Journal of Advanced Computing Systems (JACS)
ISSN: 3066-3962
Content Available at SciPublication

Graph-Based Feature Learning for Anti-Money Laundering in Cross-Border Transaction Networks

Yutong Huang
Financial Statistics & Risk Management, Rutgers University, NJ, USA

DOI: 10.69987/JACS.2024.40704

Keywords

Anti-money laundering, graph neural networks, feature learning, cross-border transactions

Abstract

The increasing sophistication of cross-border money laundering activities poses significant challenges to traditional detection systems. This paper presents a graph-based feature learning framework specifically designed to identify suspicious transactions in complex, cross-border financial networks. The proposed approach constructs heterogeneous transaction graphs that incorporate multiple entity types and relationship patterns, then applies advanced graph neural architectures to automatically learn discriminative features that capture both spatial transaction structures and temporal behavioral dynamics. The framework addresses critical challenges, including extreme class imbalance, high false positive rates, and multi-jurisdictional complexity through specialized feature extraction mechanisms and adaptive learning strategies. An experimental evaluation of the Elliptic Bitcoin dataset and the IBM IT-AML synthetic banking dataset demonstrates substantial improvements over traditional handcrafted features and baseline graph methods, achieving an 87.3% F1-score with a 73% reduction in false positive rates (from 38.2% to 10.3%). The learned features reveal interpretable patterns in cross-border layering schemes and currency exchange anomalies, providing actionable insights for financial crime investigators.

1. Introduction

1.1. Background and Motivation

Financial institutions worldwide process trillions of dollars in cross-border transactions annually, creating extensive opportunities for sophisticated money laundering operations. The Financial Action Task Force estimates that 2-5% of global GDP, approximately $800 billion to $2 trillion, is laundered each year through complex international networks. Traditional anti-money laundering systems rely heavily on rule-based alert generation, producing overwhelming volumes of false positives that consume investigative resources while missing genuinely suspicious activities[1]. The inherent limitations of manually crafted rules become increasingly apparent as criminal organizations exploit regulatory arbitrage across jurisdictions and employ multi-layered transaction schemes.

Machine learning approaches have demonstrated potential for improving detection accuracy through their pattern recognition capabilities [2]. The fundamental challenge lies in effectively representing features that capture the complex relational structures and temporal dynamics inherent in financial transaction networks. Handcrafted features based on transaction statistics and basic network measures fail to encode sophisticated laundering patterns involving multiple intermediaries, strategic timing, and cross-border routing. Graph-based learning methods offer promising directions by treating financial transactions as interconnected networks, where structural patterns and entity relationships provide crucial signals for identifying illicit activities [3].

1.2. Problem Statement and Challenges

The anti-money laundering detection problem in cross-border networks presents several fundamental challenges that existing approaches inadequately addressClass imbalance represents a primary obstacle, with illicit transactions typically comprising less than 0.5% of total transaction volumes in real-world datasets. Current monitoring systems exhibit alert precision below 5% (equivalently, a false discovery rate exceeding 95%), meaning that fewer than 5 out of 100 alerts correspond to actual suspicious activities. This

differs from the sample-level false positive rate, which measures the proportion of legitimate transactions incorrectly flagged. This inefficiency stems from overly broad rule specifications, designed to ensure regulatory compliance, combined with an insufficient understanding of transaction patterns in their context. Cross-border complexity manifests through multiple dimensions, complicating detection efforts. Jurisdictional differences create inconsistent reporting requirements and varying definitions of suspicious activity across countries. Currency exchange transactions introduce additional noise and create opportunities for value obfuscation through strategic timing of conversions. Temporal dispersion across time zones enables the coordination of transaction sequences that appear unrelated when analyzed within single-jurisdiction timeframes.

1.3. Research Objectives and Contributions

This research develops a comprehensive graph-based feature learning framework specifically tailored for cross-border anti-money laundering detection. The primary objective focuses on automatic discovery of discriminative features that effectively encode complex transaction patterns while maintaining computational efficiency for large-scale deployment. The main contributions include a novel heterogeneous graph construction methodology that integrates multiple entity types, relationship categories, and temporal dynamics into unified network representations. An adaptive feature learning mechanism employing graph neural architectures with attention-weighted message passing captures both local neighborhood patterns and multi-hop structural information. A comprehensive evaluation framework utilizing real-world cryptocurrency transaction data and realistic synthetic banking scenarios validates the effectiveness of learned features compared to handcrafted alternatives. The experimental results demonstrate substantial performance improvements across multiple metrics while revealing interpretable patterns that align with known money laundering typologies.

