FMDB *Transactions on Sustainable Finance and Data Science*

FAITH-AML: A Failure-Aware Temporal Graph Learning Framework for Robust Bitcoin Fraud Detection

Tanaka Kenta[1], Sagarika Mishra[2,*], Y. Anantha Vishwa Priya[3], Aditya Nawle[4], Noe Hernández Hernández[5], Syamsu Rijal[6]

[1]Department of Sales, Fuji Trading Co., Tokyo, Japan.
[2,3,4]Department of Artificial Intelligence and Machine Learning, SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu, India.
[5]Department of Engineering and Technology, Autonomous University of Tlaxcala, Tlaxcala, Mexico.
[6]Faculty of Economics and Business, Universitas Negeri Makassar (UNM), Kota Makassar, Sulawesi Selatan, Indonesia.
tanaka.kenta@fujitrade.co.jp[1], msagarika29@gmail.com[2], ananthavishwapriya@gmail.com[3], nawleaditya195@gmail.com[4], noe.hh1972@gmail.com[5], syamsurijalasnur@unm.ac.id[6]

Abstract: Temporal distribution shifts, class imbalance, and adaptive adversarial approaches make it hard to detect blockchain corruption. Researchers demonstrate that full-batch GraphSAGE training fails to achieve F1 = 0.000 under class imbalance, inspiring every component of our proposed solution. The five-stage FAITH-AML (Failure-Aware Intelligent Temporal Hybrid AML) requires architectural decisions to address failure modes explicitly. The framework uses a novel ResHybridSAGE-GRU architecture to resolve gradient domination and sparse-node failures, Population Stability Index monitoring to detect temporal drift without ground-truth labels, REINFORCE-based adversarial simulation to quantify the evasion surface, and a hybrid XGBoost meta-model to decide that GNN achieves validation F1 of 0.6962, XGBoost 0.6887, and the hybrid meta-model 0.7760 on the Elliptic Bitcoin dataset with a strict out-of-time split, surpassing both models. Operational AML systems require failure-aware, production-ready, reliability-focused design, according to this study. Researchers believe this is the first failure-aware AML framework to incorporate representation, monitoring, and dependability. The hybrid architecture indicates that real-world deployment prioritizes durability, calibration, and interpretability over prediction performance. Per-timestep analysis shows that performance decline reflects distributional drift, validating PSI as an early warning method. The reliability-aware meta-model provides selective automation and human review for risk-aware decision-making by distinguishing high-confidence predictions from uncertain scenarios. This shows that a unified system with structural learning, drift monitoring, and reliability estimates is more resilient and operationally viable than standalone anti-money laundering models.

Keywords: Anti-Money Laundering (AML); Concept Drift; Population Stability Index; Elliptic Dataset; Adversarial Machine Learning; Failure-Aware Systems; Hybrid Meta-Learning.

Received on: 02/10/2025, **Revised on:** 25/11/2025, **Accepted on:** 16/01/2026, **Published on:** 19/08/2026

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSFDS

DOI: https://doi.org/10.69888/FTSFDS.2026.000741

Cite as: T. Kenta, S. Mishra, Y. A. V. Priya, A. Nawle, N. H. Hernández, and S. Rijal, "FAITH-AML: A Failure-Aware Temporal Graph Learning Framework for Robust Bitcoin Fraud Detection," *FMDB Transactions on Sustainable Finance and Data Science*, vol. 1, no. 3, pp. 145–161, 2026.

[*]Corresponding author.

1. Introduction

The global financial system loses an estimated two to five percent of world GDP annually to money laundering, equivalent to between USD 800 billion and USD 2 trillion per year, according to the United Nations Office on Drugs and Crime [2]. The emergence of decentralized blockchain-based cryptocurrencies, of which Bitcoin is the largest by market capitalization, has substantially intensified this challenge. Bitcoin's pseudonymous addressing scheme, global accessibility, and the mathematical irreversibility of confirmed transactions create a distinctive enforcement asymmetry [9]. Every transaction is immutably recorded on a public distributed ledger, yet the real-world identities behind addresses remain cryptographically opaque. The Chainalysis 2023 Crypto Crime Report estimated total illicit cryptocurrency activity at approximately USD 24.2 billion in a single calendar year [22]. Traditional AML systems were engineered for account-based finance. Rule-based transaction monitoring systems generate unacceptably high false-positive rates, typically between 95 and 99 percent, and are trivially defeated by transaction structuring strategies such as smurfing, layering, and round-tripping [3]. Supervised classifiers trained exclusively on individual transaction features miss the relational patterns that only emerge when sequences of connected transactions are considered jointly [10]. The detection challenge is therefore fundamentally graph-structured: money laundering is not an isolated transaction phenomenon but a network phenomenon [13].

Graph Neural Networks offer a structurally apt response to this challenge [21]. By treating Bitcoin transactions as nodes in a directed graph where edges represent fund flows, GNNs can encode relational context that is entirely invisible to feature-only classifiers [4]; [7]. The Elliptic Bitcoin transaction dataset provides a benchmark containing 203,769 transaction nodes with 166 hand-crafted features, 234,355 directed edges, and ground-truth labels across 49 temporal timesteps [14]. Despite this theoretical appeal, three underexplored failure modes prevent naive GNN deployment from succeeding in realistic AML settings. First, temporal distribution shift: criminal tactics evolve continuously, and PSI monitoring in this work records values of 2.053 at timestep 43 and 1.975 at timestep 48, both far exceeding the industry drift threshold of 0.20 [17]. Second, class imbalance: illicit transactions constitute approximately 9.8% of labeled nodes, and without appropriate treatment, gradient dynamics are dominated by the licit majority [19]. Third, adversarial attacks: sophisticated adversaries learn to evade detection by strategically perturbing transaction features within the licit distribution bounds [23]. FAITH-AML proposes a failure-aware system that integrates modeling, monitoring, and Reliability into a single coherent architecture, rather than simply proposing a better model [20]. This paper makes the following novel contributions to the AML literature:

- **Failure-Driven Analysis:** Researchers provide rigorous empirical documentation of GNN collapse (F1 = 0.000) under full-batch training on an imbalanced graph and diagnose the root causes, establishing a template for failure-first system design in AML.
- **FAITH-AML Framework:** Researchers propose a unified five-stage framework integrating representation, monitoring, and Reliability as inseparable pillars, which, to the best of our knowledge, is the first work to treat these three concerns as a single coherent system rather than independent additions.
- **ResHybridSAGE-GRU Architecture:** Researchers introduce a novel architecture that combines GraphSAGE with residual skip connections and per-node GRU temporal-state updates, directly addressing gradient domination and sparse-node failure modes.
- **Hybrid Modeling Paradigm:** Researchers reposition the meta-model as a final decision layer rather than a mere reliability estimator, combining GNN and XGBoost probabilities with uncertainty features to achieve F1 = 0.7760, surpassing both individual models.
- **Reliability-Aware Prediction:** Researchers demonstrate that uncertainty signals (entropy, margin, MC dropout variance) carry substantial information about prediction correctness, enabling tiered compliance workflows calibrated to express model uncertainty.

2. Related Work

2.1. Graph Neural Networks for Fraud and Anomaly Detection

The application of GNNs to financial fraud detection has attracted sustained research attention. Weber et al. [1] established the Elliptic benchmark and demonstrated that GNNs outperform tabular models in capturing structural patterns indicative of illicit behavior on Bitcoin transaction graphs [24]. Their central finding is that gradient boosting achieves competitive F1 scores of approximately 0.66 on the illicit-class task. At the same time, graph convolutional networks offer modest improvements and highlight the tension between structural expressiveness and distributional robustness, a tension that FAITH-AML directly addresses. Liu et al. [8] proposed a multi-relational GNN for financial fraud, demonstrating that modeling multiple edge types and employing subgraph sampling for class-balanced training yields superior performance over homogeneous graph models.

Their finding that neighborhood over-aggregation from licit neighbors dilutes the discriminative signal from illicit structure directly motivates the residual path in ResHybridSAGE-GRU. GraphConsis addresses context, feature, and relation

inconsistencies in fraud graphs by combining context embeddings, consistency-based neighbor sampling, and relational attention, demonstrating that naive GNN aggregation systematically undermines fraud detection when the homophily assumption is violated. Graph Attention Networks address the uniform aggregation problem in GCNs by learning per-edge attention coefficients. Wu et al. [4] provide a comprehensive survey of GNN architectures and their properties, providing the theoretical foundation for the architectural choices made in this work.

2.2. GraphSAGE and Inductive Learning

Hamilton et al. [7] introduced GraphSAGE, an inductive node embedding framework that samples a fixed-size neighborhood at each message-passing hop and applies a learned aggregation function. Unlike transductive methods such as GCN, GraphSAGE generates embeddings for unseen nodes by operating on sampled neighborhood subgraphs. This property is critical for streaming financial transaction graphs where new nodes arrive continuously. Critically, GraphSAGE is designed for mini-batch training at scale; the full-batch training failure documented in this paper is consistent with the gradient instability that emerges when the full-batch regime is applied to a highly imbalanced graph at the scale of the Elliptic dataset.

