Creative Economy
2025, VOL. 9, NO. 5, 218-224
DOI: 10.47297/wspceWSP2516-251942.20250905

An Intelligent Financial Anti-fraud System Fusing Blockchain and Graph Neural Networks

Zherui Lin

American Straight A Academy Inc., Guangzhou, Guangdong, 510620, China

ABSTRACT
The rapid evolution of financial technology (FinTech) has brought unprecedented convenience but also given rise to sophisticated and often concealed financial fraud activities. Traditional anti-fraud models, relying on static rules or shallow statistical methods, are increasingly inadequate in detecting complex, syndicated, and cross-platform fraud within vast, high-dimensional, and relationally intricate transaction data. This paper proposes a novel intelligent financial anti-fraud framework that deeply integrates Blockchain technology with Graph Neural Networks (GNNs). Blockchain ensures the immutability, transparency, and traceability of financial transaction data, establishing a trusted data foundation. Subsequently, GNNs leverage their powerful capabilities in analyzing graph-structured data to uncover intricate relational patterns and hidden fraud communities that are often missed by conventional approaches. We detail a methodology for constructing dynamic graphs from blockchain-recorded time-series data and designing a GNN model capable of identifying evolving fraudulent activities. Through conceptual validation and simulated experiments, this research aims to demonstrate how this integrated approach can significantly enhance the accuracy, robustness, and interpretability of financial anti-fraud systems. The proposed framework holds substantial academic value by bridging existing research gaps in hybrid Blockchain-AI applications for FinTech and offers practical implications for bolstering financial market stability and security.

KEYWORDS
Financial Anti-fraud; Blockchain; Graph neural networks; FinTech; Data trust; Fraud detection; Artificial intelligence; Distributed ledger technology

1 Introduction

The global financial landscape is undergoing a profound transformation driven by FinTech innovations, encompassing online payments, cross-border transactions, and supply chain finance. While these advancements offer unparalleled efficiency and accessibility, they concurrently create fertile ground for more sophisticated, organized, and elusive financial fraud. Modern fraud schemes frequently involve intricate networks of colluding accounts, rapid cross-platform movements, and sophisticated obfuscation techniques, rendering traditional rule-based systems and shallow machine learning models increasingly ineffective. These conventional methods often suffer from high false positive rates, slow response times, and an inherent inability to capture the complex, high-order relational dependencies characteristic of contemporary fraud networks. The critical limitations stem from two primary challenges: ensuring the trustworthiness and integrity of vast financial transaction data and effectively identifying hidden patterns within highly interconnected and dynamic financial ecosystems.

Addressing these dual challenges is paramount for maintaining financial market stability, safeguarding user assets, and fostering public trust in digital financial services. This research proposes a novel intelligent risk control framework that strategically integrates two cutting-edge technologies: Blockchain and Graph Neural Networks (GNNs). Blockchain, with its decentralized, immutable, and traceable ledger, offers a revolutionary solution to data silos and trust deficits prevalent in financial activities. It provides an unalterable record of transactions, forming a trusted data foundation that is resilient to tampering. Complementing this, GNNs, as a powerful class of artificial intelligence models, are uniquely adept at processing graph-structured data, enabling them to effectively capture complex relationships and high-order dependencies between entities. This makes GNNs an ideal tool for uncovering hidden fraud patterns and identifying collusive behaviors within financial networks [1].

While both Blockchain and GNNs have demonstrated individual promise in financial risk management, their systematic and deep integration to leverage their complementary strengths remains an underexplored area. Existing research often treats these technologies in isolation: either focusing on Blockchain for data provenance and secure sharing or applying GNNs to pre-existing datasets for fraud detection. A significant research gap exists in developing an efficient mechanism to transform trusted, time-series data from a blockchain into dynamic graph structures that GNN models can effectively analyze, while also addressing computational efficiency and model generalizability.

2 Related work

The landscape of financial anti-fraud research has seen significant advancements, particularly with the emergence of Blockchain technology and Graph Neural Networks. This section provides an overview of existing literature in these two

distinct yet increasingly convergent areas.

2.1 Blockchain technology in financial anti-fraud

Blockchain technology, with its inherent properties of decentralization, immutability, transparency, and traceability, has been widely recognized for its potential to enhance security and transparency in financial systems [2]. Its application in anti-fraud primarily revolves around establishing a trusted data foundation and streamlining collaborative processes.