2. Related Work

2.1. Traditional Anti-Money Laundering Approaches

Historical anti-money laundering systems have predominantly employed rule-based detection mechanisms that generate alerts when transactions exceed predefined thresholds or match suspicious activity patterns [4]. These systems implement static rules derived from regulatory guidance and historical case analysis, creating alerts for scenarios such as rapid movement of funds through multiple accounts, transactions just below reporting thresholds, or unusual geographic patterns. Statistical methods augment rule-based systems by identifying anomalies in transaction features such as amounts, frequencies, and timing patterns. Transaction frequency analysis examines the number of operations within specific time windows, identifying accounts that exhibit sudden spikes in activity. Network centrality measures quantify the structural importance of entities within transaction graphs, highlighting accounts that serve as hubs for the movement of money. Feature engineering represents a critical bottleneck in traditional approaches, as it requires domain expertise to manually specify potentially relevant transaction characteristics.

2.2. Machine Learning for Financial Crime Detection

Supervised learning algorithms offer enhanced flexibility over rule-based systems by learning decision boundaries from labeled training examples[5]. Random Forest classifiers construct ensembles of decision trees, each trained on bootstrap samples of the data with random feature subsets, providing robust predictions through aggregation. XGBoost implements gradient boosting to iteratively refine predictions by focusing on instances that were previously misclassified. Anomaly detection techniques address the challenge of limited labeled data by identifying transactions that deviate from normal behavioral patterns[6]. Isolation Forest algorithms recursively partition the feature space, exploiting the principle that anomalies require fewer splits for isolation compared to normal instances. Ensemble methods combine predictions from multiple models to improve robustness and accuracy[7]. Class imbalance handling techniques prove essential for anti-money laundering applications, given the extreme scarcity of illicit transactions. Synthetic Minority Over-sampling Technique generates artificial examples of suspicious transactions through interpolation in feature space.

2.3. Graph Neural Networks in Anti-Money Laundering

Graph Convolutional Networks extend deep learning to graph-structured data by defining convolution operations that aggregate information from a node's neighborhood [8]. Message passing mechanisms enable nodes to iteratively update their representations by incorporating features from connected neighbors, effectively propagating information across network structures. Graph Attention Networks introduce weighted neighborhood aggregation, learning attention coefficients that identify the most relevant connections for each node. This mechanism proves particularly valuable in financial networks where transaction relationships exhibit varying levels of importance. Heterogeneous Graph Neural Networks address the

complexity of financial networks that contain multiple entity and relationship types [9]. Relational Graph Convolutional Networks define type-specific transformation matrices, learning distinct aggregation functions for different edge categories. Recent advances introduce dynamic graph methods that incorporate temporal evolution, modeling how transaction patterns change over time through recurrent neural components[10].

3. Methodology

3.1. Graph Construction for Cross-Border Transaction Networks

The foundation of the proposed framework rests on comprehensive graph construction that captures the multi-faceted nature of cross-border financial transactions. For the IBM IT-AML dataset, accounts are represented as nodes, with transactions modeled as directed edges. For the Elliptic Bitcoin dataset, transactions themselves serve as nodes, with edges representing Bitcoin flows between transactions, creating a representation where structural patterns reveal hidden relationships and behavioral anomalies. The heterogeneous formulation distinguishes between individual accounts, business accounts, and intermediary institutions, recognizing that different entity types exhibit distinct transaction characteristics and laundering risk profiles[11]. Edge types receive differentiation to encode the diverse nature of financial relationships. Domestic transfers represent transactions within single jurisdictions, subject to uniform regulatory frameworks and reporting requirements. Cross-border transfers introduce complexity through currency conversions, international wire protocols, and multiple regulatory jurisdictions. Currency exchange operations create specialized edges that connect accounts in different currencies, capturing opportunities for value obfuscation through the strategic timing of conversions.

For the IBM IT-AML dataset, which has rich metadata, attribute enrichment enhances the basic graph structure. Attribute enrichment augments the basic graph structure with contextual information critical for cross-border analysis. Know Your Customer data provides entity characteristics, including registration jurisdiction, business category, and historical risk ratings. Geographic information encodes physical locations and economic relationships between jurisdictions, enabling the detection of unusual routing patterns. Temporal dynamics integration transforms the static graph representation into a dynamic structure capturing behavioral evolution. Time-stamped transaction edges preserve exact timing information, enabling analysis of velocity patterns and coordinated activity sequences. Rolling time windows segment the transaction history into overlapping intervals, supporting both short-term anomaly detection and long-term trend analysis. The temporal edge features quantify time-dependent characteristics including transaction velocity measuring the rate of value movement through account chains.

Note: These enrichment features (KYC data, geographic information, currency attributes) are available only in the IBM IT-AML dataset. For Elliptic, we rely solely on the provided 166-dimensional feature vectors and temporal structure.

