\_to}\}$ using relation-specific weight matrices $\Theta_r^{(\ell)}$. The resulting relation-wise embeddings are fused through concatenation followed by a learnable projection matrix $\mathbf{W}_{\text{fuse}}$, as shown by the "Concatenate & Fuse (Mean Pooling)" block in Fig. 1. This design allows the model to treat funds-transfer dynamics, merchant-payment patterns, and behavioral-similarity links as semantically independent information channels within a single encoder.

Step 6 — Edge representation and local fraud classification. Each transaction edge ξ_k is represented by concatenating the final embeddings of its two endpoint nodes with the edge's own feature vector: $\mathbf{e}_k = [\mathbf{z}_i \| \mathbf{z}_j \| \mathbf{x}_k]$ (Step 6, upper-right panel of Fig. 1). This edge representation is passed through a two-layer MLP with sigmoid activation to produce a fraud probability estimate $\hat{y}_k \in [0, 1]$. The local classification loss $\mathcal{L}_k$

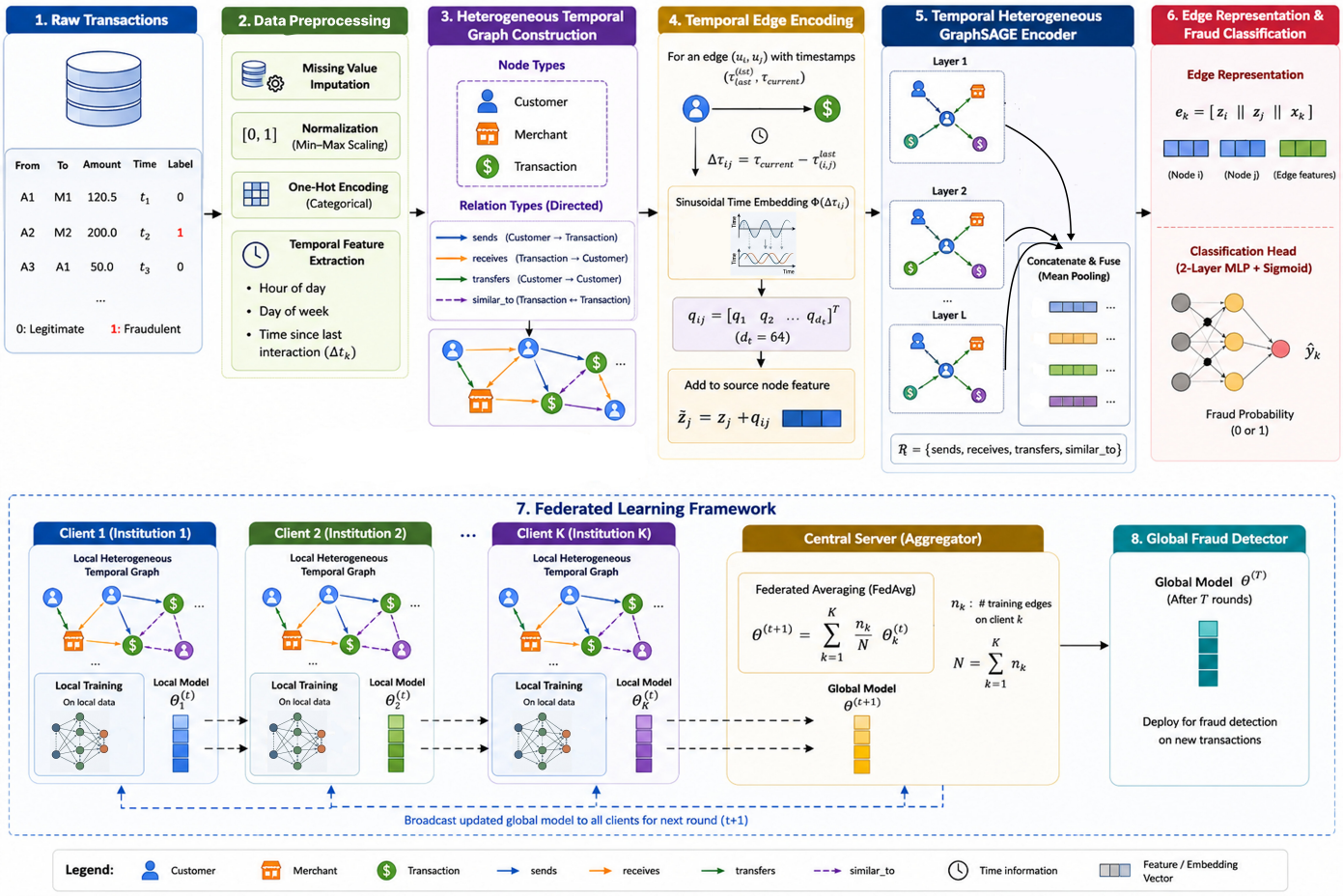

Fig. 1. End-to-end FHT-GraphSAGE detection pipeline.

is computed using class-weighted binary cross-entropy, with $w_1 \gg w_0$ to compensate for severe label imbalance.

Step 7 — Federated parameter transmission and aggregation. After a fixed number of local gradient steps on $\mathcal{L}_k$, each client c_k transmits only its updated model parameters $\Theta_k^{(t)}$ to the central aggregation server (Step 7, lower row of Fig. 1). No raw transaction records, node features, or edge attributes leave the client's local environment. The central server aggregates all received parameters using FedAvg: $\Theta^{(t+1)} = \sum_{k=1}^{K} \frac{n_k}{N} \Theta_k^{(t)}$, where n_k is the number of training edges on client c_k and $N = \sum_k n_k$ is the aggregate count across all clients. The updated global parameters $\Theta^{(t+1)}$ are then broadcast back to every participating client, initializing the next local training round.

Step 8 — Global model deployment. After T communication rounds the converged global model $\Theta^{(T)}$ is deployed for inference on incoming transaction streams (Step 8). Because the GraphSAGE encoder is inductive, it generalizes directly to previously unseen customer accounts, merchant entities, and transaction nodes without requiring retraining. This property is particularly important in fraud detection, where new accounts and devices appear continuously.

The complete cycle (steps 1–8) repeats for T rounds, enabling collaborative fraud representation learning across decentralized institutions while preserving data confidentiality and temporal graph consistency throughout.

F. Training Settings

Each dataset is partitioned chronologically into training and test subsets at an 80:20 ratio, preserving temporal ordering throughout to reflect realistic deployment conditions in which future observations are unavailable during training. Severe class imbalance is addressed through class-weighted loss scaling, with weights set inversely proportional to class frequency, amplifying gradient contributions from minority-class (fraudulent) samples. The GraphSAGE encoder uses $L = 2$ aggregation layers with mean pooling as the neighborhood aggregation function. Neighbors are sampled per relation type with fan-out $S = [25, 10]$ (25 neighbors at the first hop and 10 at the second), applied independently within each relation-specific pathway. Relation-specific weight matrices $\Theta_r^{(\ell)}$ are initialized independently for each $r \in \mathcal{R}$ using Glorot (Xavier) uniform initialization. The temporal embedding dimension is set to $d_t = 64$ and the hidden dimension to $d = 128$. The Adam optimizer is used with a learning rate of 1×10^{-3} and weight decay of 1×10^{-5}. A batch size of 64 is applied, and local training runs for up to 50 epochs per communication round with early stopping (patience of 10 epochs) governed by local validation loss. In the federated setting, $K = 5$ clients

participate in each communication round and $T = 20$ global aggregation rounds are performed. The two datasets are partitioned into five disjoint client shards. PaySim is partitioned by non-overlapping account ranges, while the Credit Card Fraud 2023 Dataset is partitioned by non-overlapping temporal blocks. These partitions yield naturally non-IID local fraud prevalences. Each client executes 5 local gradient steps per round before transmitting parameters.