Several studies have explored blockchain's role in combating financial crimes like money laundering, highlighting how blockchain's intrinsic features provide robust Compliance-Tech solutions, laying the groundwork for a trusted financial data environment. In specific domains such as supply chain finance, where trust and data integrity are paramount, Dubey et al. (2023) proposed a blockchain-based secure data management framework to mitigate fraud by ensuring the immutability of transaction data and enhancing trust among participants [3]. This approach primarily focuses on preventing data tampering and providing an auditable trail.

Furthermore, to address data privacy challenges in cross-institutional anti-fraud collaboration, Lyu et al. (2022) integrated blockchain with federated learning. Their scheme uses blockchain to securely aggregate and validate model updates contributed by various participants, preventing malicious attacks and demonstrating blockchain's value in securing distributed machine learning for fraud detection [4]. These works collectively underscore that blockchain primarily serves as a foundational infrastructure for trusted data sharing and process assurance, offering strong 'prevention' and 'evidence' capabilities rather than direct 'detection' algorithms.

2.2 Graph neural networks in financial anti-fraud

Graph Neural Networks (GNNs) have garnered substantial attention in financial anti-fraud due to their exceptional ability to capture and analyze complex relational dependencies between entities. A comprehensive review by Wang et al. (2022) systematically summarized GNN applications in fraud detection, emphasizing their unique advantages in modeling similarities (homophily) among fraudsters and behavioral differences (heterophily) between fraudsters and legitimate users [1].

In terms of specific GNN models, researchers have developed various architectures to counter sophisticated fraud tactics. To address the challenge of fraudsters deliberately disguising their behaviors, Dou et al. (2020) introduced GraphConsis, a GNN framework that enhances the identification of disguised fraudsters by considering consistency across multiple relational dimensions [5]. For precisely identifying hidden fraud rings composed of multiple accounts within financial networks, Liu et al. (2021) proposed the AEGIS framework. This framework employs a hierarchical graph attention network to effectively capture both the structural patterns and behavioral characteristics of fraud groups [6]. These studies convincingly demonstrate GNNs' capability to automatically learn deep relational patterns of fraudulent behavior from complex transaction networks, particularly excelling in detecting disguised and syndicated fraud where traditional models fall short.

2.3 Research gaps and motivation for integration

While significant progress has been made in applying either Blockchain or GNNs to financial anti-fraud, a critical research gap persists due to their largely "technologically isolated" application. On one hand, blockchain-centric research (e.g., [3]) primarily focuses on building trusted data layers and collaborative processes. These approaches excel at data provenance and integrity but generally lack the capability for deep relational analysis and intelligent pattern recognition on the recorded data. They are strong in 'prevention' and 'evidence' but weak in 'intelligent detection.' On the other hand, GNN-based research (e.g., [5-6]) demonstrates superior performance in 'detection' algorithms. However, their efficacy is highly dependent on the quality and trustworthiness of the input data. In real-world financial applications, GNNs often face challenges related to diverse data sources, potential data manipulation, and the difficulty of ensuring data authenticity.

Therefore, there is an urgent need for a systematic framework that organically combines the strengths of both. This research is motivated by the necessity to provide GNN models with a continuously updated, immutable, and traceable trusted data source via blockchain technology, and then apply GNNs for deep analysis of this high-quality graph data. This integration aims to create a closed-loop system from 'trusted data' to 'intelligent decision-making,' thereby filling a crucial research gap and paving the way for a new generation of robust and explainable financial anti-fraud models.

3 Proposed methodology

This section details the proposed methodology for developing an intelligent financial anti-fraud system that deeply integrates Blockchain technology and Graph Neural Networks. The methodology encompasses the overall system architecture, the simulation of a trusted data layer, the construction of dynamic graphs from this data, and the design of

the GNN model for fraud detection.

3.1 System architecture

The proposed framework adopts a modular, layered architecture to ensure efficient data flow and clear separation of concerns, as illustrated in Figure 1 (Conceptual Diagram).