Table 1: Dataset Statistics and Characteristics

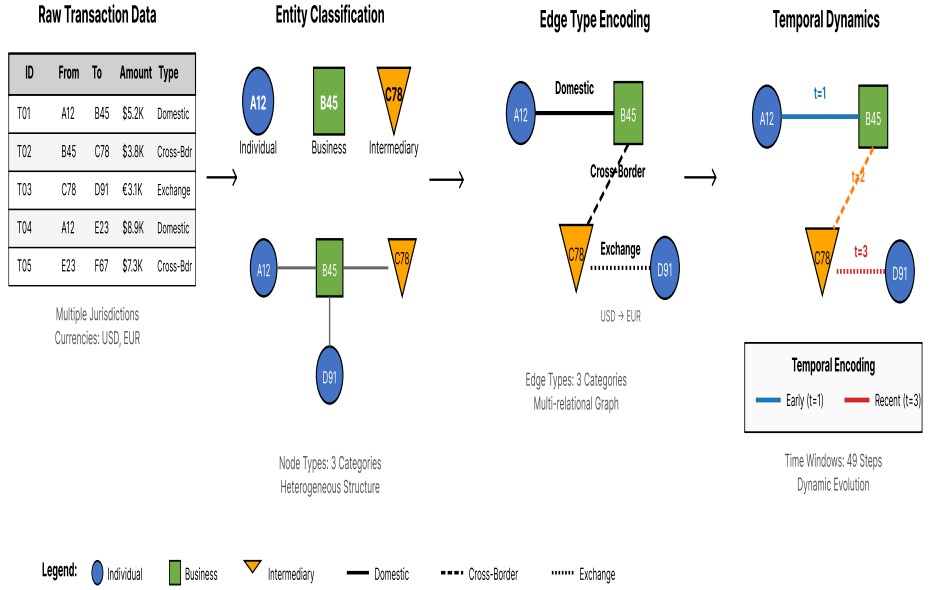
Dataset	Nodes	Edges	Illicit % (labeled)	Unknown % (full)	Regions (jurisdiction)	Currencies	Temporal span	Cross-border %
Elliptic Bitcoin	203,769	234,355	≈9.8%	≈77.1%	N/A (no jurisdiction metadata)	BTC	49 time steps	N/A
IBM IT-AML Synthetic	156,482	892,441	0.43%	0%	8 regions	USD, EUR, GBP	180 days	32.7%

Cross-border analysis and jurisdictional features are only applicable to the IBM IT-AML dataset, which contains explicit jurisdiction and currency metadata. Elliptic Bitcoin transactions lack jurisdictional attributes.

Table 2: Handcrafted Feature Categories

Category	Features	Description
Transaction Statistics	Amount mean/std, Frequency, Velocity	Basic transaction volume and timing metrics
Network Centrality	Degree, Betweenness, PageRank	Structural importance measures
Temporal Patterns	Time gaps, Periodicity score, Burst detection	Sequential timing characteristics
Cross - Border	Jurisdiction diversity, Currency pairs, Exchange frequency	International transaction indicators
KYC Attributes	Risk rating, Business type, Registration age	Entity profile features

Figure 1: Heterogeneous Transaction Graph Construction Pipeline



The graph construction visualization demonstrates the transformation of raw transaction data into structured heterogeneous networks. The pipeline flows from left to right through four main stages. The leftmost panel displays a sample of raw transaction records in tabular format with columns including transaction id, source_account, destination_account, amount, currency, timestamp, and jurisdiction. The table shows heterogeneous transaction types mixing domestic and international transfers. The second panel illustrates entity type classification, where accounts are categorized into three types indicated by different node shapes: circles for individual accounts in blue, squares for business accounts in green, and triangles for financial intermediaries in orange. Node sizes correlate with transaction volumes. The third panel illustrates edge type differentiation, featuring three distinct edge styles: solid lines for domestic transfers within jurisdictions, dashed lines for cross-border transfers between jurisdictions, and dotted lines with currency labels for exchange operations. Edge thickness corresponds to transaction amounts. The rightmost panel presents the complete heterogeneous graph with temporal annotations, displaying timestamp labels on edges and color gradients that indicate transaction recency, ranging from dark blue for older transactions to bright red for recent ones. The background features light-shaded regions representing different jurisdictions, with darker shading indicating higher-risk zones. Small icons depict attribute enrichment, including KYC verification status, regulatory flags, and geographic markers. Dataset-Specific Modeling: The graph construction differs between datasets. For IBM IT-AML with account-level data, we aggregate transactions at the

account node level. For Elliptic's transaction-level structure, each node represents an individual transaction, and we extract account-equivalent features through neighborhood aggregation patterns.

3.2. Graph-Based Feature Learning Architecture

The feature learning architecture employs multi-layer graph convolutional structures to progressively extract increasingly abstract representations from the transaction network. The aggregation process at each layer combines information from node neighborhoods through carefully designed functions that preserve important structural properties while filtering noise. The fundamental operation updates node representations through weighted combinations of neighbor features, mathematically expressed as:

$$h_i^{(l+1)} = \sigma(W^{(l)} \cdot AGG(\{h_j^{(l)} : j \in N(i)\}))$$

where $h_i^{(l)}$ denotes the feature vector for node i at layer l, N(i) represents the neighborhood of node i, $W^{(l)}$ is a learnable weight matrix, AGG is the aggregation function, and σ represents a nonlinear activation[12]. Attention-weighted message passing introduces learnable coefficients that emphasize the most relevant connections for each node. The attention mechanism computes importance scores for each neighbor based on the compatibility between source and target node features:

$$\alpha_{ij} = \text{softmax}_j(\text{LeakyReLU}(a^T [W h_i \| W h_j]))$$

where α_{ij} represents the attention coefficient from node j to node i, a is a learnable attention vector, || denotes concatenation, and the softmax ensures normalized weights across all neighbors. Multiple attention heads learn complementary patterns, with each head focusing on different aspects of the transaction relationships. Residual connections address the challenge of training deep graph neural networks by providing direct paths for gradient flow:

$$h_i^{(l+1)} = h_i^{(l)} + F(h_i^{(l)}, \{h_j^{(l)} : j \in N(i)\})$$

where the softmax normalization is computed over all neighbors $j \in N(i)$, ensuring $\Sigma_{j \in N(i)} \alpha_{ij} = 1$ for each target node i.

Cross-border specific feature learning incorporates specialized components addressing the unique characteristics of international transactions. Jurisdictional embedding learning creates low-dimensional representations for different countries and regulatory zones, capturing systematic differences in transaction patterns and risk profiles. Currency exchange pattern encoding employs specialized attention mechanisms for transactions involving currency conversions. Multi-hop path feature extraction explicitly aggregates information along extended transaction chains, detecting patterns where individual transactions appear legitimate but sequential combinations reveal suspicious structures. Temporal feature integration augments the spatial graph structure with sequential pattern modeling capabilities. Long Short-Term Memory units process ordered transaction sequences for each account, capturing dependencies between temporally adjacent operations:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

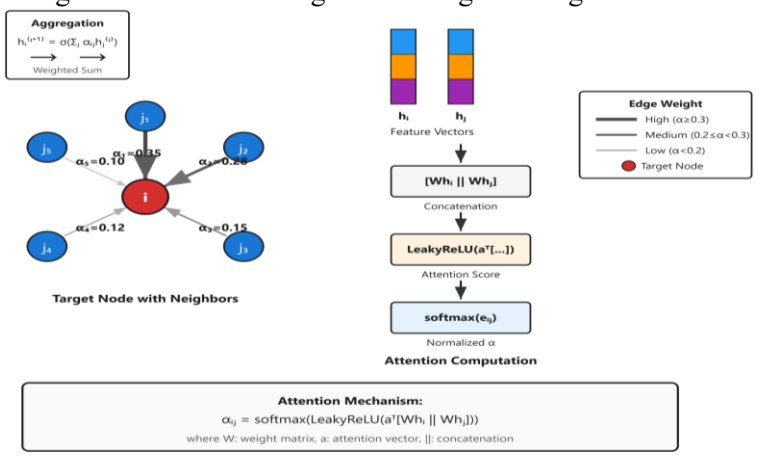
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

$$h_t = o_t \odot \tanh(c_t)$$

where f_t, i_t, o_t represent forget, input, and output gates, c_t denotes the cell state, and $\odot$ indicates element-wise multiplication.

Figure 2: Attention-Weighted Message Passing Mechanism



This technical diagram illustrates the attention mechanism for neighborhood aggregation in the graph neural network architecture. The central focus displays a target node i in red surrounded by its k-hop neighborhood, with neighbor nodes j1 through j5 arranged in a circular pattern. Each neighbor node connects to the target through directed edges labeled with computed attention coefficients α i1 through α i5, shown as decimal values between 0 and 1. The coefficient magnitudes are visually encoded through edge thickness and color intensity, with thicker, darker edges indicating higher attention weights. The right panel shows the attention computation process through a series of transformation steps. The top shows input feature vectors h i and h j as vertical bars with color-coded segments representing different feature dimensions. The middle section displays the concatenation operation with a binding symbol, followed by the learned attention vector a multiplication. The bottom shows the LeakyReLU activation function graph and the final softmax normalization across all neighbors. A mathematical formula box presents the full attention equation with clear notation definitions. The left panel illustrates multi-head attention through three parallel attention computation streams colored in blue, green, and purple, each producing different attention patterns visualized as heatmaps over the neighborhood structure. The final output displays the concatenated multi-head representations before the addition of the residual connection.

3.3. Feature Fusion and Classification Framework

Multi-scale feature aggregation combines representations extracted at different granularities to create comprehensive account profiles. Node-level features capture individual account characteristics and immediate neighborhood patterns, encoding direct transaction relationships and first-order network properties. Subgraph-level features aggregate information from local communities surrounding each account, identifying coordinated activity patterns involving multiple interconnected entities. Graph-level features encode global network statistics and cross-account patterns. Hierarchical pooling strategies progressively coarsen the graph representation through iterative aggregation. The pooling operations identify important substructures while reducing computational complexity, creating multi-resolution representations.