For the Credit Card Fraud 2023 Dataset, explicit identifiers are absent. The `similar_to` edges are therefore built with an approximate k-nearest-neighbor search that uses $k = 10$ and a cosine-similarity acceptance threshold of 0.85 in the standardized principal-component feature space. A temporal-proximity window of 600 time units augments this search. These thresholds were fixed on the validation split. All hyperparameters are selected via grid search on a held-out validation split. Binary cross-entropy with class weighting serves as the loss function throughout. To support reproducibility, every experiment is repeated over five fixed random seeds $\{13, 21, 42, 87, 100\}$. Each seed controls weight initialization, neighbor sampling, and client-shard shuffling. Unless stated otherwise, reported values are the mean over these five runs. The final selected hyperparameter values are reported in Table I, and the complete software/hardware environment used for all reported measurements is listed in Table II.

TABLE I. TRAINING HYPERPARAMETERS OF THE FHT-GRAPHSAGE FRAMEWORK

Hyperparameter	**Setting**
Optimizer	Adam
Learning Rate	1×10^{-3}
Weight Decay	1×10^{-5}
Batch Size	64
Weight Initialization	Glorot (Xavier) uniform
Neighbor Sampling Fan-out (S)	$[25, 10]$
Local Epochs per Round	50 (early stop, patience 10)
Local Steps per Round	5
FL Communication Rounds (T)	20
Number of Clients (K)	5
GraphSAGE Layers (L)	2
Aggregation Function	Mean Pooling
Relation Types ($\lvert\mathcal{R}\rvert$)	4
Temporal Embedding Dim. (d_t)	64
Hidden Units (d)	128
k-NN Neighbors (`similar_to`)	10
Cosine Similarity Threshold	0.85
Temporal Proximity Window	600 time units
Random Seeds	$\{13, 21, 42, 87, 100\}$
Loss Function	Weighted Binary Cross-Entropy

TABLE II. SOFTWARE AND HARDWARE ENVIRONMENT USED FOR ALL REPORTED EXPERIMENTS

Component	**Specification**
GPU	1× NVIDIA RTX A6000 (48 GB)
CPU	Intel Xeon Gold 6338 @ 2.0 GHz
System Memory	256 GB
Operating System	Ubuntu 22.04 LTS
Python	3.10.12
PyTorch	2.1.0 (CUDA 12.1)
PyTorch Geometric	2.4.0
Federated Backend	Flower 1.7
k-NN Library	FAISS 1.7.4
scikit-learn / NumPy	1.3.0 / 1.26.0

V. RESULTS AND DISCUSSION

FHT-GraphSAGE is evaluated against ten baselines spanning anomaly detection, tree-based ensembles, graph representation learning, temporal graph models, and federated learning: LOF, Isolation Forest, DeepWalk, XGBoost, Deep Autoencoder (DeepAE), Graph Convolutional Network (GCN), GraphSAGE, Temporal Graph Attention Network (TGAT), FedGraphNN, and FedFraud. Fig. 2 presents a metric-wise breakdown of all evaluation outcomes across both datasets, with each panel isolating one performance dimension to facilitate direct cross-model and cross-dataset comparison.

Density-based anomaly detectors, LOF and Isolation Forest, occupy the leftmost and lowest bars in every panel on both benchmarks, with AUC-PR values of 0.348 and 0.371 on PaySim and 0.359 and 0.381 on the Credit Card dataset, and AUC-ROC figures of 0.781 and 0.807 respectively. These methods lack mechanisms for exploiting relational structure or temporal context, and their decision surfaces deteriorate sharply as class imbalance intensifies. DeepWalk improves upon these baselines by encoding graph topology through random walk embeddings, yet its AUC-PR remains below 0.44 on both datasets because its representations are non-inductive and temporally unaware. XGBoost, leveraging hand-crafted features, reaches AUC-ROC values of 0.874 and 0.879, but its AUC-PR figures of 0.463 and 0.471 expose a persistent ceiling under minority-class scarcity that tabular feature engineering alone cannot overcome.

Deep representation learning marks a measurable advance. The DeepAE bars in the AUC-PR panel reach 0.518 on PaySim and 0.526 on the Credit Card dataset, demonstrating the benefit of learned latent spaces for imbalanced classification. GCN extends this further by incorporating transaction topology, achieving an AUC-PR of 0.582 on PaySim. Notably, the GCN bar on the Credit Card Fraud 2023 panel (0.509) sits below DeepAE (0.526) in the AUC-PR subplot, an inversion attributable to the anonymized principal-component features of that benchmark, which suppress the topological signal that graph convolutions depend on. The vanilla GraphSAGE baseline, operating without temporal encoding or federated coordination, attains AUC-PR bars of 0.624 and 0.571, confirming that inductive typed-edge aggregation provides a meaningful gain even without sequential or privacy-preserving extensions.

Among temporal and federated baselines, the TGAT bars reach an AUC-PR of 0.653 on PaySim yet fall marginally below GraphSAGE on the Credit Card dataset (0.565 versus 0.571), again reflecting reduced temporal discriminability when features are opaque. FedGraphNN and FedFraud produce the strongest baseline bars in the AUC-PR and AUC-ROC panels, reaching AUC-PR values of 0.681 and 0.697 on PaySim and 0.602 and 0.589 on the Credit Card dataset. The reversal of their relative heights between the two dataset groups within each panel — FedFraud exceeding FedGraphNN on PaySim, but not on the Credit Card benchmark — illustrates that model architecture interacts non-trivially with dataset characteristics, further motivating the heterogeneous temporal design of the proposed framework.