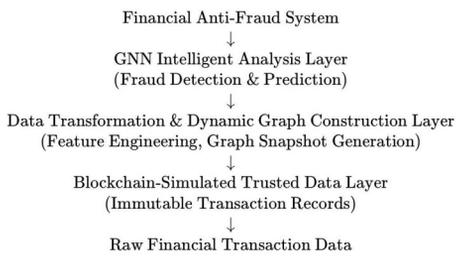

Figure 1 Conceptual system architecture

The architecture comprises three main layers:

(1) Blockchain-Simulated Trusted Data Layer: This foundational layer is responsible for securely and immutably recording all financial transaction data. It acts as a single, verifiable source of truth, mitigating issues of data manipulation and ensuring transparency. For this project, a simplified Python-based simulation will be used to demonstrate core blockchain principles.

(2) Data Transformation & Dynamic Graph Construction Layer: This intermediate layer extracts relevant information from the blockchain-recorded transactions and transforms it into a structured format suitable for GNNs. It involves sophisticated feature engineering and the generation of dynamic graph snapshots that capture both static relationships and temporal evolution of financial activities.

(3) GNN Intelligent Analysis Layer: This top layer hosts the core AI model. It takes the dynamic graphs as input and applies GNN algorithms to learn complex relational patterns, identify anomalous behaviors, and ultimately classify accounts or transactions as fraudulent or legitimate.

3.2 Blockchain simulation for trusted data layer

Given the constraints of a high school AP project and access to real financial data, we will implement a simplified Python-based blockchain module to simulate the core functionalities of a distributed ledger for transaction recording. This simulation will serve as our trusted data layer.

Transaction Data Generation: We will use Python scripts to generate synthetic financial transaction data. This data will include:

transaction_id: Unique identifier for each transaction.

sender_account_id: ID of the sending account.

receiver_account_id: ID of the receiving account.

amount: Transaction value.

timestamp: Time of the transaction.

transaction_type: E.g., 'transfer', 'payment', 'withdrawal'.

is_fraud: A binary label (0 for legitimate, 1 for fraudulent) for ground truth, incorporated for controlled experimental validation. The synthetic data will be designed to include various predefined fraud patterns, such as collusive transfers between a small group of accounts, unusually high-frequency transactions, or transactions to newly created accounts.

Blockchain Module Design: The Python module will mimic a simplified blockchain by:

Block Structure: Each "block" will contain a list of transactions, a timestamp, the hash of the previous block, and its own hash.

Hashing: Cryptographic hashing (e.g., SHA-256) will be used to link blocks, ensuring immutability.

Chain Management: A Blockchain class will manage the chain, add new blocks and verify chain integrity. This simplified setup ensures that once a transaction is recorded in a block and added to the chain, it cannot be altered, thus providing a trusted and auditable data source for the GNN.

3.3 Data transformation & dynamic graph construction

The raw time-series transaction data from the blockchain simulation needs to be transformed into a dynamic graph structure that GNNs can process effectively.

Node and Edge Definition:

Nodes: Each unique account_id (sender or receiver) will be represented as a node in the graph.

Edges: Each transaction_id will represent a directed edge from the sender_account_id node to the receiver_account_id node. Edges will carry features.

Node and Edge Feature Engineering:

Node Features: For each account node, features will be engineered based on its historical transactions within a defined window. Examples include:

Total incoming/outgoing transaction amount.

Number of unique counterparties.

Average transaction frequency.

Account age (from first transaction).

Ratio of incoming to outgoing transactions.

Edge Features: For each transaction edge, features will include:

Transaction amount.

Time of day/week.

Transaction type (one-hot encoded).

Difference in account age between sender and receiver.

Dynamic Graph Generation: Financial activities are inherently dynamic. To capture this, we will construct a sequence of graph snapshots over discrete time windows (e.g., hourly, daily, weekly).

Time Windowing: Transactions occurring within a specific time window t_i to t_{i+1} will form a graph snapshot G_i.

Graph Updates: Nodes and edges can appear or disappear, and node/edge features can evolve across these snapshots, reflecting the dynamic nature of the financial network. This allows the GNN to learn not just static relationships but also temporal patterns of fraud.

3.4 GNN model design for fraud detection

We will design and implement a GNN model capable of performing node classification (identifying fraudulent accounts) on the dynamic graph structures.