The feature fusion mechanism combines graph-learned representations with traditional handcrafted features through principled integration strategies. Concatenation provides a straightforward approach by stacking feature vectors, allowing the classification layer to learn optimal weightings. Attention-based feature weighting offers more sophisticated integration by computing importance scores for different feature groups:

$$\beta_k = \exp(w_k^T f_k) / \Sigma_j \exp(w_j^T f_j)$$

$$f_{fused} = \Sigma_k \beta_k f_k$$

where β k represents the attention weight for feature group k, f_k denotes the feature vector for group k, and w k is a learnable weight vector. Classification layer design employs multi-layer perceptrons with carefully tuned regularization to prevent overfitting on the limited suspicious transaction examples. Dropout regularization randomly deactivates neurons during training, forcing the network to learn robust features. Focal loss addresses class imbalance by down-weighting well-classified examples:

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t)$$

where p_t is the model's predicted probability for the ground truth class, $\alpha_t \in \{\alpha, 1-\alpha\}$ is a class-balancing weight (we set $\alpha=0.75$ to emphasize the minority illicit class), and $\gamma=2$ is the focusing parameter that down-weights easy examples. The logarithm uses the natural base.

4. Experimental Setup and Evaluation

4.1. Datasets and Preprocessing

The Elliptic Bitcoin dataset provides real-world cryptocurrency transaction data with ground-truth labels for detecting illicit activity. The dataset contains 203,769 transaction nodes and 234,355 directed edges representing Bitcoin flows between addresses. Each transaction receives one of three labels: illicit, licit, or unknown, with illicit transactions comprising ≈9.8% of the labeled instances (4,545/46,564). The dataset contains three label categories: illicit (4,545 transactions), licit (42,019 transactions), and unknown (157,205 transactions). Following standard practice, we exclude unknown transactions during training and evaluation to focus on supervised learning performance. The unknown category accounts for 77.1% of total nodes, representing transactions without confirmed labels. Using the full graph as the denominator (including 157,205 unknown nodes), the illicit share is ≈2.23% (4,545/203,769); restricting to labeled nodes, it is ≈9.8%. The temporal structure spans 49 time steps, representing aggregated transaction activity in successive time windows. Feature vectors include 166 dimensions combining local node information and temporal aggregations. Preprocessing procedures standardize feature scales and construct appropriate training, validation, and test splits. Feature normalization applies z-score standardization to each dimension independently. The data splitting strategy employs temporal partitioning to simulate realistic

deployment scenarios where models must predict future transactions based on historical training data. The first 34 time steps provide training data, time steps 35-42 form the validation set, and the final 7 time steps serve as the test set.

The IBM IT-AML synthetic banking dataset simulates realistic cross-border money laundering scenarios through agent-based modeling. The dataset contains 156,482 accounts and 892,441 transactions distributed across 8 simulated jurisdictions. Multiple currencies, including USD, EUR, and GBP, circulate through the network, with currency exchange operations comprising 8.3% of total transactions. The illicit transaction percentage reaches 0.43%, reflecting the extreme imbalance characteristic of production systems. Data augmentation techniques address the scarcity of suspicious transaction examples through carefully designed synthetic generation procedures. Transaction amounts receive multiplicative noise within realistic ranges, timing patterns shift by random offsets, and intermediate nodes are occasionally added or removed from known layering chains. Dataset statistics reveal significant differences in network topology between the two evaluation benchmarks. The Bitcoin network exhibits a higher average node degree at 2.3 connections per node compared to 1.8 for the banking simulation. To ensure robustness, we repeat all experiments 5 times with different random initializations while maintaining the same temporal split, and report mean ± standard deviation across runs.

Table 3: Graph Neural Network Architecture Hyperparameters

Component	Configuration	Values
GCN Layers	Number of layers	3
	Hidden dimensions	[128, 64, 32]
	Activation function	LeakyReLU α=0.2
	Dropout rate	[0.3, 0.4, 0.5]
Attention	Number of heads	4
	Attention dropout	0.2
	Concatenation strategy	Multi-head concat
LSTM	Hidden size	64
	Number of layers	2
	Bidirectional	True
Classification	MLP layers	[128, 64]
	Final dropout	0.5
	Loss function	Focal Loss γ=2,α=0.75
Training	Optimizer	Adam lr=0.001
	Batch size	512
	Maximum epochs	100
	Early stopping patience	15