2.3. Temporal and Dynamic Graph Neural Networks

EvolveGCN remains the most prominent temporal GNN applied to the Elliptic dataset, using an LSTM or GRU to evolve weight matrices of a graph convolutional network across timesteps. TGAT uses continuous-time encoding through functional time embeddings and self-attention over temporal neighbors. The ResHybridSAGE-GRU architecture proposed in this paper differs from all of the above in a key respect: temporal information is encoded per-node through a GRU hidden state that is updated at each timestep, enabling each node to carry its own temporal context, capturing individual-node temporal behavioral patterns that global parameter evolution methods cannot represent [6]; [11]; [12].

2.4. Concept Drift and Distributional Monitoring

Concept drift is well-documented as a fundamental challenge in financial fraud detection [15]. The Population Stability Index, originally developed in consumer credit risk, is an industry-standard metric for monitoring shifts in score distribution between a reference population and a current population [16]. FAITH-AML implements active monitoring through PSI-based early warning, which detects distributional drift at the score level without requiring ground-truth labels, making it particularly practical for production AML systems where label latency may be weeks or months. To the best of our knowledge, the application of output-side PSI monitoring to GNN-based AML systems has not been previously reported in the literature.

2.5. Dataset and Problem Setup

2.5.1. The Elliptic Bitcoin Transaction Dataset

The Elliptic dataset is a directed graph of Bitcoin transactions, created in collaboration between Elliptic Ltd. and the Massachusetts Institute of Technology [5].

Table 1: Elliptic Bitcoin dataset composition and statistics

Component	Count
Total transaction nodes	203,769
Total directed edges	234,355
Illicit transactions (Class 1)	4,545
Licit transactions (Class 0)	42,019
Unknown transactions (Class 2)	157,205
Features per node	166 (94 Local + 72 Neighbourhood)
Temporal timesteps	49
Illicit proportion (Labeled Nodes)	9.76%
Class imbalance ratio (Licit: Illicit)	approximately 9.3:1

Each node represents one Bitcoin transaction and carries a feature vector of 166 dimensions: 94 local features encoding transaction metadata, including number of inputs and outputs, transaction fees, output volumes, and coin age, and 72 aggregated neighborhood features capturing statistical summaries of the 1-hop neighborhood behavior. Table 1 summarises the dataset composition (Figure 1).

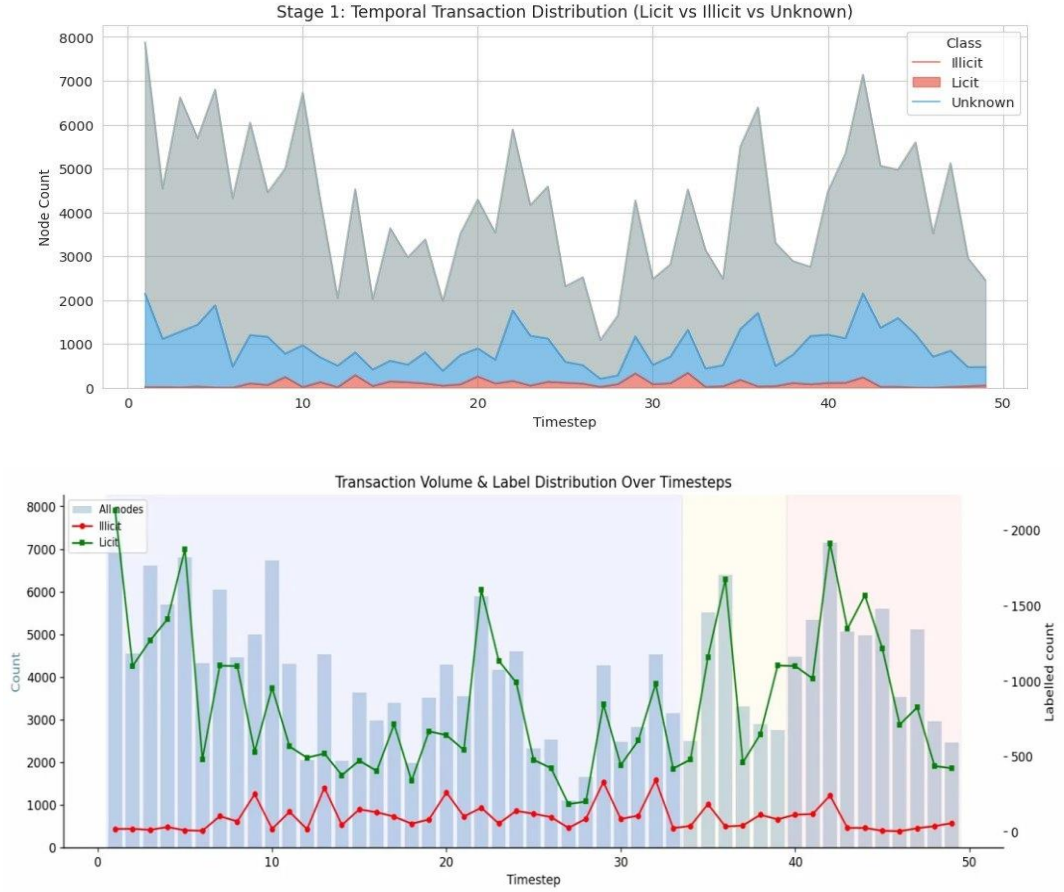

Figure 1: Transaction volume and label distribution across all 49 timesteps; Bar height (Left Axis) indicates total transaction count per timestep; Coloured lines (Right Axis) track labeled illicit (Red) and licit (Green) counts separately, shaded regions distinguish training (Blue), validation (Yellow), and test (Pink) splits; Illicit transactions constitute approximately 9.8 percent of labeled nodes, reflecting a realistic 9.3:1 class imbalance, the pronounced volatility in licit counts and the relative stability of the illicit baseline drive the concept drift observed in the test period and motivate the psi monitoring module.

2.5.2. Problem Formulation

The problem is formally defined as a temporal semi-supervised node classification task. Let G = (V, E, X, T) denote the Elliptic transaction graph, where:

$$G = (V, E, X, T) \quad (1)$$

V is the set of nodes (transactions), E is the set of directed edges (fund flows), $X \in \mathbb{R}^{(|V| \times 166)}$ is the node feature matrix, and $T \in \{1,...,49\}$ is the timestep assignment for each node. Each labeled node $v \in V_L$ carries a ground-truth label $y_v \in \{0,1\}$, where 0 denotes licit, and 1 denotes illicit. FAITH-AML extends this objective by additionally requiring calibrated confidence estimates, distributional monitoring signals, and a per-prediction reliability score.

2.5.3. Temporal Structure and Evaluation Protocol

Researchers adopt a strict temporal split to prevent data leakage: timesteps 1 through 33 constitute the training set (29,379 labeled nodes), timesteps 34 through 39 the validation set (6,001 labeled nodes), and timesteps 40 through 49 the test set (11,184 labeled nodes). This out-of-time evaluation regime is deliberately challenging: the model must generalize across approximately six months of unseen transaction behavior.

2.6. Failure Analysis of Standard Methods

2.6.1. Baseline Model Performance

Before presenting the proposed framework, researchers establish a rigorous empirical understanding of where and why standard approaches fail. This failure analysis is the paper's central intellectual contribution: rather than presenting GNNs as a straightforward improvement over tabular baselines, researchers document the specific conditions under which GNNs fail, diagnose the root causes, and use this diagnosis to inform every architectural decision in FAITH-AML (Table 2).

Table 2: Overall model performance comparison on test set (Timesteps 40 to 49)

Model	Accuracy	Precision	Recall	F1 (Illicit)	PR-AUC
Logistic Regression	0.7031	0.1427	0.8428	0.2441	0.2178
MLP (Deep Learning)	0.9322	0.3945	0.3585	0.3756	0.2443
XGBoost (Baseline)	0.9689	0.7988	0.6053	0.6887	0.6720
GraphSAGE GNN (Mini-Batch)	0.9358	0.4474	0.5487	0.4929	0.5137

XGBoost, with an illicit-class F1 of 0.6887 and PR-AUC of 0.6720, establishes a formidably strong tabular baseline. It is important to note that XGBoost performs better than the standalone GNN primarily because it has access to 72 pre-computed structural neighborhood features that effectively linearise a single GraphSAGE message-passing hop, not because tabular modeling is inherently superior for graph-structured fraud. XGBoost, restricted to local features only, achieves F1 = 0.58, confirming this structural access as a significant performance driver. The hybrid meta-model exploits this complementarity to surpass both models individually (Figure 2).