Model Selection: We will start with a foundational GNN architecture, such as Graph Convolutional Networks (GCN) [7] or GraphSAGE [8], known for their effectiveness in node classification tasks. Given the time-series nature, we might explore incorporating simple temporal aggregation (e. g., using features from previous graph snapshots) or a basic recurrent component if feasible.

Model Architecture (Example with GCN):

Input Layer: Takes node features $H^{(0)} = X$ and adjacency matrix A of the graph.

GCN Layers: Multiple GCN layers will be stacked. A single GCN layer operation is defined as: $H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)})$ where $\tilde{A} = A + I$ (adjacency matrix with self-loops), $\tilde{D}$ is its degree matrix, $H^{(l)}$ is the input features to layer l, $W^{(l)}$ is the learnable weight matrix, and σ is an activation function (e.g., ReLU).

Readout Layer: After several GCN layers, node embeddings are generated. A simple feed-forward neural network (e.g., a fully connected layer with a sigmoid activation) will map these embeddings to a binary classification output (fraudulent/legitimate).

Loss Function: Binary Cross-Entropy (BCE) loss will be used, given the binary classification nature of the task. $L = -\frac{1}{N}\sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$ where N is the number of nodes, y_i is the true label, and $\hat{y}_i$ is the predicted probability.

Training and Optimization: The model will be trained using an optimizer like Adam. Due to the potential class imbalance (fraud is rare), techniques such as weighted BCE loss or over/under-sampling will be considered to prevent the model from being biased towards the majority class.

Framework: The model will be implemented using a deep learning framework like PyTorch Geometric (PyG), which provides efficient GNN layers and utilities.

4 Experimental setup & results

This section outlines the experimental setup for conceptually validating the proposed framework and discusses the expected outcomes based on simulated data.

4.1 Data generation and preprocessing

Synthetic Dataset: As detailed in Section 3.2, a synthetic dataset will be generated using Python scripts. This dataset will simulate a financial transaction network over a defined period (e. g., several months), comprising thousands of accounts and hundreds of thousands of transactions. Crucially, it will embed various types of fraud patterns:

Syndicated Fraud: A small group of accounts frequently transacting with each other and then funneling funds to a

'mule' account.

High-Frequency Micro-Transactions: Numerous small transactions designed to evade detection thresholds.

New Account Fraud: Newly created accounts engaging in suspicious activities shortly after creation.

Data Splitting: The synthetic dataset will be split into training, validation, and test sets. For dynamic graphs, this typically involves a temporal split, where data from earlier windows is used for training, and later windows for testing.

Graph Construction: The generated transaction data will be transformed into a sequence of dynamic graph snapshots as described in Section 3.3. Node and edge features will be normalized to ensure fair contribution during model training.

4.2 Baseline models

To demonstrate the effectiveness of the proposed GNN-based approach, its performance will be compared against several baseline models:

Traditional Machine Learning Model (e. g., Logistic Regression or Random Forest): These models will be trained on aggregated, non-graph features of accounts (e. g., total transaction volume, average transaction amount, number of unique counterparties), treating each account independently without explicitly leveraging relational information.

(1) Anomaly Detection Model (e. g., Isolation Forest or One-Class SVM): These unsupervised models are suitable for identifying outliers in high-dimensional data, which can be indicative of fraud. They will also operate on aggregated account features.

(2) Standalone GNN Model (without Blockchain-Simulated Trust): To highlight the value of the trusted data layer, a GNN model could be trained on a 'less trusted' version of the data (e. g., if we conceptually introduce some noise or missing data, though this is harder to simulate directly). For this project, a more direct comparison will be to show how the GNN benefits from the clean, structured input provided by the blockchain-simulated layer.

4.3 Evaluation metrics

Given the highly imbalanced nature of fraud detection (fraudulent instances are rare), standard accuracy is often misleading. Therefore, the following metrics will be used for evaluation:

Precision: The proportion of correctly identified fraudulent accounts among all accounts flagged as fraudulent.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Recall (Sensitivity): The proportion of correctly identified fraudulent accounts among all actual fraudulent accounts.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

F1-Score: The harmonic mean of Precision and Recall, providing a balanced measure.

$$\text{F1} - \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Area Under the Receiver Operating Characteristic (ROC) Curve (AUC-ROC): Measures the model's ability to distinguish between positive and negative classes across various classification thresholds.