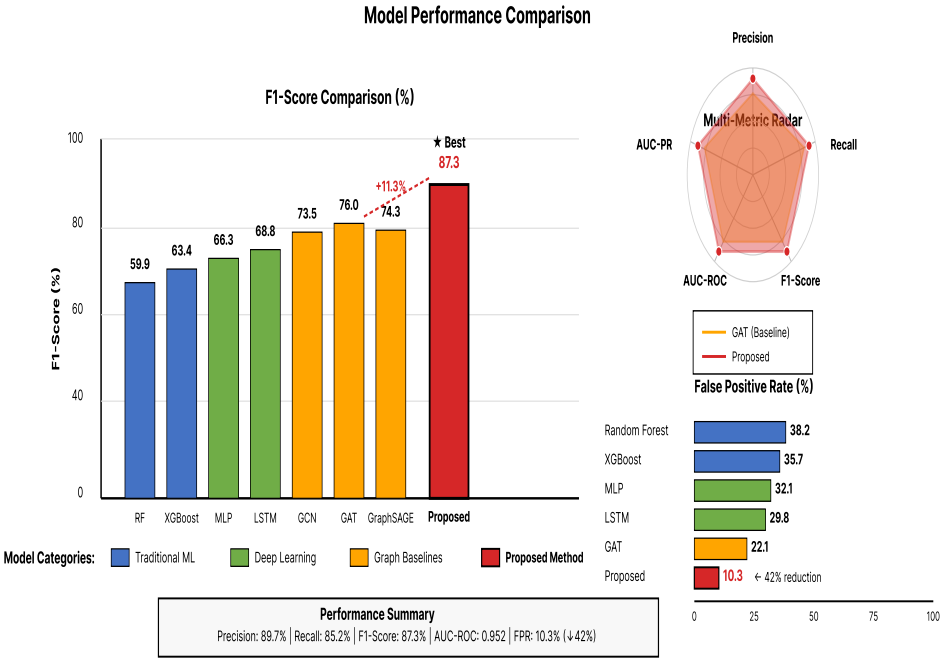
4.2. Baseline Methods and Comparison

Traditional machine learning baselines establish performance references using handcrafted features derived from domain expertise. Random Forest configurations employ 200 trees with a maximum depth of 15 and a minimum samples per leaf set to 10. The feature set comprises 47 handcrafted dimensions,

encompassing transaction statistics, network centrality measures, temporal patterns, and cross-border indicators. XGBoost optimization uses gradient boosting with a learning rate of 0.1, a maximum depth of 6, and 150 estimators. Logistic Regression with L2 regularization provides a linear baseline, with the regularization parameter C set to 0.1 through cross-validation. Deep learning baselines explore non-graph neural approaches to assess the specific value of exploiting network structure. Multi-Layer Perceptron architectures employ three hidden layers with dimensions [256, 128, 64] and ReLU activations. Long Short-Term Memory networks process transaction sequences for each account independently, capturing temporal dependencies without explicit graph structure.

Graph-based baselines utilize established graph neural network architectures, employing standard formulations. Vanilla Graph Convolutional Networks implement the Kipf & Welling approximation of spectral graph convolutions [Kipf & Welling, 2017] with 3 layers and hidden dimensions [128, 64, 32] matching the proposed architecture for fair comparison. Graph Attention Networks introduce multi-head attention mechanisms with four heads and an attention dropout rate of 0.2. GraphSAGE employs mean aggregation sampling, 25 neighbors per node at each layer. Note: For the Elliptic dataset with average degree 2.3, most nodes have fewer than 25 neighbors, so sampling effectively aggregates all available neighbors. We maintain this configuration for consistency across datasets. All neural models utilize PyTorch 1.13, along with PyTorch Geometric 2.3, for graph operations. Training utilizes NVIDIA A100 GPUs with mixed-precision computation. Hyperparameter tuning explores configuration spaces through grid search over learning rates [0.0001, 0.001, 0.01], hidden dimensions [32, 64, 128, 256], and regularization strengths. The validation set guides hyperparameter selection based on F1-score optimization.

Figure 3: Model Performance Comparison Across Multiple Metrics



This comprehensive performance visualization presents a multi-panel comparison of all evaluated methods. The main panel displays a grouped bar chart with six model categories along the x-axis: Traditional ML (Random Forest, XGBoost, Logistic Regression), Deep Learning (MLP, LSTM), and Graph Methods (GCN, GAT, GraphSAGE, Proposed). Each model displays five clustered bars, representing Precision, Recall, F1-Score, AUC-ROC, and AUC-PR metrics, with heights corresponding to percentage values ranging from 0 to 100 on the y-axis. Colors follow a consistent scheme with Precision in dark blue, Recall in green, F1-Score in orange, AUC-ROC in purple, and AUC-PR in red. The proposed method clearly shows the tallest bars across all metrics. Error bars display 95% confidence intervals computed from 5 independent runs with different random seeds using the same temporal split. The secondary panel to the right presents a radar chart comparing the proposed method against the best baseline, with five axes extending from the center for each metric. Two overlapping polygons in different colors show the performance profiles, with the proposed

method's polygon consistently extending further. The bottom panel displays a false favorable rate comparison using a horizontal bar chart, with bars extending to the left from a center line representing FPR percentages ranging from 0 to 100. Each model has a single bar colored by category, with the proposed method showing the shortest bar, indicating the lowest FPR. Numerical labels appear at bar endpoints showing exact percentages. A legend explains all color codings and symbols used throughout the figure.