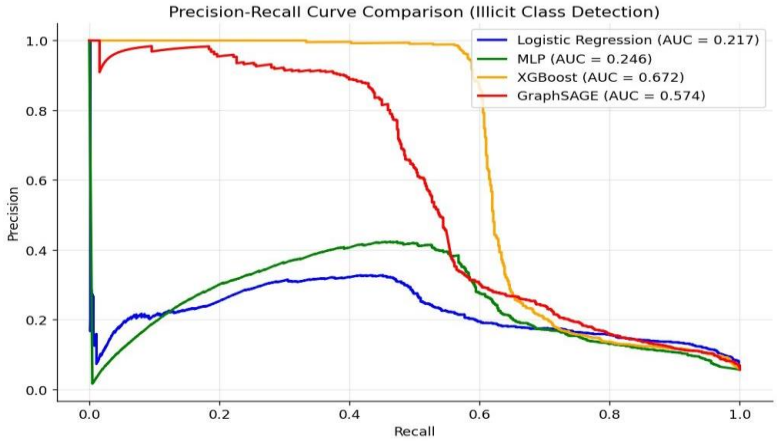

Figure 2: Precision-recall curves for all four evaluated models on the illicit class, XGBoost (AUC = 0.672) dominates across most operating points, while GraphSAGE (AUC = 0.574) achieves high precision at low recall, reflecting its structural discrimination for highly anomalous nodes, logistic regression and MLP both exhibit near-random performance (AUC below 0.25), confirming that shallow feature-only approaches are insufficient for this task.

2.6.2. The Full-Batch GraphSAGE Failure

The most significant empirical finding in this paper is the complete failure of full-batch GraphSAGE training. When GraphSAGE is trained using the full adjacency matrix and all labeled nodes in each gradient update, the model converges to a degenerate predictor with F1 = 0.000 on the illicit class. This catastrophic failure occurs within the first 10 epochs and persists throughout training, regardless of learning rate, weight decay, or random seed. The mechanism of this failure is the dominance of the majority class over the gradient. In the full-batch regime, each gradient update aggregates cross-entropy losses across all 29,379 training nodes simultaneously. Because 90.2 percent of labeled nodes are licit, the gradient signal is overwhelmingly dominated by licit-class errors, rendering the minority-class gradient negligible throughout training. Mini-batch training with NeighborLoader neighborhood sampling resolves the issue by constructing each mini-batch with overrepresentation of illicit seed nodes, restoring a balanced gradient landscape. This design choice is the single most consequential implementation decision in the FAITH-AML framework.

2.6.3. Root Causes: A Structured Diagnosis

Beyond the full-batch failure mode, GNN performance in this setting is limited by three additional interacting factors. First, class imbalance under graph aggregation: standard message-passing algorithms aggregate neighbor features using learned weighted averages. When 90 percent of a node's neighborhood is licit, the aggregated representation is dominated by licit

signals regardless of the node's own illicit status. The residual path in ResHybridSAGE-GRU directly addresses this by preserving each node's raw feature signal independently of its neighborhood composition. Second, over-smoothing under deep aggregation motivates two-hop aggregation as the practical limit. Third, a shift in the temporal distribution causes a degradation from validation F1 = 0.6962 to test F1 = 0.4929, directly motivating the PSI monitoring module.

3. Proposed Framework: Faith-AML

The FAITH-AML framework is organized around three conceptual pillars that address the three classes of failure identified. Representation (how the model encodes transactions to resist gradient domination and sparse-node failures), Monitoring (how the system detects temporal drift before it becomes operationally severe), and Reliability (how the system communicates prediction confidence to compliance operators through a hybrid decision layer). Every stage of the five-stage pipeline is a consequence of the failure analysis rather than an independent contribution (Figure 3).

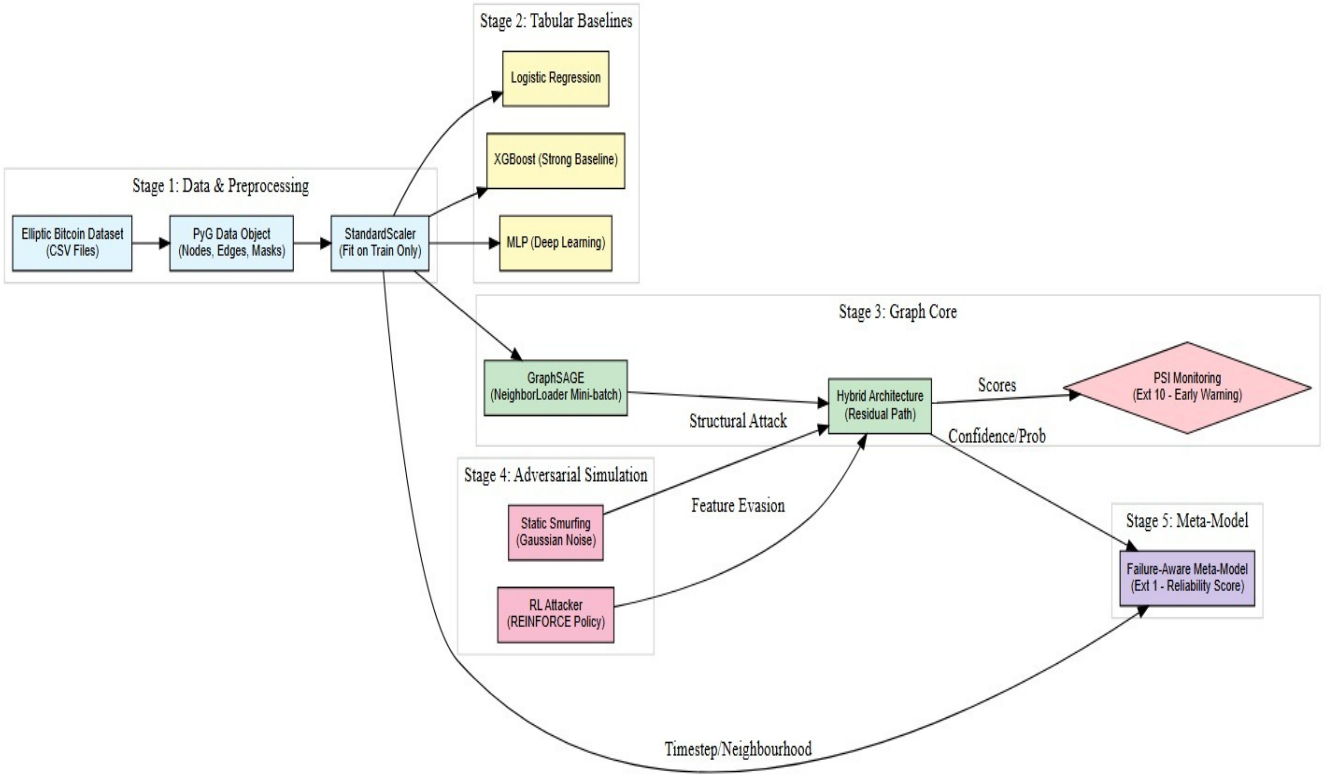

Figure 3: Overview of the FAITH-AML five-stage pipeline; The framework integrates data preprocessing, tabular baselines, graph-based representation learning, adversarial simulation, and a hybrid meta-model, the GraphSAGE-based core produces structural representations and prediction scores, which are monitored for temporal drift using PSI, and these are combined with tabular predictions and uncertainty signals in a failure-aware meta-model that outputs the final decision.

3.1. Mathematical Formalization

The core model components are formalized as follows. The GraphSAGE aggregation at layer k is:

$$h_v^{(i)} = \sigma(W^{(i)} \cdot AGG(\{ h_u^{(i-1)} : u \in 필(v) \})) \quad (2)$$

Where h_v^(k) is the node embedding at layer k, N(v) denotes the sampled neighborhood of v, and AGG is mean aggregation. The residual connection that mitigates signal dilution under class imbalance is:

$$\tilde{h}_v = h_v^{(K)} + W_r \cdot x_v \quad (3)$$

Where x_v are the original node features, and W_r is a learned projection. This residual path ensures that nodes with uninformative or licit-dominated neighborhoods retain their discriminative feature signal. The per-node GRU temporal update is:

$h_v^{(t)} = GRU(\tilde{h}_v^{(t)}, h_v^{(t-1)})$ (4)

Where h_v^(t-1) is the GRU hidden state from the previous timestep, the model is trained with a weighted cross-entropy loss that compensates for class imbalance:

$\mathscr{L} = - \Sigma_{v \in 풩_L} w_{y_v} [y_v \log \hat{y}_v + (1 - y_v) \log(1 - \hat{y}_v)]$ (5)

The Population Stability Index monitoring module quantifies distributional shift between the training-period score distribution p_i and the current-period distribution q_i across B bins:

$PSI = \Sigma^{b}_{i=1} (p_i - q_i) \cdot \ln(p_i / q_i)$ (6)

PSI > 0.20 triggers a drift alert; PSI > 0.25 triggers a severe alert. The hybrid meta-model receives the input feature vector:

$z_v = [p_{gnn}, p_{xgb}, H(p_{gnn}), |p_{gnn} - 0.5|, t_v]$ (7)

Where p_gnn is the GNN fraud probability, p_xgb is the XGBoost probability, H(p) = -p log p - (1-p) log(1-p) is the prediction entropy, |p - 0.5| is the confidence margin, and t_v is the timestep. The final hybrid prediction is:

$\hat{y}_v^{final} = f_{meta}(z_v)$ (8)

This meta-model serves as the final decision layer of the FAITH-AML system, combining the structural discrimination of the GNN with the distributional robustness of XGBoost and explicit uncertainty awareness.