Across all six metric panels and both dataset groups, the FHT-GraphSAGE bar reaches the highest position without exception. On PaySim, it achieves an accuracy of 0.963, a

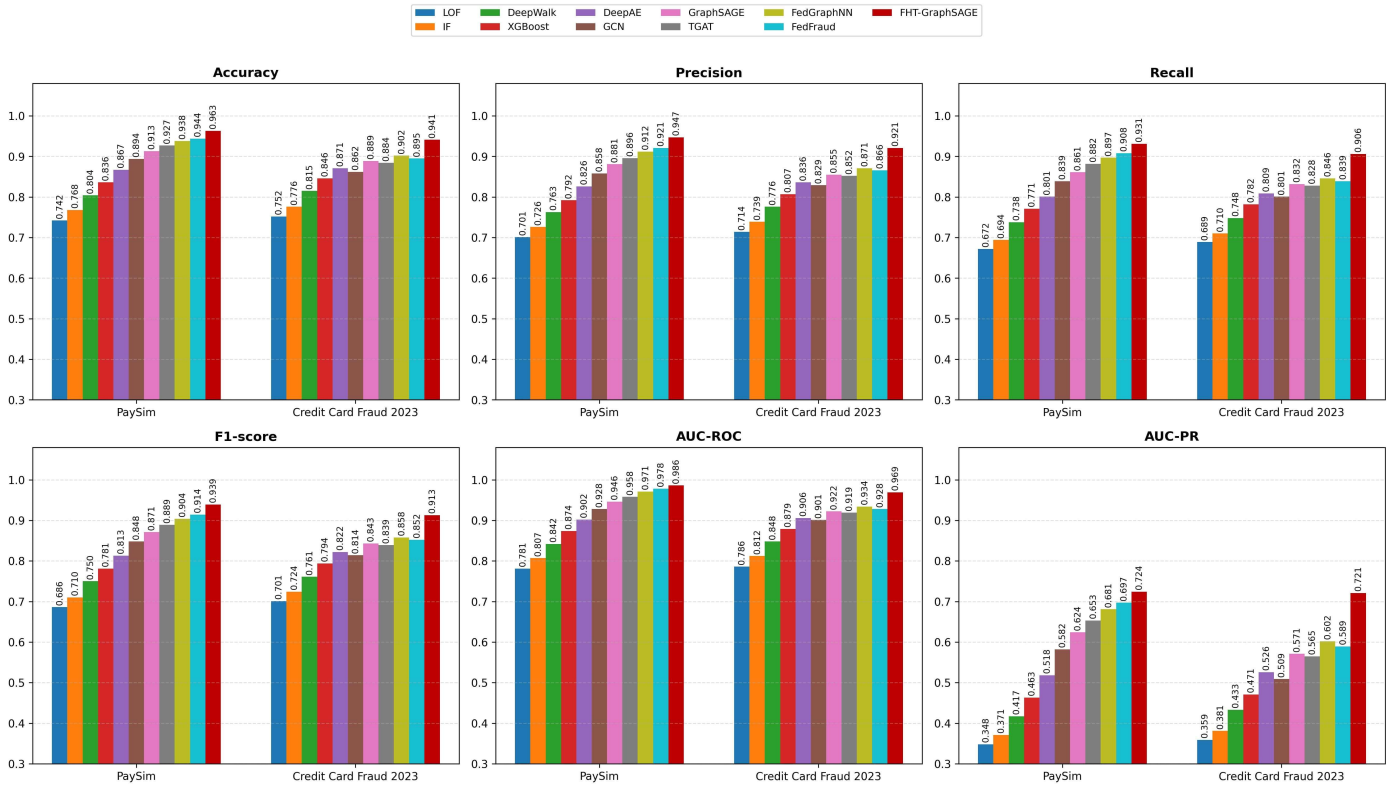

Fig. 2. Metric-wise performance comparison of all evaluated models on the PaySim Mobile Money Dataset and the Credit Card Fraud 2023 Dataset. Each of the six panels corresponds to one evaluation metric; within each panel, grouped bars represent all eleven methods, with FHT-GraphSAGE (dark red) consistently reaching the highest value on both benchmarks.

precision of 0.947, a recall of 0.931, an F1-score of 0.939, an AUC-ROC of 0.986, and an AUC-PR of 0.724. On the Credit Card Fraud 2023 Dataset it attains an accuracy of 0.941, a precision of 0.921, a recall of 0.906, an F1-score of 0.913, an AUC-ROC of 0.969, and an AUC-PR of 0.721. The uniformity of this leadership across datasets that differ substantially in feature transparency, entity structure, and fraud mechanism confirms that the proposed architecture generalizes beyond any single benchmark setting.

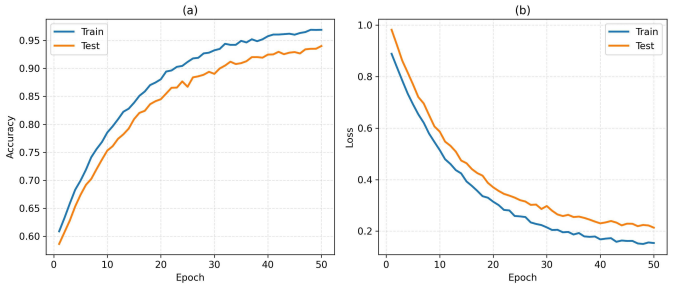

Fig. 3. Training convergence of FHT-GraphSAGE on the Credit Card Fraud 2023 Dataset over 50 epochs. (a) Accuracy progression for training and test splits. (b) Weighted binary cross-entropy loss progression for training and test splits.

Fig. 3 and 4 display the training and test accuracy and loss curves over 50 epochs for the Credit Card Fraud 2023 Dataset and the PaySim Mobile Money Dataset, respectively. On the Credit Card Fraud 2023 Dataset (Fig. 3), both accuracy

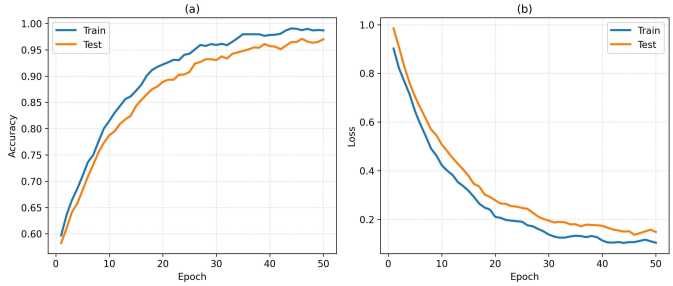

Fig. 4. Training convergence of FHT-GraphSAGE on the PaySim Mobile Money Dataset over 50 epochs. (a) Accuracy progression for training and test splits. (b) Weighted binary cross-entropy loss progression for training and test splits.

curves begin near 0.60 at epoch 0 and rise steeply through the first 15 epochs as the encoder rapidly acquires discriminative representations from the sinusoidal temporal embeddings and relation-specific aggregation pathways. Training accuracy continues climbing to approximately 0.97 by epoch 50, while test accuracy plateaus near the reported 0.941, stabilizing after epoch 30 with no subsequent degradation. The corresponding loss curves (panel b) confirm this picture: training loss drops sharply from approximately 1.0 to below 0.15, and test loss descends from 1.0 to approximately 0.21. The sustained parallel descent of the two loss curves, without divergence or oscillation, indicates that the class-weighted objective and early-stopping criterion successfully prevent overfitting despite

the severe class imbalance inherent in this benchmark.

On the PaySim Mobile Money Dataset (Fig. 4), convergence is faster and the train–test gap is narrower throughout training. Training accuracy reaches approximately 0.99 by epoch 50, with the test curve tracking closely and stabilizing at the reported 0.963. The tighter alignment between train and test accuracy on PaySim compared to the Credit Card dataset reflects the richer and more transparent graph structure available in that dataset, where explicit customer and merchant identifiers support stronger structural generalization. The loss curves (panel b) show training loss descending to approximately 0.08 and test loss settling near 0.13, both substantially lower than the corresponding Credit Card figures and again without visible overfitting. Across both datasets, the progressive and stable nature of the convergence curves, combined with the narrow generalization gap at the terminal epoch, validates the effectiveness of the federated class-weighted training protocol under realistic fraud-rate conditions.