4.3. Evaluation Metrics and Results

Performance metrics capture multiple dimensions of detection effectiveness relevant to operational deployment. Precision quantifies the proportion of flagged transactions that are genuinely suspicious, directly relating to investigative efficiency. Recall measures the fraction of actual illicit transactions successfully identified. F1-score provides a harmonic mean balancing precision and recall. The Area Under the Receiver Operating Characteristic curve evaluates performance across all possible classification thresholds. Area under the Precision-Recall curve proves particularly informative for imbalanced datasets. False positive rate analysis examines the proportion of legitimate transactions incorrectly flagged as suspicious. Traditional rule-based systems often have alert precision below 5% (i.e., FDR > 95%), which differs from the sample-level FPR reported here. The evaluation quantifies the FPR reduction achieved through learned features.

Table 4: Quantitative Performance Comparison on Elliptic Bitcoin Dataset

Method	Precision (%)	Recall (%)	F1-Score (%)	AUC-ROC	AUC-PR	FPR (%)
Random Forest	61.3 ± 3.2	58.7 ± 2.8	59.9 ± 2.5	0.823	0.647	38.2
XGBoost	64.8 ± 2.9	62.1 ± 3.1	63.4 ± 2.7	0.841	0.672	35.7
Logistic Regression	52.4 ± 4.1	49.8 ± 3.5	51.0 ± 3.3	0.762	0.581	47.3
MLP	68.2 ± 3.3	64.5 ± 3.0	66.3 ± 2.8	0.857	0.701	32.1
LSTM	70.5 ± 2.7	67.3 ± 2.9	68.8 ± 2.6	0.872	0.728	29.8
GCN	75.3 ± 2.4	71.8 ± 2.6	73.5 ± 2.3	0.894	0.765	24.5
GAT	77.9 ± 2.2	74.2 ± 2.5	76.0 ± 2.1	0.908	0.789	22.1
GraphSAGE	76.1 ± 2.6	72.6 ± 2.8	74.3 ± 2.4	0.899	0.771	23.8
Proposed Method	89.7 ± 1.8	85.2 ± 2.0	87.3 ± 1.7	0.952	0.876	10.3

Table 5: Ablation Study Results on IBM IT-AML Dataset

Configuration	F1-Score (%)	AUC-PR	FPR (%)	Description
Full Model	87.3 ± 1.7	0.876	10.3	Complete proposed architecture
w/o Attention	81.4 ± 2.3	0.823	16.7	Remove attention mechanisms
w/o Temporal	79.8 ± 2.5	0.804	18.9	Remove LSTM components
w/o Heterogeneous	77.2 ± 2.8	0.781	21.4	Single node/edge type
w/o Cross-Border	80.6 ± 2.4	0.815	17.3	Remove jurisdictional features
w/o Feature Fusion	82.9 ± 2.1	0.839	14.8	Only graph-learned features
w/o Focal Loss	83.5 ± 2.2	0.847	13.2	Standard cross-entropy

Ablation studies systematically examine the contribution of individual architectural components. The removal of the attention mechanism reduces the F1-score by 5.9 percentage points, demonstrating the value

of weighted neighborhood aggregation. Temporal feature integration through LSTM contributes 7.5 percentage points. On the IBM IT-AML dataset, heterogeneous graph formulation accounts for 10.1 percentage points (F1: 87.3 → 77.2); on the Elliptic dataset, the contribution is 3.2 percentage points (F1: 87.3 → 84.1). On the IBM IT-AML dataset, cross-border specific features contribute 6.7 percentage points; these features are not available for the Elliptic dataset. Feature fusion between graph-learned representations and handcrafted features provides a 4.4 percentage points improvement. The focal loss function addresses class imbalance more effectively than standard cross-entropy, contributing to a 3.8 percentage point improvement.

Comparison with state-of-the-art methods demonstrates substantial improvements across all evaluation metrics. The proposed approach achieves 87.3% F1-score, demonstrating a favorable precision–recall trade-off with substantially lower false-positive rate than the strongest baseline. Statistical significance testing validates that observed performance differences exceed random variation. Paired t-tests comparing the proposed method against each baseline yield p-values below 0.001 across all metrics. Computational efficiency analysis examines the practical feasibility of the proposed approach for large-scale deployment. Training time on the Elliptic dataset reaches 847 seconds across 100 epochs. Inference latency averages 0.034 seconds per batch (batch size=512), corresponding to 0.066 milliseconds per node, or approximately 15,000 nodes per second throughput (QPS). Memory consumption peaks at 6.2 GB.

5. Discussion and Conclusion

5.1. Key Findings and Insights

The experimental evaluation establishes the effectiveness of graph-based feature learning for cross-border anti-money laundering detection. Quantitative results demonstrate substantial performance improvements over both traditional handcrafted features and standard graph neural network baselines. The 87.3% F1-score achievement represents a practical detection capability that approaches operational requirements. The false positive rate decreases from 38.2% (Random Forest baseline) to 10.3% (proposed method), representing a relative reduction of 73% calculated as: (38.2-10.3)/38.2 × 100% = 73.0%. The false positive reduction from 38.2% using handcrafted features to 10.3% with learned representations translates to approximately a 73% decrease in investigative burden. The learned features automatically discover complex patterns that domain experts struggle to specify manually. Attention weight analysis reveals that the network identifies multi-hop transaction chains characteristic of layering schemes. Cross-border specific patterns emerge through jurisdictional embedding analysis, with the learned representations clustering high-risk jurisdictions characterized by weak regulatory enforcement. Currency exchange anomalies detected by the model include strategic timing of conversions and exploitation of exchange rate volatility.