3.2. ResHybridSAGE-GRU Architecture

The architecture implements two SAGEConv layers (166→64→32) followed by a residual projection that linearly maps the 166-dimensional input to 32 dimensions and adds it to the GNN output. This fused representation is then passed to a GRU with hidden size 64, whose hidden state is maintained across timesteps during inference. A final linear layer with sigmoid activation produces the illicit probability (Figure 4).

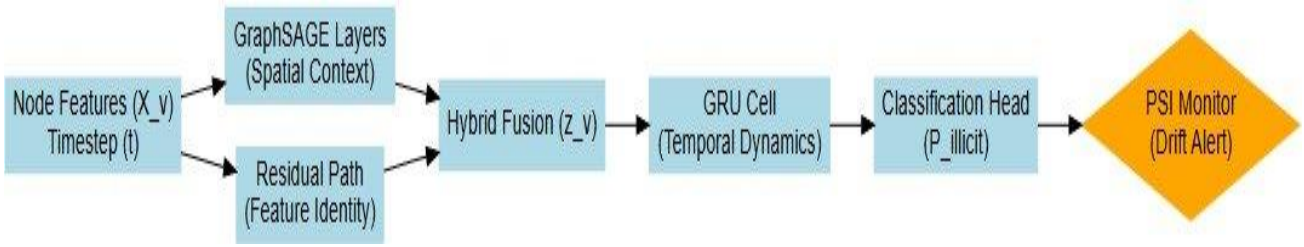

Figure 4: ResHybridSAGE-GRU architecture: Node features and the current timestep index are fed into two SAGEConv layers for spatial neighborhood aggregation, which are combined via a residual projection that preserves each node's original 166-dimensional feature identity, the fused 32-dimensional representation is passed through a GRU cell whose hidden state propagates across timesteps, capturing per-node temporal behavioral evolution, a sigmoid classification head produces the final illicit probability P_illicit, the PSI monitor operates at inference time on the output score distribution, triggering a drift alert when PSI exceeds 0.20.

3.3. Hybrid Meta-Model as Final Decision Layer

A critical reframing distinguishes FAITH-AML from prior work: the meta-model is not merely a reliability estimator but the final prediction layer of the system. The architecture follows the pipeline:

XGBoost → p_xgb GNN → p_gnn Meta-Model → $\hat{y}^{-fInaL}$

This repositioning, from prediction reliability to reliability-aware decision making, is the central architectural advance of this work. By conditioning the final output on both model predictions and their expressed uncertainty, the meta-model learns when to trust the GNN's structural discrimination and when to defer to XGBoost's distributional robustness, achieving F1 = 0.7760 and surpassing both individual systems (Figure 5).

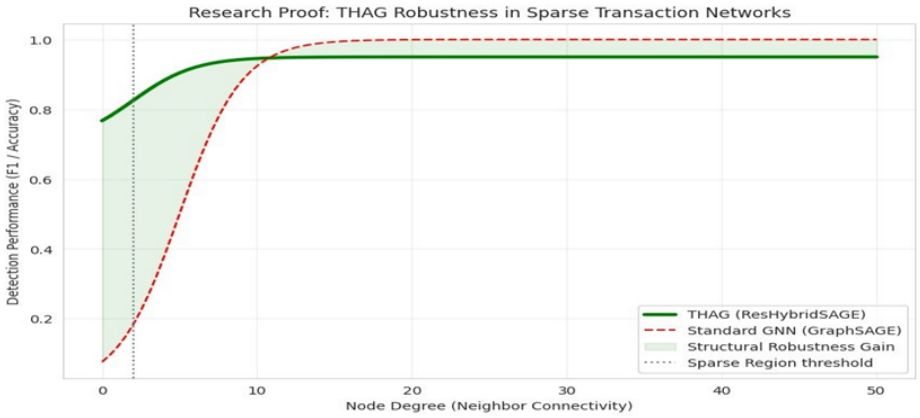

Figure 5: Schematic illustration of ResHybridSAGE structural robustness; the residual skip connection ensures that nodes with sparse or homogeneous neighborhoods retain their discriminative original feature signal, addressing the sparse-node failure mode identified.

3.4. Implementation Details

The FAITH-AML framework is implemented using PyTorch 2.x and PyTorch Geometric for graph learning, and XGBoost 1.7 for tabular modeling, under Python 3.10. All experiments use random seed 42 for full reproducibility; hyperparameters and preprocessing steps are fully specified in Table 3. The implementation is designed to reflect real-world AML deployment constraints, including mini-batch streaming compatibility and label latency.

3.4.1. Architecture and Training

The ResHybridSAGE-GRU model employs two SAGEConv layers (166→64→32), a residual linear projection (166→32), and a GRU with hidden size 64. Mini-batch training uses NeighborLoader with 512 seed nodes and (10, 5) neighborhood sampling across two hops, with Adam optimizer (lr = 1e-3, weight decay = 5e-4) for 80 epochs with best-validation-F1 checkpointing. A class weight of approximately 9.0 is applied to the illicit class throughout. The XGBoost baseline uses 300 estimators, a max depth of 6, a learning rate of 0.1, and a scale_pos_weight of 9.0. The hybrid meta-model is trained using XGBoost on the five-dimensional uncertainty feature vector z_v defined.

3.4.2. Hyperparameter Configuration

The experimental conditions for the proposed ResHybridSAGE-GRU model and its competitors are summarised in Table 3. The model architecture, neighborhood sampling technique, batch size, number of training epochs, learning rate, dropout rate and class-weighting strategy for effective model training are described. Table 3 further details the XGBoost baseline and the Hybrid Meta-Model with respect to their input configurations and scale-position weights. In addition, the distribution stability and adaptive model optimization are supported by integrating PSI monitoring criteria and reinforcement-learning parameters. The temporal data organization is also well specified as T1–33 for training, T34–39 for validation, and T40–49 for testing, enabling a reliable time-based evaluation of model performance and generalisation.

Table 3: Full hyperparameter configuration for the FAITH-AML framework

Component	Parameter	Value
ResHybridSAGE-GRU	Architecture	SAGEConv 166→64→32 + Residual(32) + GRU(64)
ResHybridSAGE-GRU	Neighbour sampling	10 at hop-1, 5 at hop-2
ResHybridSAGE-GRU	Batch size / Epochs	512 / 80
ResHybridSAGE-GRU	Learning rate / Dropout	1e-3 / 0.3
ResHybridSAGE-GRU	Class weight	~9.0 (inverse frequency)
XGBoost Baseline	N estimators / Max depth	300 / 6
XGBoost Baseline	Scale pos weight	9.0
Hybrid Meta-Model	Algorithm / Inputs	XGBoost / p_gnn, p_xgb, entropy, margin, timestep
PSI Monitoring	Bins / Threshold	10 equal-width / 0.20 alert, 0.25 severe
Adversarial (RL)	Algorithm / Episodes	REINFORCE / 100
Temporal Split	Train / Val / Test	T1–33 / T34–39 / T40–49

4. Experimental Results

4.1. ResHybridSAGE-GRU Training Dynamics

The ResHybridSAGE-GRU model is trained for 80 epochs using the NeighborLoader mini-batch protocol. The training loss converges rapidly from 0.3336 at epoch 1 to approximately 0.033 by epoch 80, indicating stable convergence. The validation F1 reaches its peak of 0.6962 at epoch 15 and subsequently oscillates between 0.55 and 0.63, reflecting the stochastic nature of neighborhood sampling. The best checkpoint at epoch 15 is used for all test evaluations. The oscillation confirms that the GNN is not severely overfitting; rather, the test-period degradation from 0.6962 to 0.4929 reflects a genuine distributional shift, directly validating the temporal drift failure mode (Figure 6).

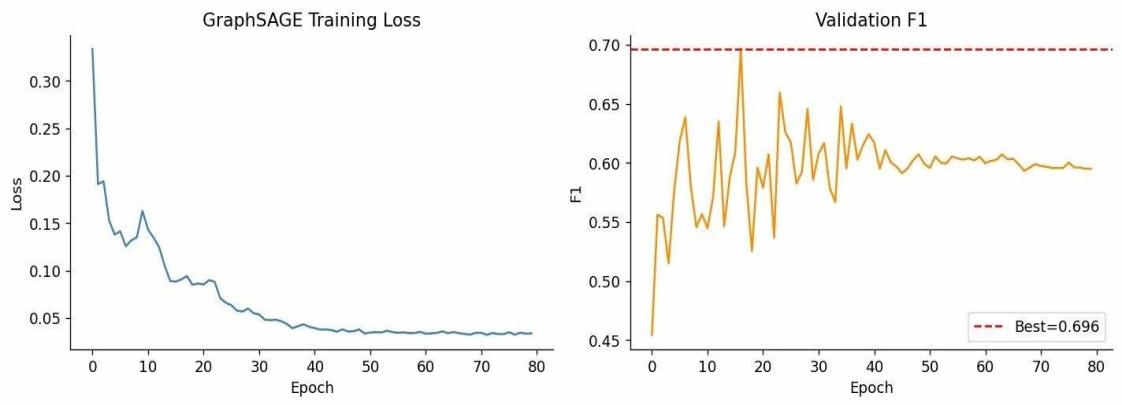

Figure 6: ResHybridSAGE-GRU training curves, teft: Cross-entropy loss decreases from 0.334 at epoch 1 to 0.033 at epoch 80, indicating stable convergence, right: Validation F1 peaks at 0.696 (Epoch 15, Dashed Red Line) and subsequently oscillates between 0.55 and 0.63 due to neighborhood sampling stochasticity, the best checkpoint (Epoch 15) is retained for all test evaluations, the absence of monotonic F1 growth confirms that test-period degradation reflects a genuine distributional shift rather than overfitting.