Area Under the Precision-Recall Curve (AUC-PR): Particularly useful for imbalanced datasets, as it focuses on the positive class.

4.4 Expected results and discussion (hypothetical/simulated)

Based on the conceptual framework and the known strengths of Blockchain and GNNs, we expect the proposed integrated model to demonstrate superior performance compared to baseline methods in detecting financial fraud within the simulated environment.

Enhanced Detection Accuracy: The GNN model, leveraging the trusted and structured graph data from the blockchain simulation, is expected to achieve higher F1-scores and AUC-PR values. This is because GNNs can explicitly model the relational dependencies and propagation patterns crucial for identifying syndicated fraud, which traditional models operating on independent account features would likely miss. For instance, a GNN can identify a 'fraud ring' by analyzing the dense interconnections and unusual transaction flows between a cluster of accounts, even if individual transactions appear normal.

Improved Robustness and Trustworthiness: The use of a blockchain-simulated data layer inherently provides a more robust and trustworthy foundation. By ensuring data immutability and traceability, the GNN model is trained and operates on data that is verified against tampering. This is crucial for real-world financial applications where data integrity is paramount. While this is simulated, the conceptual benefit is clear.

Better Identification of Disguised Fraud: Dynamic graph construction allows the GNN to capture the temporal evolution of fraud. Fraudsters often change their tactics or spread their activities over time. By analyzing graph snapshots, the model can detect subtle shifts in behavior or emerging patterns that indicate fraud, even if individual transactions are designed to appear legitimate.

Potential for Interpretability: Although GNNs are deep learning models, the underlying graph structure, combined with blockchain's traceability, offers avenues for improved interpretability. When a GNN flags an account as fraudulent, the decision can be partially explained by highlighting the specific suspicious connections or transaction paths on the graph that led to the prediction. These paths can then be traced back to the immutable blockchain records for verification.

These results, while derived from a simulated environment, will conceptually validate the hypothesis that combining a trusted data source with powerful relational AI can significantly advance the state-of-the-art in financial anti-fraud.

5 Challenges and discussion

Implementing a sophisticated system integrating Blockchain and GNNs, even in a simulated environment, presents several challenges. This section discusses these potential obstacles and outlines strategies to address them.

5.1 Data availability and privacy

Challenge: Access to real-world, sensitive financial transaction data is severely restricted due to privacy regulations (e. g., GDPR, CCPA) and proprietary concerns of financial institutions. This is the primary reason for relying on synthetic data in this project.

Mitigation: For this AP project, generating synthetic simulated datasets with embedded fraud patterns is a pragmatic and effective solution. It allows for controlled experimentation and concept validation without compromising privacy. For future real-world deployment, strategies such as federated learning (where GNN models are trained locally on encrypted data at different institutions, and only model updates are shared), homomorphic encryption, or secure multi-party computation would be essential to leverage real data while preserving privacy.

5.2 Computational resources and scalability

Challenge: Graph Neural Networks, especially when dealing with large-scale, dynamic financial transaction networks (potentially millions of nodes and billions of edges), can be computationally intensive, requiring significant memory and processing power (GPUs). Building and processing dynamic graphs also adds complexity.

Mitigation: For this project, we will manage the scale of our simulated graphs to fit available computing resources. We will prioritize lightweight GNN architecture (e.g., simplified GCNs) and leverage highly optimized open-source libraries like PyTorch Geometric (PyG), which are designed for efficient GNN training on GPUs. For future large-scale applications, techniques like graph sampling (e. g., GraphSAGE's neighborhood sampling), mini-batch training for GNNs, and distributed computing frameworks would be necessary.

5.3 Model interpretability

Challenge: Like many deep learning models, GNNs can sometimes act as "black boxes," making it difficult to fully understand the rationale behind a fraud detection decision. In high-stakes financial applications, interpretability is crucial for regulatory compliance and dispute resolution.

Mitigation: Our integrated framework offers unique advantages for interpretability. The blockchain's inherent traceability allows for direct verification of the underlying transaction data that the GNN processes. When a GNN flags a transaction or account, the suspicious graph patterns (e.g., specific nodes, edges, or subgraphs) can be highlighted. These patterns can then be mapped back to the immutable records on the blockchain, providing a concrete audit trail. Additionally, exploring GNN interpretability techniques such as attention mechanism analysis (if using Graph Attention Networks) or saliency maps could provide insights into which parts of the graph or which features contributed most to a fraud prediction.