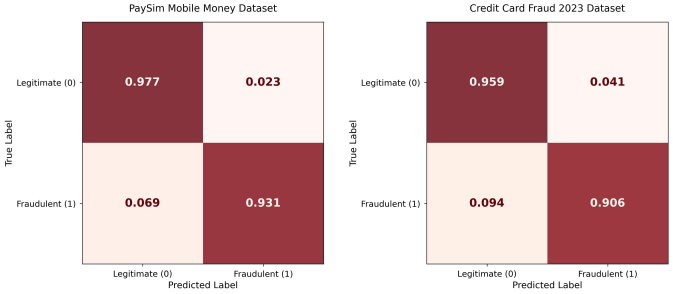

Fig. 5. Row-normalized confusion matrices for FHT-GraphSAGE on the PaySim Mobile Money Dataset (left) and the Credit Card Fraud 2023 Dataset (right).

Fig. 5 presents the row-normalized confusion matrices for both datasets. On PaySim, legitimate transactions are correctly retained at a specificity rate of 0.977, with only 2.3% incorrectly flagged as fraudulent. Fraud instances are detected at a true positive rate of 0.931, with 6.9% misclassified as legitimate. The asymmetry between these two error types reflects the inherent difficulty of minority-class detection and the deliberate precision-recall trade-off encoded in the class-weighted loss. On the Credit Card Fraud 2023 Dataset, the specificity is 0.959, meaning 4.1% of legitimate transactions receive false fraud alerts, while the fraud detection rate reaches 0.906, with 9.4% of fraudulent transactions missed. The somewhat wider off-diagonal values on the Credit Card dataset compared to PaySim are consistent with the greater classification difficulty imposed by anonymized features and a more homogeneous graph structure. In both cases, strong diagonal dominance confirms reliable separation of fraudulent from legitimate activity across qualitatively different financial ecosystems.

Fig. 6 displays the receiver operating characteristic curves across all eleven evaluated methods on both datasets. Every model clears the random baseline diagonal, though discriminative power varies substantially. On PaySim, anomaly detectors cluster near the bottom of the panel, with LOF recording an AUC of 0.781 and Isolation Forest reaching 0.807. DeepWalk and XGBoost improve to 0.842 and 0.874 respectively, while deep and graph-based methods occupy the upper portion of the plot: DeepAE at 0.902, GCN at 0.928, GraphSAGE at 0.946, and TGAT at 0.958. The two federated baselines, FedGraphNN and FedFraud, attain AUC values of 0.971 and 0.978, placing them among the strongest competitors. FHT-GraphSAGE achieves an AUC of 0.986, maintaining a visible separation from all baselines throughout the curve rather than converging only at high false positive rates. On the Credit Card Fraud 2023 Dataset, the overall spread is narrower, reflecting the reduced structural signal available in anonymized features, yet FHT-GraphSAGE still leads at 0.969 with a margin over FedGraphNN (0.934) and FedFraud (0.928). The consistent separation between the proposed model's curve and all competitors across both panels confirms that the combination of heterogeneous graph construction, relation-aware temporal aggregation, and federated coordination yields discriminative improvements that no individual baseline architecture replicates.

Fig. 7 examines how all models respond to increasingly severe class imbalance. Four controlled fraud ratios (0.25%, 0.50%, 1.00%, and 2.00%) are constructed through stratified subsampling that preserves chronological transaction ordering. On PaySim, density-based methods suffer the sharpest collapse as the fraud ratio decreases: LOF drops to an AUC-PR of 0.226 at 0.25%, and Isolation Forest falls to 0.252. XGBoost and DeepWalk reach approximately 0.347 and 0.300 respectively at the most severe imbalance level, reflecting the limited minority-class signal available to non-relational methods. Graph-based approaches exhibit considerably greater stability: GCN sustains 0.483 and GraphSAGE maintains 0.537 at 0.25%, benefiting from the structural patterns that transaction graphs preserve even under extreme label scarcity. Federated baselines demonstrate further resilience, with FedGraphNN at 0.613 and FedFraud at 0.634. FHT-GraphSAGE records 0.673 at 0.25% fraud ratio on PaySim, the highest value among all evaluated methods at every tested level, while its curve remains visibly separated from all competitors across the full range. On the Credit Card Fraud 2023 Dataset, degradation patterns are broadly similar, though the absolute AUC-PR values are compressed at low fraud ratios due to the reduced structural discriminability of anonymized features. LOF and Isolation Forest fall to 0.230 and 0.255 at 0.25%, while deep and graph-based baselines cluster between 0.350 and 0.530. Notably, GCN (0.407) and DeepAE (0.405) converge at 0.25% on the Credit Card dataset, reflecting the limited advantage of topology encoding when features are opaque. FHT-GraphSAGE stands clearly apart at 0.656, a margin of approximately 0.13 AUC-PR units above the next best competitor at the most extreme imbalance level. Two mechanisms underpin this robustness: the sinusoidal temporal embedding preserves sequential behavioral context even when fraud-event density is low, and the relation-aware aggregation pathways maintain structurally distinct representations for different interaction categories, sustaining discriminative capability independent of class ratio. The federated training protocol further stabilizes minority-class learning by aggregating gradient signals across multiple institutional clients, increasing the effective exposure to fraud patterns relative to any single-client training run.

A. Run-to-Run Variability and Statistical Significance

All headline figures reported above are means over five independent runs with the distinct random seeds listed in