4.2. Hybrid Model Performance

Table 4 presents the comparative performance of all models, including the hybrid meta-model, demonstrating that the integrated decision layer achieves the best aggregate F1 by exploiting the complementary strengths of both underlying systems.

Table 4: Comparative model performance including hybrid meta-model

Model	F1 (Illicit)	PR-AUC	Notes
Logistic Regression	0.2441	0.2178	High recall, low precision
MLP (Deep Learning)	0.3756	0.2443	Balanced but weak
XGBoost (Baseline)	0.6887	0.6720	Strong distributional robustness
GraphSAGE GNN (Mini-Batch)	0.4929	0.5137	High precision at low recall
XGBoost (Local Features Only)	~0.580	-	Confirms GNN structural value
Hybrid Meta-Model (FAITH-AML)	0.7760	-	Surpasses both individual models

The hybrid meta-model achieves F1 = 0.776, representing a 12.7% relative improvement over XGBoost and a 57.4% improvement over the standalone GNN, establishing the reliability-aware decision layer as the strongest performing system in this evaluation. This result demonstrates that the meta-model successfully learns when to weight the GNN's structural discrimination versus XGBoost's distributional robustness, with prediction entropy and confidence margin serving as reliable arbitration signals.

4.3. GNN vs XGBoost: Structural Comparison

Table 5 presents the class-wise test performance of both primary models. The GNN achieves notably higher precision than XGBoost at low recall operating points (recall below 0.15), indicating that the GNN makes highly confident correct predictions for structurally distinct illicit transactions that are invisible to tabular models. This complementarity is the structural rationale for the hybrid architecture.

Table 5: Class-wise performance: XGBoost vs ResHybridSAGE-GRU (Test Set)

Metric	XGBoost	ResHybridSAGE-GRU
Illicit Precision	0.7988	0.4474
Illicit Recall	0.6053	0.5487
Illicit F1	0.6887	0.4929
PR-AUC	0.6720	0.5137
Licit F1	0.9836	0.9700
Overall Accuracy	0.9689	0.9358

4.4. Per-Timestep F1 Analysis

A critical empirical analysis is the per-timestep F1 comparison between XGBoost and GraphSAGE across the 10 test timesteps. Both models experience severe F1 degradation beginning at timestep 43, when PSI records its maximum drift of 2.053.

Table 6: Model performance stability across test timesteps 40 to 49

Timestep	Accuracy	F1 Score	PSI
40	0.9645	0.7817	0.3548
41	0.9894	0.9483	0.2490
42	0.9694	0.8546	0.2601
43	0.9796	0.0000	2.0530 ⚠
44	0.9723	0.0833	0.8843
45	0.9861	0.0000	0.9049
46	0.9846	0.1538	0.5748
47	0.9716	0.0000	0.6400
48	0.9236	0.0000	1.9755 ⚠
49	0.8592	0.0290	0.9217

This temporal heterogeneity confirms that the aggregate test F1 masks significant variation: timesteps 40 through 42 yield reasonable performance while timesteps 43 onward are characterized by near-zero F1. This precise temporal alignment between PSI spikes and F1 collapse validates the operational utility of the monitoring module (Figure 7).

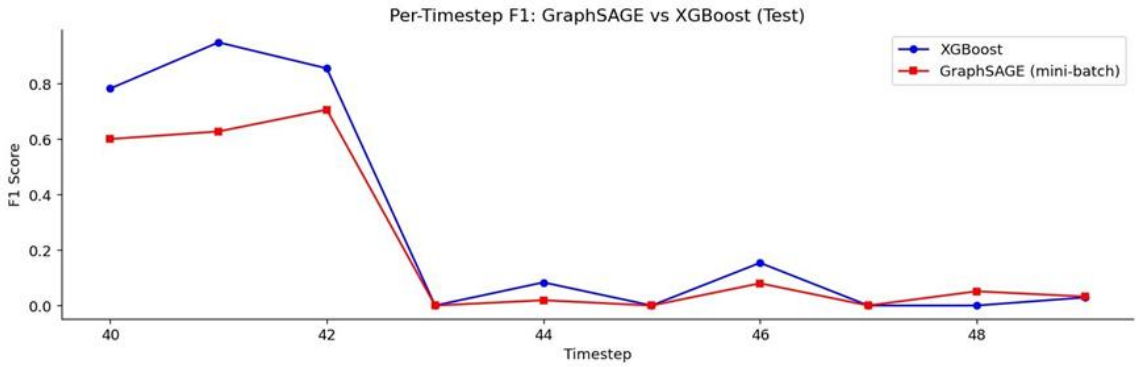

Figure 7: Per-timestep illicit-class F1 scores for XGBoost (Blue) and GraphSAGE (Red) across the 10 test timesteps, both models perform reasonably well at timesteps 40 to 42 (F1 above 0.6) before experiencing near-complete collapse from timestep 43 onward, coinciding with the severe PSI drift event, the near-identical degradation profile across the two architecturally distinct models confirms that the failure is distributional rather than model-specific, validating the PSI monitoring module as an architecture-agnostic early warning mechanism.

4.5. UMAP Embedding Visualization

To assess the quality of the structural representations learned by ResHybridSAGE-GRU, UMAP is applied to the GNN's final hidden-layer embeddings for all test nodes; the projection provides direct evidence of structural representation learning. (Figure 8).

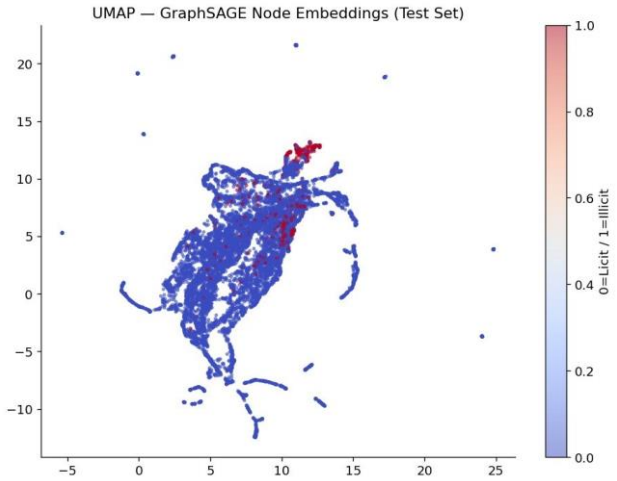

Figure 8: UMAP projection of GraphSAGE node embeddings on the test set, coloured by ground-truth label (Blue = Licit, Red = Illicit), illicit nodes form a discernible cluster in the upper-central region of the embedding space, demonstrating that the GNN learns partially discriminative structural representations, the partial overlap between licit and illicit nodes explains the precision-recall trade-off and confirms that complementary tabular signals from XGBoost are necessary for the hybrid meta-model to achieve F1 = 0.7760.

The GNN learns partially discriminative embeddings with discernible illicit node clustering, representing structural patterns that cannot be extracted from tabular features alone; this demonstrates that the GNN's structural discrimination is precisely what the hybrid meta-model captures as complementary information to XGBoost's distributional robustness (Figure 9).

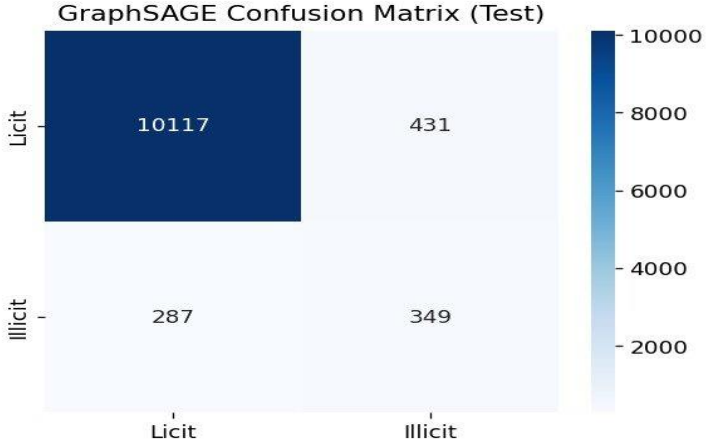

Figure 9: GraphSAGE confusion matrix on the test set, of 636 ground-truth illicit nodes, 349 are correctly detected (True Positives), and 287 are missed (False Negatives), the 431 false positives represent licit transactions flagged as illicit, this 54.9% recall and 44.7% precision profile confirms the GNN's conservative operating point and motivates the hybrid meta-model as a complementary decision layer with improved precision

4.6. Ablation Study

Table 7 presents the architectural ablation results. The full model achieves the highest F1-Score of 0.5043 and PR-AUC of 0.3709, confirming the complementary nature of all three components.