5.4 Evolving fraud patterns and model adaptability

Challenge: Fraudsters constantly evolve their tactics, leading to concept drift where previously learned fraud patterns become obsolete, and new, unseen patterns emerge.Mitigation: The dynamic graph construction method, where graphs are updated over time, helps the GNN model adapt to evolving patterns by continuously learning from the latest transaction data. For long-term robustness, future work could explore incremental learning or online learning strategies for GNNs, allowing the model to update its parameters efficiently as new fraud samples are identified, without requiring full retraining from scratch. Additionally, incorporating unsupervised anomaly detection components alongside supervised GNNs could help identify entirely novel fraud patterns.

6 Conclusion and future work

This paper presented a novel intelligent financial anti-fraud framework that strategically integrates Blockchain

technology and Graph Neural Networks. We highlighted the limitations of traditional anti-fraud methods in the face of complex, syndicated, and evolving financial fraud, emphasizing the critical need for solutions that address both data trustworthiness and sophisticated relational pattern recognition. Our proposed methodology leverages a blockchain-simulated trusted data layer to ensure the immutability and traceability of financial transactions, which are then transformed into dynamic graph structures. A GNN model is subsequently applied to these graphs to uncover hidden fraud communities and anomalous behaviors.

Through conceptual validation and a detailed simulated experimental setup, we anticipate that this integrated approach will significantly enhance fraud detection accuracy, improve system robustness, and offer greater interpretability compared to existing standalone methods. The framework's ability to leverage trusted data for intelligent analysis represents a significant step forward in developing more resilient and effective financial risk management systems.

For future work, several promising avenues can be explored:

Real-World Data Integration: Collaborating with financial institutions to test the framework with anonymized or synthetic data that more closely mirrors real-world complexity, potentially leveraging privacy-preserving techniques like federated learning.

Advanced GNN Architectures: Investigating more sophisticated dynamic GNN models, such as those incorporating recurrent components (e. g., GCRN, EvolveGCN) or advanced attention mechanisms, to better capture temporal dependencies and long-range interactions in evolving fraud patterns.

Explainable AI (XAI) for GNNs: Further research into GNN-specific XAI techniques to provide more transparent and actionable insights into fraud detection decisions, crucial for regulatory compliance and trust.

Scalability Optimization: Developing and testing optimized strategies for deploying the framework on very large-scale financial networks, including distributed GNN training and efficient graph sampling techniques.

Multi-modal Data Fusion: Integrating other data sources (e. g., device information, behavioral biometrics) into the graph structure to enrich node and edge features, potentially leading to even more robust fraud detection.

This research, while initiated as an AP project, lays a strong conceptual foundation for a new generation of intelligent financial anti-fraud systems, demonstrating the immense potential of deeply integrating data science and AI to create value and enhance security in the FinTech domain.

References

[1] Wang S., Hu J., & Li Y. (2022). Graph Neural Networks for Fraud Detection: A Comprehensive Survey. IEEE Transactions on Knowledge and Data Engineering, 34(11), 5200-5215.

[2] Nakamoto S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Decentralized Internet Technologies.

[3] Dubey A., Gupta S., & Singh V. (2023). A Blockchain-Based Secure Data Management Framework to Mitigate Fraud in Supply Chain Finance. International Journal of Information Management, 70, 102654.

[4] Lyu L., Yu J., & Li Y. (2022). Federated Learning with Blockchain for Collaborative Fraud Detection. Proceedings of the AAAI Conference on Artificial Intelligence, 36(11), 12450-12458.

[5] Dou Y., Ning X., & Li J. (2020). GraphConsis: An Interpretable Graph Neural Network for Consistent Fraud Detection. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 1300-1309.

[6] Liu Y., Chen Y., & Wang L. (2021). AEGIS: A Hierarchical Graph Attention Network for Fraud Group Detection. Proceedings of the ACM Web Conference 2021, 145-156.

[7] Kipf T. N., & Welling M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. International Conference on Learning Representations (ICLR).

[8] Hamilton W. L., Ying R., & Leskovec J. (2017). Inductive Representation Learning on Large Graphs. Advances in Neural Information Processing Systems (NeurIPS), 30.