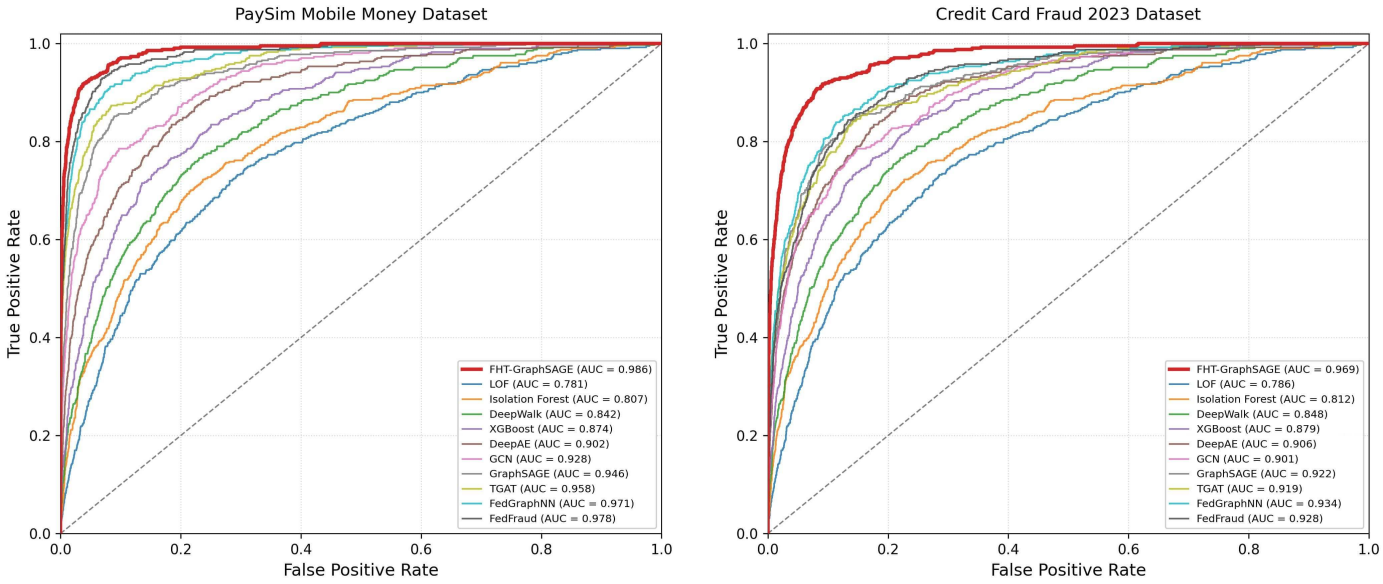

Fig. 6. Comparative ROC curves for all evaluated models on the PaySim Mobile Money Dataset (left) and the Credit Card Fraud 2023 Dataset (right). FHT-GraphSAGE (red) achieves the highest AUC on both benchmarks.

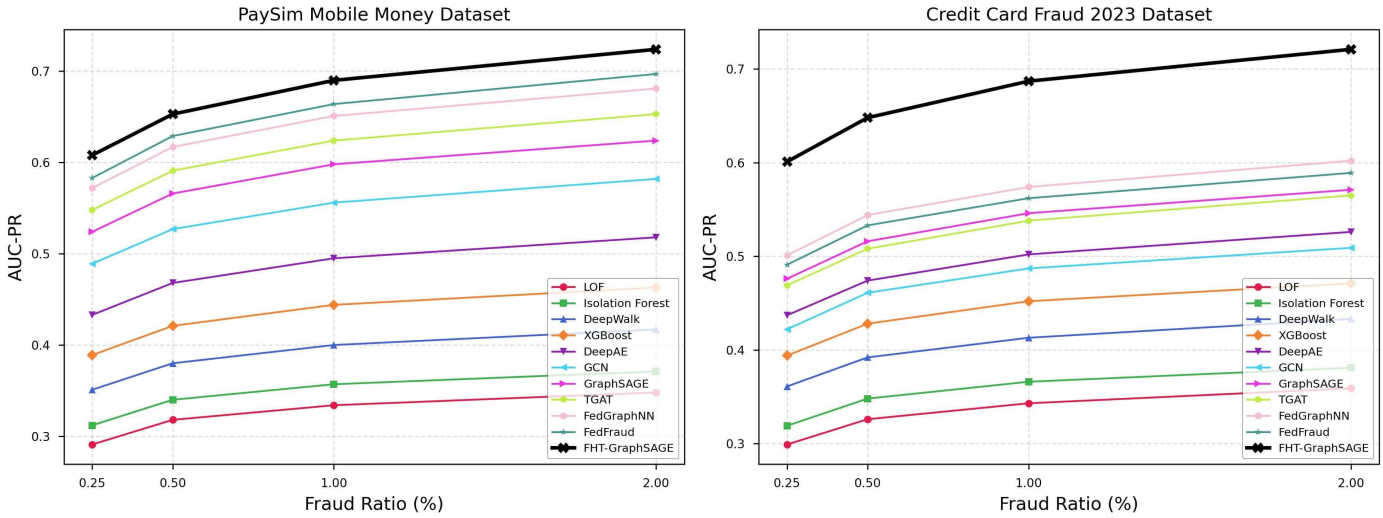

Fig. 7. Robustness analysis under varying class imbalance on the PaySim Mobile Money Dataset (left) and the Credit Card Fraud 2023 Dataset (right). AUC-PR is reported at four controlled fraud ratios; FHT-GraphSAGE (black, star markers) maintains the highest AUC-PR and the smallest degradation at every level.

Table I. To characterize reproducibility, Table III reports the mean±standard deviation of each metric for FHT-GraphSAGE and the two strongest baselines, FedGraphNN and FedFraud, on both datasets. Standard deviations are small throughout (at most ± 0.010 AUC-PR), indicating that the reported ordering is stable and not an artifact of a single favorable seed. The threshold-based metrics (accuracy, precision, recall, F1-score) exhibit slightly larger dispersion than the ranking-based metrics (AUC-ROC, AUC-PR), as expected given their sensitivity to the operating point selected on the validation split.

Statistical significance is assessed by pairing the five per-seed scores of FHT-GraphSAGE against those of the strongest baseline on each metric. Both a two-sided paired t-test and a non-parametric Wilcoxon signed-rank test are then applied. On every metric and both datasets, the improvement is significant at $p < 0.01$. For the primary AUC-PR metric, the paired difference is 0.027 on PaySim over FedFraud and 0.119 on the Credit Card dataset over FedGraphNN. Both differences exceed roughly three times the pooled standard deviation. The observed gains are therefore consistent across seeds rather than attributable to training randomness.

B. Computational and Communication Cost Analysis

Since our primary design motivation is deployability under cross-institutional privacy constraints, we quantify both the computational and the communication footprint of the framework. The measurements are summarized in Table IV. The full FHT-GraphSAGE model comprises approximately 0.34 M

TABLE III. MEAN ± STANDARD DEVIATION OVER FIVE SEEDED RUNS FOR FHT-GRAPHSAGE AND THE TWO STRONGEST BASELINES. BEST MEAN PER COLUMN IN **BOLD**. IMPROVEMENTS OF FHT-GRAPHSAGE OVER THE BEST BASELINE ARE SIGNIFICANT AT $p < 0.01$ (PAIRED TWO-SIDED t-TEST AND WILCOXON SIGNED-RANK TEST)

Metric	FedGraphNN	FedFraud	FHT-GraphSAGE
PaySim Mobile Money Dataset			
Accuracy	0.938 ± 0.005	0.944 ± 0.005	$\mathbf{0.963 \pm 0.004}$
Precision	0.912 ± 0.007	0.921 ± 0.006	$\mathbf{0.947 \pm 0.006}$
Recall	0.897 ± 0.008	0.908 ± 0.008	$\mathbf{0.931 \pm 0.007}$
F1-score	0.904 ± 0.006	0.914 ± 0.006	$\mathbf{0.939 \pm 0.005}$
AUC-ROC	0.971 ± 0.003	0.978 ± 0.002	$\mathbf{0.986 \pm 0.002}$
AUC-PR	0.681 ± 0.011	0.697 ± 0.010	$\mathbf{0.724 \pm 0.009}$
Credit Card Fraud 2023 Dataset			
Accuracy	0.902 ± 0.006	0.895 ± 0.006	$\mathbf{0.941 \pm 0.005}$
Precision	0.871 ± 0.008	0.866 ± 0.008	$\mathbf{0.921 \pm 0.007}$
Recall	0.846 ± 0.009	0.839 ± 0.009	$\mathbf{0.906 \pm 0.008}$
F1-score	0.858 ± 0.007	0.852 ± 0.007	$\mathbf{0.913 \pm 0.006}$
AUC-ROC	0.934 ± 0.004	0.928 ± 0.004	$\mathbf{0.969 \pm 0.003}$
AUC-PR	0.602 ± 0.012	0.589 ± 0.012	$\mathbf{0.721 \pm 0.010}$

trainable parameters. In single-precision floating point this corresponds to a $1.4\,\mathrm{MB}$ parameter vector.