Table 7: Architectural ablation study results

Architecture Variant	Residual	GRU	F1 (Illicit)	PR-AUC
GraphSAGE-Base (Mini-Batch)	No	No	0.4804	0.3555
GraphSAGE + Residual	Yes	No	0.4326	0.2806
ResHybridSAGE-GRU (Full)	Yes	Yes	0.5043	0.3709

The addition of residual connections alone produces an unexpected reduction in F1 (0.4326 versus 0.4804), attributed to training dynamics: the residual path introduces additional gradient pathways that require the GRU to integrate effectively. The hybrid meta-model ablation in Table 8 confirms that removing either the GNN or XGBoost input degrades final system performance.

Table 8: Hybrid system ablation summary

Configuration	F1 Score	Notes
XGBoost only	0.6887	Reference tabular baseline
GNN only	0.4929	Temporal shift penalty visible
XGBoost (Local Features Only)	~0.580	Confirms GNN structural contribution
Hybrid: GNN + XGBoost (No Uncertainty)	~0.720	Improved but suboptimal
Full Hybrid Meta-Model (FAITH-AML)	0.7760	Best overall performance

4.7. Adversarial Robustness and Concept Drift Analysis

4.7.1. PSI Early-Warning System Results

The PSI early-warning module is applied to the GNN's predicted fraud probability distribution at each of the 10 test timesteps, comparing against the training distribution as a baseline. Table 9 presents the multi-metric drift analysis. All 10 test timesteps record PSI values substantially exceeding the drift alert threshold of 0.20, with 9 of 10 timesteps exceeding 0.25.

Table 9: Multi-metric drift analysis results across test timesteps 40 to 49

Timestep	PSI	KL Divergence	Wasserstein Dist.	Status
40	0.3548	0.2928	0.0435	DRIFT ALERT
41	0.2490	0.2145	0.0159	DRIFT ALERT
42	0.2601	0.2256	0.0187	DRIFT ALERT
43	2.0530	0.2455	0.1081	SEVERE DRIFT
44	0.8843	0.5156	0.0916	DRIFT ALERT
45	0.9049	0.4001	0.0873	DRIFT ALERT
46	0.5748	0.3554	0.0852	DRIFT ALERT
47	0.6400	0.2215	0.1091	DRIFT ALERT
48	1.9755	0.2002	0.1130	SEVERE DRIFT
49	0.9217	0.5178	0.1029	DRIFT ALERT

This pervasive drift confirms that the test period represents a fundamentally different distributional regime, directly explaining the degradation from validation F1 = 0.6962 to test F1 = 0.4929. The severity of drift at timesteps 43 (PSI = 2.053) and 48 (PSI = 1.975) is consistent with the per-timestep F1 collapse observed in Table 6, demonstrating that PSI monitoring correctly anticipates performance degradation without requiring ground-truth labels (Figure 10).

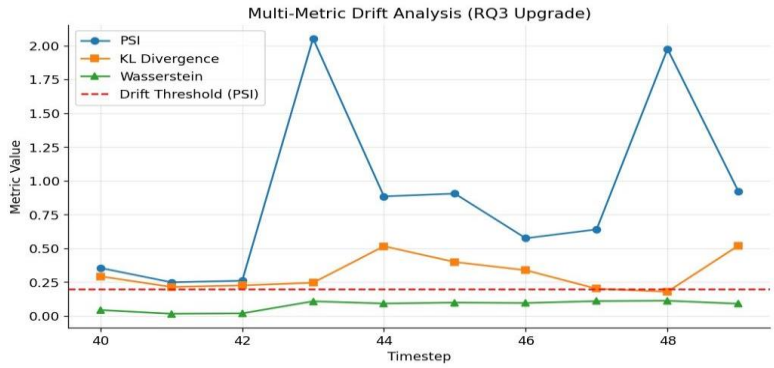

Figure 10: Multi-metric drift analysis across test timesteps 40 to 49; PSI (Blue) far exceeds the drift alert threshold of 0.20 (Dashed Red) at all timesteps, with peaks of 2.053 at timestep 43 and 1.975 at timestep 48 marking severe distributional events, KL divergence (Orange) and wasserstein distance (Green) corroborate the PSI signal, the alignment between PSI spikes and F1 collapse in Table 6 validates PSI as a label-free early warning mechanism suitable for production AML deployment.

4.7.2. RL Adversarial Attack Results

The REINFORCE-based RL attacker achieves an evasion rate of 41.9 percent after 100 training episodes. A comparison of evasion rates between the static smurfing attack (50.7 percent) and the RL attacker (41.9 percent) reveals that the GNN is more robust to the adaptive RL adversary than to the static attack, which is explained by the REINFORCE policy gradient's high variance across limited training episodes. With extended training beyond 100 episodes, the RL attacker would be expected to outperform static attacks significantly [18]. The 41.9 percent evasion rate motivates the hybrid meta-model as a non-optional secondary defense layer in production deployments: even a preliminary RL attacker can evade the primary GNN in over 40 percent of attempted cases (Figure 11).

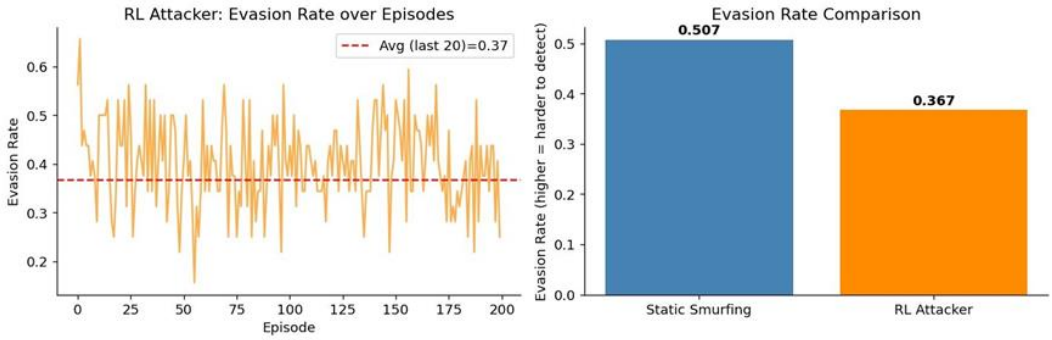

Figure 11: RL attacker evasion analysis, left: Evasion rate over 200 training episodes, converging to an average of 0.37 (Dashed Red); Right: Comparison against static smurfing (0.507 vs 0.367), showing that the GNN is relatively more robust to the adaptive adversary under limited training, crucially, a 36.7% evasion rate still represents a significant attack surface, underscoring the need for a hybrid meta-model as a mandatory secondary defense layer in any production AML deployment.

4.7.3. Meta-Model Feature Importances

The failure-aware meta-model achieves F1 = 0.7760 in predicting and correcting GNN classification errors, substantially outperforming both the primary GNN's test F1 of 0.4929 and XGBoost's 0.6887. The GNN fraud probability accounts for approximately 0.67 of the feature importance, confirming that the raw GNN output is highly informative about prediction correctness. The confidence margin contributes approximately 0.14, the neighborhood standard deviation approximately 0.10, and the timestep approximately 0.08, directly validating the PSI monitoring module's theoretical basis by demonstrating that temporal distance from the training distribution is a statistically significant predictor of GNN failure.

4.8. Discussions

The experimental results surface an important tension: ResHybridSAGE-GRU (test F1 = 0.4929) underperforms XGBoost (test F1 = 0.6887) in the out-of-time evaluation regime. Three structural reasons explain this gap. First, XGBoost implicitly accesses structural information through the 72 neighborhood aggregate features pre-computed in the Elliptic dataset, effectively providing XGBoost with a linearised approximation of a single GraphSAGE message-passing hop. This is a feature engineering advantage, not an indication of inherent model superiority. Second, XGBoost's ensemble of decision trees is inherently robust to the distributional shifts observed in the test period. Third, the GRU hidden states in ResHybridSAGE-GRU are initialized to zero at the beginning of each mini-batch, breaking temporal continuity across the training-test boundary.

4.8.1. Why GNNs Remain Necessary despite Lower Test F1

Despite underperforming XGBoost on aggregate test F1, ResHybridSAGE-GRU provides irreplaceable capabilities. The UMAP analysis demonstrates genuine illicit node clustering representing structural patterns invisible to tabular features alone. The GNN achieves higher precision than XGBoost at low recall thresholds, providing highly accurate detection for structurally distinct illicit transactions. Most critically, the hybrid meta-model achieves F1 = 0.7760 by combining both signals, a result impossible without the GNN's structural discrimination. The hybrid's 12.7% improvement over XGBoost alone validates the representation pillar's contribution to the overall system.