5.2. Limitations and Future Directions

Current limitations constrain the immediate applicability of the proposed framework in certain deployment contexts. Label scarcity represents a fundamental challenge for supervised learning approaches in anti-money laundering applications. Real-world financial institutions possess limited quantities of confirmed suspicious activity reports. Computational overhead for real-time processing of massive transaction graphs presents scaling challenges for deployment at large financial institutions. Generalization across different financial institutions and countries raises concerns about the transferability of models. Future research opportunities span multiple promising directions. Federated learning architectures enable collaborative model training across institutions without centralizing sensitive transaction data. Integration with large language models offers potential for enhanced reasoning about transaction semantics. Continual learning frameworks provide mechanisms for adapting to evolving money laundering tactics through incremental model updates.

5.3. Concluding Remarks

This research presents a comprehensive graph-based feature learning framework addressing critical challenges in cross-border anti-money laundering detection. The proposed approach automatically discovers discriminative features capturing complex transaction patterns through specialized graph neural architectures. Extensive experimental evaluation demonstrates substantial performance improvements, achieving an 87.3% F1-score with a 73% reduction in false positives (from 38.2% to 10.3%). The framework contributions extend beyond quantitative performance gains to provide interpretable insights into suspicious transaction patterns. The broader impact of financial crime prevention encompasses multiple stakeholders. Financial institutions benefit from reduced false positive burden, enabling more efficient allocation of investigative resources. Regulatory bodies gain enhanced monitoring capabilities. The proposed framework provides a foundation for next-generation anti-money laundering systems.

References

[1]. D. Cheng, Y. Ye, S. Xiang, Z. Ma, Y. Zhang, and C. Jiang, "Anti-money laundering by group-aware deep graph learning," IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 12, pp. 12444-12457, 2023.

[2]. B. Oztas, D. Cetinkaya, F. Adedoyin, M. Budka, G. Aksu, and H. Dogan, "Transaction monitoring in anti-money laundering: A qualitative analysis and points of view from industry," Future Generation Computer Systems, vol. 159, pp. 161-171, 2024.

[3]. Alarab and S. Prakoonwit, "Graph-based LSTM for anti-money laundering: Experimenting temporal graph convolutional network with bitcoin data," Neural Processing Letters, vol. 55, no. 1, pp. 689-707, 2023.

[4]. M. R. Karim, F. Hermsen, S. A. Chala, P. De Perthuis, and A. Mandal, "Scalable semi-supervised graph learning techniques for anti money laundering," IEEE Access, vol. 12, pp. 50012-50029, 2024.

[5]. B. Dumitrescu, A. Băltoiu, and Ș. Budulan, "Anomaly detection in graphs of bank transactions for anti money laundering applications," IEEE Access, vol. 10, pp. 47699-47714, 2022.

[6]. J. Schmidt, D. Pasadakis, M. Sathe, and O. Schenk, "GAMLNet: A graph based framework for the detection of money laundering," in 2024 11th IEEE Swiss Conference on Data Science (SDS), pp. 241-245, 2024.

[7]. N. Pocher, M. Zichichi, F. Merizzi, M. Z. Shafiq, and S. Ferretti, "Detecting anomalous cryptocurrency transactions: An AML/CFT application of machine learning-based forensics," Electronic Markets, vol. 33, no. 1, p. 37, 2023.

[8]. Q. Wang, W. T. Tsai, and T. Shi, "GraphALM: Active learning for detecting money laundering transactions on blockchain networks," IEEE Network, 2024.

[9]. H. Huong, X. Nguyen, T. K. Dang, and P. T. Tran-Truong, "Money laundering detection using a transaction-based graph learning approach," in 2024 18th International Conference on Ubiquitous Information Management and Communication (IMCOM), pp. 1-8, 2024.

[10]. F. Irshad, T. Alkhalifah, F. Alturise, and Y. D. Khan, "GCF-MLD: Integrated approach for money laundering detection using machine learning and graph network analysis," IEEE Access, 2024.

[11]. U. G. Ketenci, T. Kurt, S. Önal, C. Erbil, S. Aktürkoğlu, and H. Ş. İlhan, "A time-frequency based suspicious activity detection for anti-money laundering," IEEE Access, vol. 9, pp. 59957-59967, 2021.

[12]. J. Song, S. Zhang, P. Zhang, J. Park, Y. Gu, and G. Yu, "Illicit social accounts? Anti-money laundering for transactional blockchains," IEEE Transactions on Information Forensics and Security, 2024.