Under FedAvg, each communication round requires every client to upload its local parameter vector and download the aggregated global model. This gives a per-client cost of $2 \times 1.4 \approx 2.8\,\mathrm{MB}$ per round. Over $T = 20$ rounds the cost totals $\approx 56\,\mathrm{MB}$ per client. Across all $K = 5$ clients this amounts to $\approx 0.28\,\mathrm{GB}$ of aggregate transfer for the entire training run. The central server sustains only $\approx 7\,\mathrm{MB}$ of ingress and $7\,\mathrm{MB}$ of egress per round. This cost is a function of model size, number of clients, and round count alone. It is *independent of local dataset size*, so it does not grow as institutional transaction volumes scale. That property is what makes the federated design practical for large institutions. Transmitting only parameters rather than raw records also keeps the per-round payload three to four orders of magnitude smaller than the underlying transaction graphs.

We use relation-wise neighbor sampling of fan-out $S = [S_1, S_2]$ over $L = 2$ layers with hidden dimension d. A single local epoch then costs $O\big(|\mathcal{E}^{(k)}|\,|\mathcal{R}|\,\prod_\ell S_\ell\, d^2\big)$, and the full federated procedure costs $O\big(T \sum_k |\mathcal{E}^{(k)}|\,|\mathcal{R}|\,\prod_\ell S_\ell\, d^2\big)$. Fixed-size sampling keeps this cost linear in the number of edges. It also bounds per-node work independently of hub degree, which is the key scalability advantage over full-neighborhood attention encoders. A complete federated run uses five clients, $T = 20$ rounds, and up to 50 early-stopped local epochs. Such a run completes in $9.4\,\mathrm{min}$ on PaySim and $12.1\,\mathrm{min}$ on the Credit Card Fraud 2023 Dataset. These times use the hardware of Table II. Graph construction is a one-time preprocessing cost. It requires about $38\,\mathrm{s}$ for PaySim, where edges follow directly from account identifiers. It requires about $74\,\mathrm{s}$ for the Credit Card dataset, where the approximate k-NN `similar_to` edges dominate at $O(n \log n)$. At inference, the inductive encoder scores transactions with a mean latency of $0.71\,\mathrm{ms}$ per edge and a batched throughput of about $8{,}400$ transactions per second. Peak GPU memory is $2.3\,\mathrm{GB}$ on PaySim and $1.8\,\mathrm{GB}$ on the Credit Card dataset. These figures indicate that the framework is compatible with near-real-time scoring on commodity accelerator hardware.

TABLE IV. COMPUTATIONAL AND COMMUNICATION COST PROFILE OF FHT-GRAPHSAGE

Quantity	PaySim	Credit Card
Trainable parameters	$\approx 0.34\,\mathrm{M}$	
Model size (fp32)	$1.4\,\mathrm{MB}$	
Comm. per client / round	$\approx 2.8\,\mathrm{MB}$	
Comm. per client (total, T=20)	$\approx 56\,\mathrm{MB}$	
Aggregate comm. (K=5, total)	$\approx 0.28\,\mathrm{GB}$	
Graph construction time	$38\,\mathrm{s}$	$74\,\mathrm{s}$
Full training time	$9.4\,\mathrm{min}$	$12.1\,\mathrm{min}$
Inference latency / edge	$0.71\,\mathrm{ms}$	
Inference throughput	$\approx 8{,}400$ tx/s	
Peak GPU memory	$2.3\,\mathrm{GB}$	$1.8\,\mathrm{GB}$

C. Discussion of Design Choices and Threats to Validity

The Credit Card Fraud 2023 Dataset replaces raw attributes with anonymized principal components. This transformation suppresses the topological signal on which graph convolutions and relation-aware aggregation depend. The effect is directly visible in the results. Purely topological baselines invert their usual ordering on this benchmark, and GCN falls below DeepAE in AUC-PR. The absolute AUC-PR values are also compressed relative to PaySim. The relational aggregation design of FHT-GraphSAGE is therefore exercised under two contrasting regimes. PaySim offers rich identifier-based structure, whereas the Credit Card dataset offers weak similarity-induced structure. The framework leads on both. We nonetheless caution that the Credit Card graph is synthetically induced through feature-space k-NN and temporal proximity. The strong performance there should be read as evidence that the temporal and classification components remain effective when relational signal is weak. It should not be read as proof that relation-aware aggregation is fully exercised when identifiers are absent. Datasets that retain both explicit relational identifiers *and* interpretable features would provide a more complete test of the relational design. Such datasets are a priority for future evaluation.

Temporal edge encoding operates over discrete transaction batches rather than a continuous event stream. This introduces a trade-off between granularity and latency. Coarser batches amortize graph-construction and aggregation cost. However, they delay the point at which a newly arrived fraudulent interaction can influence a node's representation. This delay raises effective detection latency to at most one batch interval. Finer batches reduce this latency. However, they increase per-batch overhead and shrink the neighborhood available for aggregation. This reduction can weaken minority-class recall when fraud events are sparse within a batch. Batch granularity also bounds drift sensitivity. Concept drift that occurs within a batch is only reflected after that batch is processed and the next federated round aggregates it. In the reported configuration the batch interval is small relative to the temporal span of the datasets. This effect is therefore not visible in the aggregate metrics. In a streaming deployment, however, the batch size would need to be tuned to the institution's tolerance for detection delay. Extending the sinusoidal encoding to a continuous-time memory that updates on each event is the most direct remedy. That extension is left to future work.

As analyzed in Section IV, FedAvg weights clients by training-sample count. This weighting is exactly optimal only

under IID client data. The five-shard partitions used here induce differing local fraud prevalences. The framework retains its lead under this non-IID setting. Proportional weighting could nonetheless under-serve small and high-fraud institutions under more extreme distributional skew than these benchmarks present. Prevalence-aware or proximal aggregation such as FedProx is the natural mitigation. It is noted in the conclusion as future work.

VI. Ablation Study

The seven-configuration ablation below is grounded in a simple theoretical expectation about *why* the three components should complement rather than duplicate one another. Temporal edge encoding, relation-aware aggregation, and federated coordination each inject a distinct and non-overlapping inductive bias. Temporal encoding supplies an ordering signal. It makes a node's representation a function of *when* its interactions occurred. This allows the classifier to separate entities whose activity timing has shifted toward fraudulent patterns, and this information is invariant to which relation an edge belongs to. Relation-aware aggregation supplies a structural signal. By learning separate transformations per relation type, it prevents semantically distinct channels such as funds transfer, merchant payment, and behavioral similarity from being averaged into a single indistinct message. This structural information is orthogonal to timing. Federated coordination supplies a distributional signal. Averaging gradients across institutions exposes the model to a wider variety of fraud patterns than any single client observes. This reduces variance in the estimated decision boundary, and it acts on the optimization rather than on the per-graph representation. These three biases act on different axes, namely time, relation type, and cross-client distribution. Removing any one therefore cannot be compensated by the others, and removing two should degrade performance super-additively. The ablation results that follow confirm this predicted pattern. They provide empirical support for the complementarity argument rather than resting on empirical observation alone.