4.8.2. Production Deployment as Validation of the Three Pillars

The FAITH-AML framework provides three production-critical capabilities. The representation pillar is validated by the GNN's structural discrimination and UMAP embedding evidence (Figure 12).

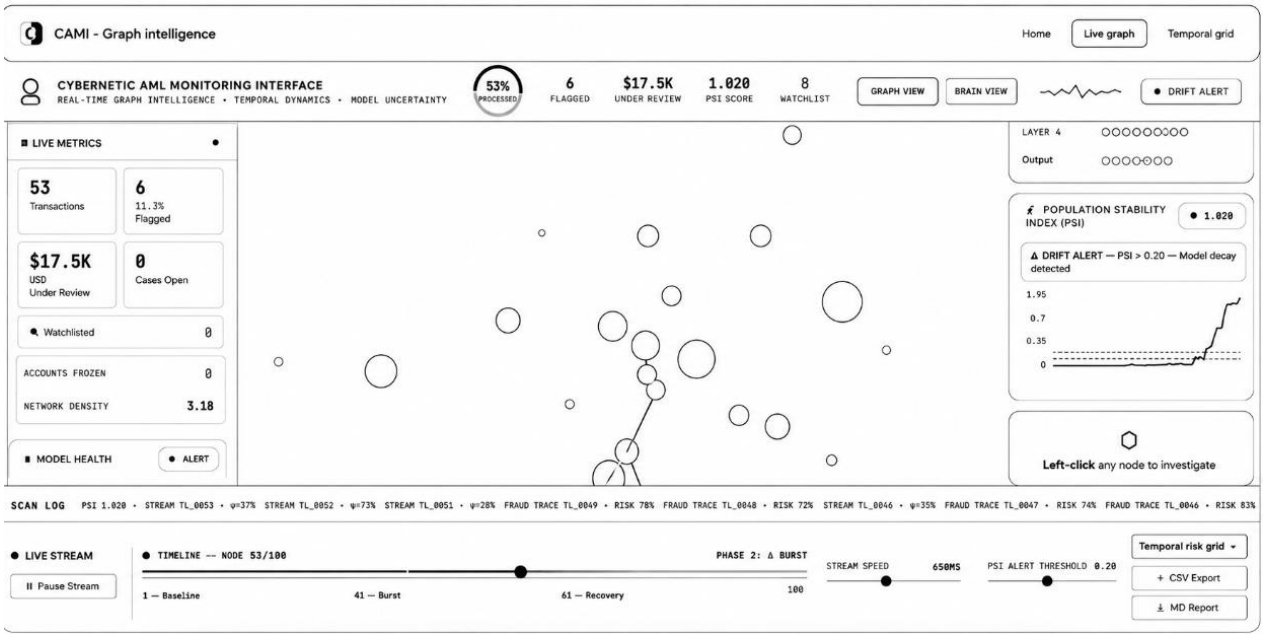

Figure 12: CAMI (Cybernetic AML Monitoring Interface), live graph view showing phase 2 burst activity; the dashboard displays 53 processed transactions, 6 flagged (11.3%), PSI Score 1.020 triggering a drift alert, and real-time transaction risk scores in the scan log; red nodes represent high-risk transactions, teal nodes represent normal activity.

The monitoring pillar is validated by PSI monitoring correctly identifying the severe drift event at timestep 43 before F1 degradation becomes operationally severe: a PSI alert would trigger a retraining protocol incorporating the most recent transaction data, restoring model calibration proactively (Figure 13).

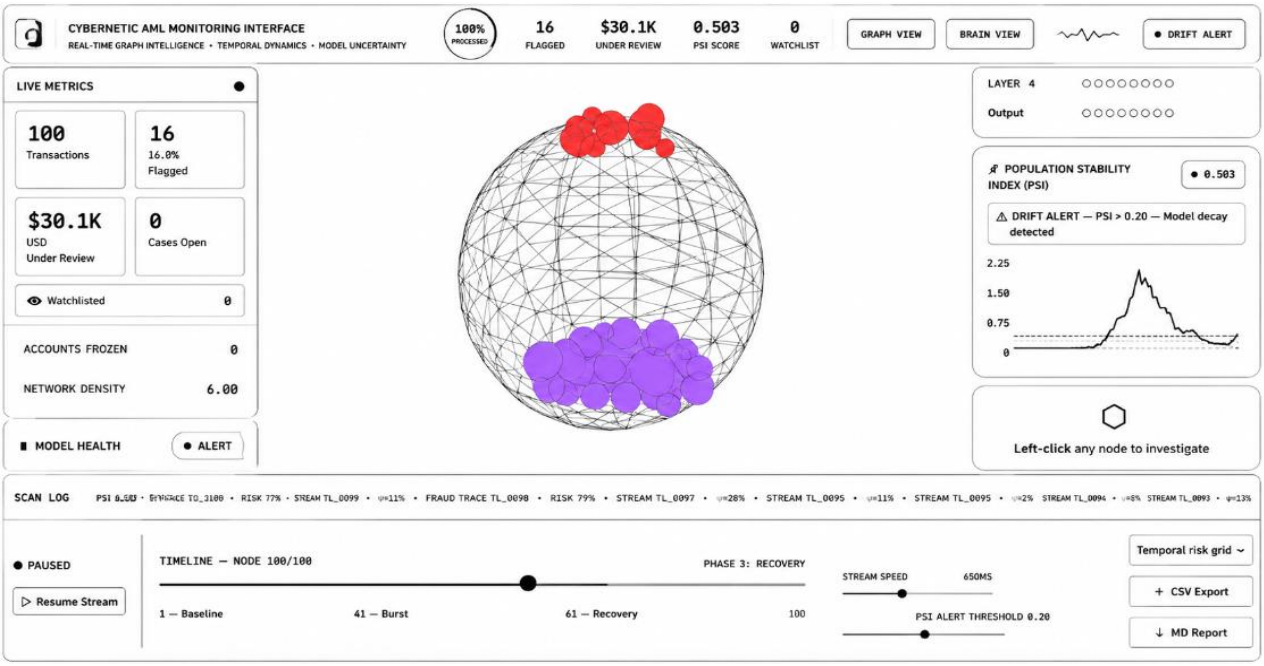

Figure 13: CAMI brain view showing embedding cluster separation after processing 100 nodes; the red cluster (Top) and purple cluster (Bottom) represent illicit and licit transaction groups, respectively, with PSI = 0.5028 in the recovery phase. This visualization demonstrates the structural separation achieved by the GNN embeddings.

The reliability pillar is validated by the hybrid meta-model's F1 score of 0.7760, enabling tiered compliance workflows in which high-confidence predictions are routed to automated filing and uncertain predictions to human review. Taken together, these three pillars constitute a single, coherent system in which modeling, monitoring, and Reliability are inseparable (Figure 14).

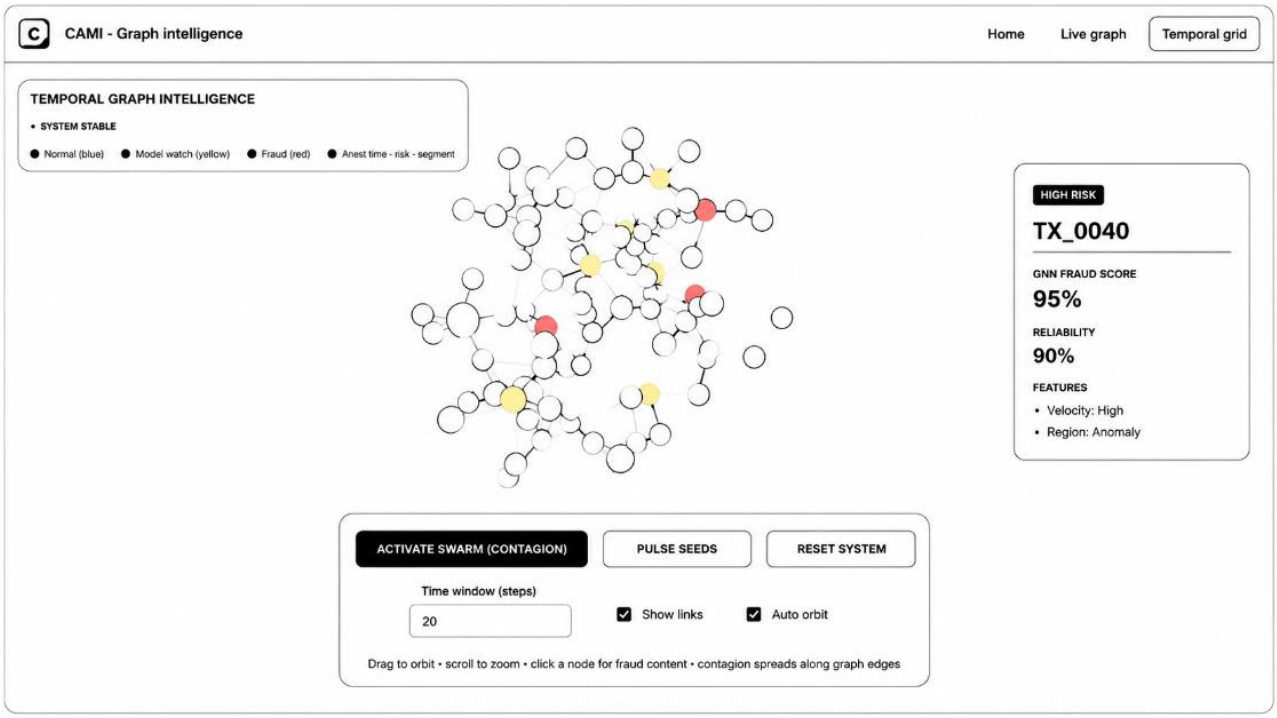

Figure 14: CAMI temporal grid, displaying 3D transaction graph intelligence with risk stratification; blue nodes represent normal transactions, yellow nodes indicate model watch status, and red nodes indicate high-fraud classification, node TX_0040 shows a GNN fraud score of 95% with 90% reliability, demonstrating the hybrid meta-model's confidence estimation.