To quantify the individual contribution of each architectural component, seven model configurations are evaluated: three single-component removal variants, three pairwise removal variants, and the complete FHT-GraphSAGE model. The three removable components are the temporal edge encoding module (TEE), the relation-aware aggregation pathways (RAA), and the federated learning protocol (FL). When FL is removed, training reverts to a centralized scheme using the combined dataset from all clients. Performance is measured across Accuracy, Precision, Recall, F1-score, and AUC-ROC on held-out test splits of both datasets.

Fig. 8 presents the comparative performance across all configurations. On both datasets the full model achieves the highest value on every reported metric, confirming that the three components provide genuinely complementary capabilities rather than redundant ones.

Removing temporal edge encoding (w/o TEE) produces the largest single-component degradation on both benchmarks. On PaySim, accuracy falls to 0.936 and AUC-ROC declines to 0.961; on the Credit Card dataset, accuracy drops to 0.912 and AUC-ROC to 0.940. Without sinusoidal time embeddings, the encoder receives no information about when interactions occurred relative to one another, preventing it from distinguishing entities whose activity frequency or timing has shifted toward patterns associated with fraudulent behavior. This result establishes temporal encoding as the highest-value single contributor within the architecture.

Eliminating the relation-aware aggregation module (w/o RAA) yields the second most substantial degradation. On PaySim, accuracy decreases to 0.944 and AUC-ROC to 0.968; on the Credit Card dataset, these figures are 0.921 and 0.950 respectively. Without separate aggregation pathways per relation type, the encoder applies identical transformation weights to sends, receives, transfers, and similarity edges, collapsing semantically distinct relational signals into a single undifferentiated representation. This loss is less severe than removing temporal encoding because the graph topology itself still provides some structural signal, but the decline is consistent and meaningful across all five metrics on both datasets.

Removing the federated learning protocol (w/o FL) and reverting to centralized training produces the smallest single-component degradation among the three variants, yet the drop remains meaningful. On PaySim, accuracy falls to 0.951 and AUC-ROC to 0.975; on the Credit Card dataset, accuracy is 0.928 and AUC-ROC is 0.957. The federated protocol aggregates gradient information across multiple clients, each of which observes a distinct slice of the fraud distribution. Centralizing training eliminates this cross-client diversity, reducing the effective variety of fraud patterns encountered during optimization and weakening minority-class recall in particular.

Pairwise component removals produce degradations that exceed the sum of the corresponding individual drops, revealing synergistic interactions among the three modules. Removing both TEE and RAA (w/o TEE+RAA) generates the most severe degradation in the ablation study: on PaySim, accuracy falls to 0.901 and AUC-ROC to 0.933; on the Credit Card dataset, these values are 0.878 and 0.909. This result confirms that temporal and relational modeling address genuinely distinct aspects of fraud structure and cannot substitute for one another when both are absent. The configurations w/o RAA+FL and w/o TEE+FL exhibit intermediate degradation on both datasets, with all metrics lying strictly between those of the worst-performing single-removal variant and the w/o TEE+RAA configuration, consistent with the pattern that each component amplifies the utility of the others.

Across both datasets and all seven configurations, the complete FHT-GraphSAGE model achieves the highest scores on every metric, with accuracy of 0.963 and AUC-ROC of 0.986 on PaySim, and accuracy of 0.941 and AUC-ROC of 0.969 on the Credit Card Fraud 2023 Dataset. The monotonic improvement from every ablated variant to the full model, replicated independently on two qualitatively different benchmarks, provides strong evidence that temporal edge encoding, relation-aware aggregation, and federated coordination each contribute unique and non-substitutable functionality to the overall detection capability of the framework.

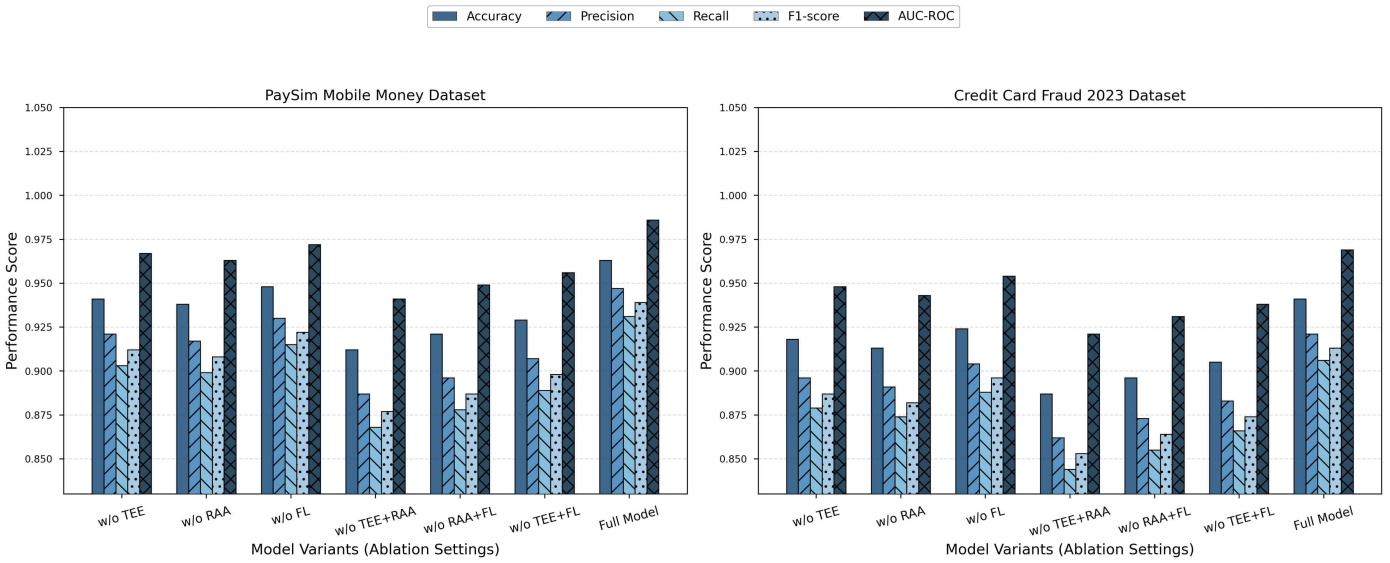

Fig. 8. Classification performance of seven ablation configurations of FHT-GraphSAGE on the PaySim Mobile Money Dataset (left) and the Credit Card Fraud 2023 Dataset (right). Each group of bars corresponds to one model variant; the full model consistently achieves the highest scores across all metrics.