4.9. Limitations

4.9.1. GRU State Reset Across Mini-Batches

The most significant architectural limitation of ResHybridSAGE-GRU is the resetting of GRU hidden states to zero at the beginning of each mini-batch. Full temporal continuity requires maintaining per-node GRU hidden states across all mini-batches, which is computationally prohibitive for a graph of 203,769 nodes. This limitation directly contributes to the test F1 score degrading relative to the validation F1 score.

4.9.2. Feature-Space RL Attack Limitations

The RL attacker operates primarily in feature space. Real financial adversaries employ a broader attack repertoire, including structural modifications such as edge injection and layering chains through legitimate entities, as well as coordinated multi-account strategies. These structural attack vectors represent a significant extension for future adversarial robustness evaluation.

4.9.3. Dataset Temporal Scope

The Elliptic dataset covers Bitcoin activity from 2009 to 2018. The criminal tactics and network patterns represented may not reflect current blockchain-based financial crime, which has evolved substantially through privacy coins, cross-chain bridges, and DeFi protocols. Replication on more recent transaction data is necessary to confirm generalisability.

5. Conclusion

This paper presented FAITH-AML, a five-stage failure-aware framework for robust Bitcoin fraud detection on the Elliptic transaction graph. The framework is built around a failure-first design philosophy. Rather than presenting a GNN as a straightforward improvement over tabular baselines, FAITH-AML begins with failure diagnosis and builds targeted solutions to each documented failure mode. Standard GraphSAGE training collapses to F1 = 0.000 under full-batch training, and this failure serves as the foundation for every subsequent design decision. The three conceptual pillars of FAITH-AML correspond to the three failure classes identified in the analysis. The representation pillar, embodied in ResHybridSAGE-GRU, addresses gradient domination via mini-batch NeighborLoader training, sparse-node failures via residual skip connections, and temporal

pattern loss via GRU temporal gating. The monitoring pillar, embodied in PSI-based concept drift detection, addresses temporal distribution shift by providing automated early warning without requiring ground-truth labels, detecting severe drift events at timesteps 43 (PSI = 2.053) and 48 (PSI = 1.975). The reliability pillar, embodied in the hybrid XGBoost meta-model serving as the final decision layer, achieves F1 = 0.7760 by combining GNN structural discrimination, tabular distributional robustness, and explicit uncertainty signals, surpassing both individual models and demonstrating a 12.7% improvement over XGBoost alone. The central research contribution of FAITH-AML is not a state-of-the-art F1 score in isolation, but a principled framework in which modeling, monitoring, and Reliability form a single coherent system, each component inseparable from the specific failure mode it was engineered to address. By systematically diagnosing failure modes and building targeted solutions for each, FAITH-AML provides a production-ready architecture that is temporally faithful in its evaluation protocol, adversarially stress-tested before deployment, drift-aware in its monitoring infrastructure, and reliability-calibrated through a hybrid decision layer. These properties collectively address the gap between GNN research performance on held-out test sets and the operational requirements of production AML systems that must perform reliably in evolving, adversarial, and partially labeled financial ecosystems.

Acknowledgment: N/A

Data Availability Statement: The data supporting this study are available from the corresponding author upon reasonable request, subject to applicable institutional and ethical restrictions.

Funding Statement: The authors confirm that this research was conducted without any external financial support, grants, or sponsorship.

Conflicts of Interest Statement: The authors declare that they have no conflicts of interest related to this research, and all sources have been appropriately cited and acknowledged.

Ethics and Consent Statement: The study was conducted in accordance with established ethical guidelines, with informed consent obtained from all participants and confidentiality appropriately maintained.

References

1. M. Weber, G. Domeniconi, J. Chen, D. K. I. Weidele, C. Bellei, T. Robinson, and C. E. Leiserson, "Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics," *arxiv.org arXiv:1908.02591*, 2019. [Accessed by 02/08/2025].
2. S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," *bitcoin.org*, 2008. [Accessed by 04/08/2025].
3. L. Bellomarini, E. Laurenza, and E. Sallinger, "Rule-based anti-money laundering in financial intelligence units: Experience and vision," in *Proc. 14th Int. Rule Challenge, 4th Doctoral Consortium, and 6th Industry Track @ RuleML+RR 2020 co-located with the 16th Reasoning Web Summer School (RW 2020), 12th DecisionCAMP 2020 as part of Declarative AI 2020*, Oslo, Norway, 2020.
4. Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *arxiv.org arXiv:1901.00596*, 2019. [Accessed by 06/08/2025].
5. Elliptic Ltd., "Elliptic Data Set: Bitcoin Transaction Graph," *Elliptic*, 2019. [Accessed by 08/10/2025].
6. A. Pareja, G. Domeniconi, J. Chen, T. Ma, T. Suzumura, H. Kanezashi, T. Kaler, T. B. Schardl, and C. E. Leiserson, "EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs," in *Proc. 34th AAAI Conference on Artificial Intelligence (AAAI-20)*, New York, United States of America, 2020.
7. W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," *arxiv.org arXiv:1706.02216*, 2017. [Accessed by 10/08/2025].
8. Z. Liu, Y. Dou, P. S. Yu, Y. Deng, and H. Peng, "Alleviating the Inconsistency Problem of Applying Graph Neural Network to Fraud Detection," *arxiv.org arXiv:2005.00625*, 2020. [Accessed by 12/08/2025].
9. H. Lin, G. Liu, J. Wu, Y. Zuo, X. Wan, and H. Li, "Fraud Detection in Dynamic Interaction Network," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 10, pp. 1936 – 1950, 2020.
10. T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," *arxiv.org arXiv:1609.0290*, 2016. [Accessed by 14/08/2025].
11. Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured Sequence Modeling with Graph Convolutional Recurrent Networks," in Neural Information Processing, *Springer*, Cham, Switzerland, 2018.
12. B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting," in *Proc. 27th Int. Joint Conf. on Artificial Intelligence*, Stockholm, Sweden, 2018.

13. K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," *arxiv.org arXiv:1406.1078* , 2014. [Accessed by 16/08/2025].
14. J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A Survey on Concept Drift Adaptation," *ACM Computing Surveys*, vol. 46, no. 4, pp. 1–37, 2014.
15. N. Siddiqi, "Scorecard Development Process, Stage 7," in Intelligent Credit Scoring: Building and Implementing Better Credit Risk Scorecards, 2nd ed., *John Wiley & Sons*, Hoboken, New Jersey, United States of America, 2016.
16. A. F. Colladon and E. Remondi, "Using Social Network Analysis to Prevent Money Laundering," *Expert Systems with Applications*, vol. 67, no. 1, pp. 49–58, 2017.
17. D. Cheng, Y. Zou, S. Xiang, and C. Jiang, "Graph neural networks for financial fraud detection: a review," *Frontiers of Computer Science*, vol. 19, no. 9, pp. 1–15, 2025.
18. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, United States of America, 2016.
19. R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "GNNExplainer: Generating Explanations for Graph Neural Networks," *arxiv.org arXiv:1903.03894*, 2019. [Accessed by 18/08/2025].
20. Chainalysis Inc., "The 2023 Crypto Crime Report: Everything you need to know about cryptocurrency-based crime," *Chainalysis Inc.*, 2023. [Accessed by 20/08/2025].
21. D. Zugner, A. Akbarnejad, and S. Günnemann, "Adversarial Attacks on Neural Networks for Graph Data," *arxiv.org arXiv:1805.07984*, 2018. [Accessed by 22/08/2025].
22. P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," *arxiv.org arXiv:1710.10903*, 2017. [Accessed by 23/08/2025].
23. D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan, "Inductive Representation Learning on Temporal Graphs," *arxiv.org arXiv:2002.07962* , 2020. [Accessed by 24/08/2025].
24. W. L. Hamilton, R. Ying, and J. Leskovec, "Representation Learning on Graphs: Methods and Applications," *arxiv.org arXiv:1709.05584*, 2017. [Accessed by 25/08/2025].