VII. Conclusion

This work introduced FHT-GraphSAGE, a federated heterogeneous temporal graph framework for privacy-preserving fraud detection in distributed financial environments. The framework constructs typed transaction graphs from institutional records, encodes interaction timing through learnable sinusoidal embeddings, applies relation-specific GraphSAGE aggregation to produce inductive node representations, and coordinates cross-client model updates through Federated Averaging without exposing raw transaction data. Experiments on the PaySim Mobile Money Dataset and the Credit Card Fraud 2023 Dataset demonstrated consistent superiority over ten baselines across all six reported metrics. AUC-PR leadership was maintained down to a fraud ratio of 0.25% on both benchmarks. Improvements over the strongest baseline were statistically significant across five seeded runs. Ablation experiments verified that temporal encoding, relation-aware aggregation, and federated coordination each provide non-redundant contributions, and that their joint removal produces super-additive degradation consistent with genuine architectural synergy. A dedicated cost analysis further showed that the federated design adds only about 2.8 MB of per-client communication per round. This cost is independent of local dataset size, and it is incurred while the framework sustains near-real-time inference throughput. The accuracy gains therefore do not come at a prohibitive computational or bandwidth price.

Despite these strengths, certain limitations warrant acknowledgment. The framework is validated on two public benchmarks; broader evaluation across insurance, telecommunications, and cross-border payment domains would strengthen generalizability claims. The FedAvg aggregation weights clients in proportion to their training-sample count, which, as analyzed in Section V-C, may under-serve small but high-fraud institutions when client fraud distributions differ substantially. Temporal encoding currently operates on discrete transaction batches rather than continuous event streams, limiting responsiveness in real-time deployment settings. Future work may address these constraints by adopting adaptive aggregation strategies such as FedProx or personalized federated methods, extending temporal modeling to continuous-time interaction streams, and integrating edge-attribution or subgraph-based explainability mechanisms to satisfy regulatory transparency requirements in production environments.

References

[1] Association of Certified Fraud Examiners, "Report to the nations: 2020 global study on occupational fraud and abuse," ACFE Global Headquarters, Austin, TX, USA, Tech. Rep., 2020.

[2] J. West and M. Bhattacharya, "Intelligent financial fraud detection: A comprehensive review," *Computers & security*, vol. 57, pp. 47–66, 2016.

[3] A. Khanum, K. Chaitra, B. Singh, and C. Gomathi, "Fraud detection in financial transactions: A machine learning approach vs. rule-based systems," in *2024 International conference on intelligent and innovative technologies in computing, electrical and electronics (IITCEE)*. IEEE, 2024, pp. 1–5.

[4] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection," *Information Sciences*, vol. 479, pp. 448–455, 2019.

[5] R. H. Chowdhury, "Advancing fraud detection through deep learning: A comprehensive review," *World Journal of Advanced Engineering Technology and Sciences*, vol. 12, no. 2, pp. 606–613, 2024.

[6] A. Roy, J. Sun, R. Mahoney, L. Alonzi, S. Adams, and P. Beling, "Deep learning detecting fraud in credit card transactions," in *2018 systems and information engineering design symposium (SIEDS)*. IEEE, 2018, pp. 129–134.

[7] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM computing surveys (CSUR)*, vol. 46, no. 4, pp. 1–37, 2014.

[8] Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, "Enhancing graph neural network-based fraud detectors against camouflaged fraudsters," in *Proceedings of the 29th ACM international conference on information & knowledge management*, 2020, pp. 315–324.

[9] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decision support systems*, vol. 50, no. 3, pp. 602–613, 2011.

[10] R. J. Bolton and D. J. Hand, "Statistical fraud detection: A review," *Statistical science*, vol. 17, no. 3, pp. 235–255, 2002.

[11] C. Whitrow, D. J. Hand, P. Juszczak, D. Weston, and N. M. Adams, "Transaction aggregation as a strategy for credit card fraud detection," *Data mining and knowledge discovery*, vol. 18, no. 1, pp. 30–55, 2009.

[12] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," *Expert systems with applications*, vol. 41, no. 10, pp. 4915–4928, 2014.

[13] D. Wang, Y. Chen, J. Zhang, W. Gao, B. Wang, Q. Wang, and M. Zhang, "Deep fraud detector: A deep learning framework for financial fraud detection," in *Proc. IEEE Int. Conf. Data Mining*, 2019, pp. 1361–1366.

[14] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data mining and knowledge discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[15] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 1, pp. 1–39, 2012.

[16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[17] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *arXiv preprint arXiv:1709.05584*, 2017.

[18] Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, "Enhancing graph neural network-based fraud detectors against camouflaged fraudsters," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, ser. CIKM '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 315–324. [Online]. Available: https://doi.org/10.1145/3340531.3411903

[19] Z. Liu, Y. Dou, P. S. Yu, Y. Deng, and H. Peng, "Alleviating the inconsistency problem of applying graph neural network to fraud detection," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1569–1572. [Online]. Available: https://doi.org/10.1145/3397271.3401253

[20] S. Khosravi, M. Kargari, B. Teimourpour, and M. Talebi, "Transaction fraud detection via attentional spatial–temporal gnn," *The Journal of Supercomputing*, vol. 81, no. 4, p. 537, 2025.

[21] S. Islam, G. Raj Gupta, A. Chakraborty, S. Singh, A. Soni, and C. Patle, "Detecting fraudulent transactions for different patterns in financial networks using layer weigthed gcn," *Human-Centric Intelligent Systems*, vol. 5, no. 2, pp. 181–195, 2025.

[22] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, "Temporal graph networks for deep learning on dynamic graphs," *arXiv preprint arXiv:2006.10637*, 2020.

[23] Q. Li, Y. Shang, X. Qiao, and W. Dai, "Heterogeneous dynamic graph attention network," in *2020 IEEE international conference on knowledge graph (ICKG)*. IEEE, 2020, pp. 404–411.

[24] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: a realistic modeling and a novel learning strategy," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 8, pp. 3784–3797, 2017.

[25] Y. Cui, X. Han, J. Chen, X. Zhang, J. Yang, and X. Zhang, "Fraudgnn-rl: a graph neural network with reinforcement learning for adaptive financial fraud detection," *IEEE Open Journal of the Computer Society*, 2025.

[26] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[27] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?" *arXiv preprint arXiv:2105.14491*, 2021.

[28] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. Pmlr, 2017, pp. 1273–1282.

[29] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[30] T. Awosika, R. M. Shukla, and B. Pranggono, "Transparency and privacy: the role of explainable ai and federated learning in financial fraud detection," *IEEE access*, vol. 12, pp. 64 551–64 560, 2024.

[31] F. Zheng, Erihe, K. Li, J. Tian, and X. Xiang, "A federated interpretable scorecard and its application in credit scoring," *International Journal of Financial Engineering*, vol. 8, no. 03, p. 2142009, 2021.

[32] N. F. Aurna, M. D. Hossain, Y. Taenaka, and Y. Kadobayashi, "Federated learning-based credit card fraud detection: Performance analysis with sampling methods and deep learning algorithms," in *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*. IEEE, 2023, pp. 180–186.

[33] E. Lopez-Rojas, A. Elmir, and S. Axelsson, "Paysim: A financial mobile money simulator for fraud detection," in *28th European modeling and simulation symposium, EMSS, Larnaca*. Dime University of Genoa, 2016, pp. 249–255.

[34] N. Elgiriyewithana, "Credit card fraud detection dataset 2023," 2023. [Online]. Available: https://www.kaggle.com/dsv/6492730